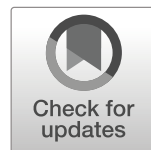



World Wide Web (2021) 24:1345–1368
<https://doi.org/10.1007/s11280-020-00800-3>



A deep-learning model for urban traffic flow prediction with traffic events mined from twitter

Aniekan Essien^{1,2}  · Ilias Petrounias³ · Pedro Sampaio¹ · Sandra Sampaio¹

Received: 3 May 2019 / Revised: 12 November 2019 / Accepted: 6 February 2020 /

Published online: 14 March 2020

© The Author(s) 2020

Abstract

Short-term traffic parameter forecasting is critical to modern urban traffic management and control systems. Predictive accuracy in data-driven traffic models is reduced when exposed to non-recurring or non-routine traffic events, such as accidents, road closures, and extreme weather conditions. The analytical mining of data from social networks – specifically twitter – can improve urban traffic parameter prediction by complementing traffic data with data representing events capable of disrupting regular traffic patterns reported in social media posts. This paper proposes a deep learning urban traffic prediction model that combines information extracted from tweet messages with traffic and weather information. The predictive model adopts a deep Bi-directional Long Short-Term Memory (LSTM) stacked autoencoder (SAE) architecture for multi-step traffic flow prediction trained using tweets, traffic and weather datasets. The model is evaluated on an urban road network in Greater Manchester, United Kingdom. The findings from extensive empirical analysis using real-world data demonstrate the effectiveness of the approach in improving prediction accuracy when compared to other classical/statistical and machine learning (ML) state-of-the-art models. The improvement in predictive accuracy can lead to reduced frustration for road users, cost savings for businesses, and less harm to the environment.

Keywords Traffic flow prediction · LSTM stacked autoencoders · Deep learning · Twitter-informed prediction · Intelligent transportation systems

1 Introduction

Reducing traffic congestion levels is an essential priority for cities worldwide with significant research interest over the past decades devoted to improving traffic speed prediction methods

This article belongs to the Topical collection: *Special Issue on Artificial Intelligence and Big Data Computing*

Guest Editors: Wookey Lee and Hiroyuki Kitagawa

✉ Aniekan Essien
aniekan.essien@manchester.ac.uk; a.e.essien@swansea.ac.uk

Extended author information available on the last page of the article

and the development of Intelligent Transportation Systems (ITS) [26]. The success of an ITS is mainly determined by the quality of traffic information provided to traffic stakeholders and the ability to apply traffic information towards developing policies, control systems and traffic prediction models. Short-term traffic prediction is a multidisciplinary research area combining contributions from various fields, such as mathematics, computer science, and engineering. Accurate traffic parameter prediction is challenging, mainly due to the dynamic, complex and stochastic nature of traffic. The complexities associated with traffic prediction stem from the nature of the traffic domain, which comprises constraints imposed by the physical traffic infrastructure such as road network capacities, traffic regulations and management policies, and behaviour of individual agents (road users), as well as exogenous factors, such as calendar (i.e. time of day, day of week, etc.), weather, accidents and incidents, events, road closures, to name a few [10].

Traffic data science has evolved over the years by expanding the multitude of data sources used to develop predictive models. Early studies investigated the importance of weather data on traffic flow parameters by influencing driving behaviour, travel demand, travel mode, road safety, and traffic flow characteristics. Furthermore, research has – over the years – shown that rainfall reduces traffic capacity and operating speeds, thereby increasing congestion and road network productivity loss. For instance, the authors in [9] reported that rainfall intensity affected urban traffic flow characteristics by reducing traffic speed by 4–9%, while traffic congestion at peak periods showed a significant relationship with temperature intensity. Despite the importance of weather as a traffic predictor, the majority of traffic prediction models used in ITSs assume clear weather conditions, thereby missing out on important sources of environmental data that could enable more accurate assessment of traffic network status [2].

Recent studies have, therefore, investigated the impact of the inclusion of non-traffic input datasets for urban traffic parameter prediction, many (if not all) of which have yielded improved prediction accuracies [11, 12, 22]. For instance, a deep bi-directional LSTM model was proposed in [11], which was trained using rainfall and temperature datasets in addition to traffic flow characteristics. The findings from the study showed an improvement in Predictive accuracy when compared to baseline (i.e. traffic-only) datasets. Similar results were obtained in studies that included non-traffic input data [22, 23]. This can be explained by the fact that data-driven traffic parameter prediction typically relies on predictive analytical techniques applied on historical data observations to extract patterns that can be used to predict future observations. This has been effective due to the typically seasonal, recurring/cyclical nature of urban traffic data. For instance, morning and evening rush hour peaks are easily predictable, and can, therefore, be anticipated. A model that is, therefore, adept at identifying and learning these patterns from historical datasets will be ‘skillful’ in predicting future traffic parameters.

However, in unusual or non-recurring situations, such as events or incidents that cannot be inferred from historical observations, even the most accurate predictive models will exhibit poor predictive performance [12]. Typical examples of non-recurring or stochastic events/incidents include accidents, lane closures, sporting, and public events. Given that such events can be sudden, unexpected and/or rare, the need for developing robust predictive models to enable accurate traffic prediction in these circumstances becomes necessary. Existing studies have presented research efforts at harnessing social media data for traffic parameter prediction, for instance, in [17] where a linear-regression-based optimisation technique is used for road traffic prediction. The authors tested their proposed model using online road traffic data obtained from the California Performance Measurement System (PeMS) in the USA. Similarly, [30] represents a study in which social media

data was used to develop a short-term traffic prediction model. The study incorporated tweet data to predict incoming traffic flow prior to sport game events, an approach that was evaluated using four models, namely ARIMA, neural network, support vector regression, and k -nearest neighbour (k -NN). A Kalman-filter model incorporating twitter data for real-time bus arrival prediction time is presented in [1].

Deep learning (DL) approaches have advanced image and speech recognition, natural language processing (NLP), and intelligent gamification, and are also being used in the field of short-term traffic prediction [8, 25, 33]. DL refers to techniques for learning complex features from high-level data in a hierarchical manner using stacked, multi-layered architectures [16]. Harnessing social media data within the traffic prediction cycle trained on the layers of DL architectures provides additional opportunities for improving the accuracy of traffic prediction. This study presents an attempt at contributing to this area of research. More specifically, we present an end-to-end implementation of a deep learning urban traffic flow prediction model that integrates real-time information obtained from traffic, weather-related data, and social media tweets (which can contain information about non-recurring or unexpected events) on urban traffic flow prediction. Road users, when stuck in traffic congestion, often tweet about their traffic status and location providing important real-time information to other road users or traffic management stakeholders.

Social media as an online discussion platform has seen a remarkable explosion in the last few years. Examples include Facebook,¹ Twitter,² Instagram,³ and Snapchat.⁴ These services are widely employed for communication, news reporting, and advertising events. Each of these social media platforms provides application programming interfaces (APIs) that enable data retrieval in real-time. Twitter is a public social media platform popular for short messages (up to 280 characters), thereby resulting in data streams with high velocity and timely dissemination of information concerning real-world events. Given the enormity and variance of information obtainable on twitter due to the large user base, numerous studies have sought to harness this online data repository for various data mining purposes, such as stock market prices [29], crime rate prediction [39], and traffic prediction [1, 15, 41].

Advanced Traveler Information Systems (ATIS), such as Waze and TomTom already capitalize on crowd-informed social media data to improve their traffic navigation and route guidance system. In general, many twitter accounts report current traffic conditions, which can be used by road users to infer future traffic conditions and inform the choice of travel mode. For instance, in Northern England, Highways England (@HighwaysNWEST), Traffic for Greater Manchester (TfGM @OfficialTfGM), @nwtrafficnews, and Waze (@WazeTrafficMAN) are typical examples of such Twitter accounts that provide road-traffic condition information. In addition to tweets posted by major organizations in the transportation sector, road users can also tweet on their respective timelines to broadcast (to their followers) current road traffic conditions, which can be mined to infer future traffic conditions.

We, therefore, present an approach towards urban traffic flow prediction that utilises information from tweet feeds that can contain information about non-recurring traffic events, in addition to traffic and weather-related datasets. The study in [11] displayed improved model predictive performance when weather-related (rainfall and temperature) datasets were

¹ <https://facebook.com/>

² <https://twitter.com/>

³ <https://www.instagram.com/>

⁴ <https://www.snapchat.com/>

integrated for urban traffic speed prediction. The approach presented in this current paper exemplifies an augmented and enhanced version of the one presented in [11], which is aimed at improving predictive accuracy. The focus on tweets as ancillary representations denoting traffic status stems from the tendency of Twitter users to react more quickly to events when compared to other social media platforms [31].

The contributions of this research are two-fold: (1) assessing empirically if the inclusion of tweets in addition to weather and traffic datasets improve urban traffic flow prediction, and (2) an end-to-end deep bi-directional LSTM autoencoder traffic flow prediction model implementation that uses tweet, weather, and traffic datasets for model training. The proposed model is evaluated using traffic, geo-specific tweet and weather datasets from A56 (Chester Road) in Stretford, Greater Manchester, United Kingdom. It is appreciated that due to the long training time of deep learning models, the addition of an additional data source (tweets) may significantly impact the overall model computational demand. We, therefore, employ an autoencoder architecture for the bi-directional LSTM neural network. The autoencoder, which can also serve as a dimensionality reduction component, allows the model to train in a shorter time since the input vector is reduced to a smaller dimensional space [16].

Owing to the spatially distributed and sequential nature of traffic (i.e. time series) data, it is critical that all the data contained therein should be maximally exploited. The underlying operation of LSTM RNNs is the positive or ‘forward’ propagation of the chronologically-arranged input time series data, in the direction of time step $t-1$ to time step t in a chain-like structure [19]. Therefore, in sequential datasets, it may be useful to include a bi-directional architecture, which considers backward propagation, passing the reverse sequence to the LSTM model. Intuitively, using a bidirectional LSTM for urban traffic data should result in a more accurate prediction, since it may sometimes be useful to ‘reverse-learn’ the data. For instance, it may be useful to learn backwards (say to a Friday night) in order to infer what the future traffic situation will be on the Saturday early morning or afternoon. If, for example, a lot of people, were celebrating until late on Friday night, then potentially fewer people will be travelling early Saturday morning or afternoon (for example, New Year’s Eve). A reverse example, on the other hand, can be shown in a situation where there is a forecast of snow for tomorrow, which may affect today’s traffic as people may want to go out for their shopping today to circumvent the forecasted snow condition tomorrow. Empirical results from the literature also demonstrate improvements when using bidirectional LSTMs over unidirectional LSTMs for traffic prediction [3, 13]. In our model, we included bi-directional LSTMs to improve the predictive performance of our model. In addition, due to the highly complex and patterned structure of urban traffic datasets, bi-directional sequence/representation learning may prove to be a more robust solution.

A significant challenge encountered in the inclusion of tweets in traffic prediction is the process of determining the level of authenticity, veracity and filtering high levels of noise in the unstructured datasets [15]. To account for this, we calibrated our algorithms to prioritise tweets from road-traffic organisation Twitter accounts – specifically Transport for Greater Manchester (@OfficialTfGM) and Waze (@WazeTrafficMAN). The findings from the study showed a significant improvement in the model prediction accuracy when information for Twitter feeds was included.

The remainder of this paper is organised as follows. Section 2 reviews existing relevant studies about short-term traffic prediction and the inclusion of tweets in traffic prediction. Section 3 outlines the proposed methodology including brief discussions about key concepts. Section 4 describes the datasets used for this study. Section 5 outlines the experimental setup,

model description and performance evaluation metrics. Section 6 presents the results of the study, while we conclude the research and present future work in Section 7.

2 Related studies

This section presents a review of related studies, beginning with a summarised technical background about short-term traffic prediction, parametric and nonparametric models. The section concludes with a review of traffic prediction studies integrating Twitter messages.

2.1 Short-term traffic prediction

Short-term traffic prediction techniques are broadly categorised into parametric approaches and nonparametric approaches. A parametric model refers to a model that summarises data by simplifying the input data to a known function. Parametric models are sometimes called model-based prediction methods, given that the model structure is predetermined using computed model parameters on empirical data [16]. One of the pioneer parametric prediction models is the autoregressive integrated moving average (ARIMA) model [37]. The ARIMA model is defined by the differencing equation:

$$\widehat{Y}_t = \phi_1 y_{t-1} + \phi_2 y_{t-2} \dots + \phi_p y_{t-p} + a_t - \theta_1 a_{t-1} \dots - \theta_q a_{t-q} \quad (1)$$

where:

y	represents a general time series.
\widehat{Y}_t	denotes the forecast of the time series at time t .
$y_{t-1} \dots y_{t-p}$	represents the previous p values of the time series, which forms the auto-regression term
$\phi_1 \dots \phi_p$	coefficients to be determined by model fitting
$a_t \dots a_{t-q}$	zero mean white noise, and is the moving average term
$\theta_1 \dots \theta_{t-q}$	coefficient to be determined by model fitting
p	number of auto-regression terms
q	number of moving-average terms

The variables in eq. (1), p and q are integers greater than or equal to zero and represent the autoregressive and moving average components respectively. It is a requirement for the input time series to be stationary for the successful application of ARIMA (p, d, q) model. For this reason, differencing is sometimes applied to induce stationarity of the dataset, which involves the consecutive differences between the observations. Thus, the third parameter, the difference (d) is calculated such that if $d = 0$; $y_t = y_t$ and if $d = 1$; $y_t = y_{t-1}$, etc.

However, the primary assumption made by this model is the stationarity of the mean, variance, and auto-correlation. This constitutes a significant drawback as it tends to miss out on extreme values, which are common in traffic datasets [14]. Traffic parameters tend to display peaks especially at rush hour, as well as the swift oscillations with incidents or accidents. Therefore, ARIMA predictors display weaknesses when applied to traffic forecasting.

In nonparametric models, the algorithms ‘learn’ from the data, thereby selecting the function that best fits the training dataset, meaning they can fit many functions to a particular

dataset [16]. The k -Nearest Neighbour (k -NN) is mostly considered the easiest to implement nonparametric machine learning model [34] and has been widely explored in traffic prediction [34, 40, 43]. The logic driving the model is that the with the categorisation of k most similar observations in a feature space, then a new observed sample will likely belong to this category [42]. The parameters of the model are: state vector, distance metric, number of nearest neighbours k , and the prediction algorithm. The distance metric measures the degree of approximation between the sample and test data. This is calculated using Euclidean distance, represented as:

$$d_i = \sqrt{\sum_j w_j (V_j - v_{ji})^2} \quad (2)$$

Artificial Neural Networks (ANN) are another class of nonparametric traffic prediction models, inspired by the inner workings of the human brain [5]. This class of prediction models can deal with multi-dimensional and non-linear data, in addition to adept learning ability. The fundamental model component for ANN models is the multilayer perceptron (MLP), which is described in eq. (3):

$$y = h \left(\varphi_0 + \sum_{j=1}^N \varphi_j g \left(\theta_0 + \sum_{i=1}^M \theta_i x_i \right) \right) \quad (3)$$

Where M and N represent the number of neurons in the input layer and hidden layer respectively, and g and h are the transfer functions. θ refers to the weight value for the neurons in the input layer, while φ is the weight or bias for the hidden layer. Neural networks reduce error by employing optimising algorithms, such as Back Propagation. However, when applied to time-series analysis, traditional neural networks show deficiencies since they neglect the temporal dimension of time-series data, which led to the development of *recurrent neural networks* (RNN) [7].

RNNs [36] are neural networks that preserve the temporal dimension of time series data using a recurrent hidden state. The basic functioning of RNN networks incorporates loops or ‘recurrent’ components to connect the neuron to itself and unfold many times such that it can preserve the temporal dimension found in sequential data. RNNs have hidden states that are updated by the sequential information obtained from input time series data with outputs that depend on these hidden states. A simple mechanism of how the RNN is unfolded into a network is shown in Figure 1. In Figure 1, U and V represent the weights of the hidden layer and output layer respectively, while W represents the transition weights of the hidden state. The hidden state of the network at time t is computed by the element-wise product of the input vector and the previous network hidden state h_{t-1} . This is mathematically computed using eq. (4):

$$h_t = \sigma(W_{hx}x_t + W_{hh}h_{t-1} + b) \quad (4)$$

Where W_{hx} represents the weight between the input and recurrent hidden nodes, W_{hh} represents the weight between the recurrent node and the previous time step of the hidden node itself, b and σ represent bias and non-linear (sigmoid) activation respectively. Although RNNs perform better in time series prediction problems, they still have issues yet to be addressed. For instance, from equation (4) above, the recurrent hidden state h_t approaches zero as the time interval increases, which leads to a diminishing gradient [20] problem. For this

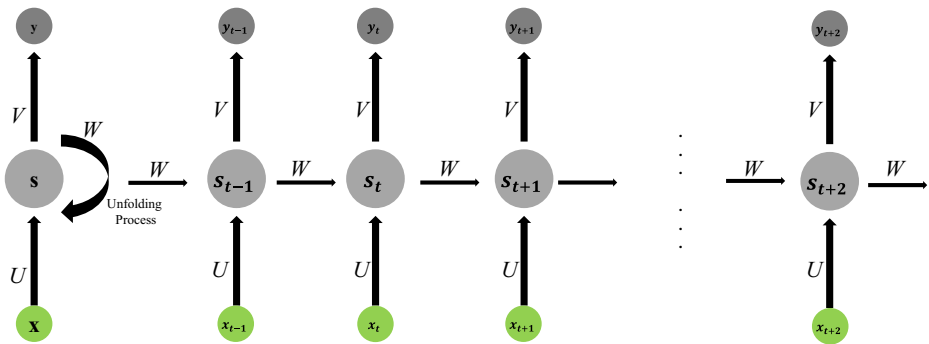


Figure 1 Structure of the Recurrent Neural Network (RNN)

reason, RNNs are unable to learn from time series having long time lags. This was resolved by the work of German engineers Hochreiter and Schmidhuber [19] – the Long Short-Term Memory RNN – which had the primary objective of modelling long-term time dependencies in time series. The LSTM model replaced the recurrent hidden unit with a ‘memory cell’.

The architecture of the LSTM-NN having one memory block is depicted in Figure 2. The memory block contains input, output, and forget gates, which respectively carry out to write, read, and reset functions on each cell. The multiplicative gates, \oplus and \otimes , refer to the matrix addition and dot product operators respectively and allow the model to store information over long periods, thereby eliminating the vanishing gradient problem commonly observed in traditional neural network models [19].

In an LSTM model, the predicted output of the next time step of a time-series input sequence $x = x_1 + x_2 + x_3 \dots x_t$ and output sequence of $y = y_1 + y_2 + y_3 \dots y_t$, is computed using the historical information supplied, without being told how many backward time steps to be traced. This is achieved using the following set of equations:

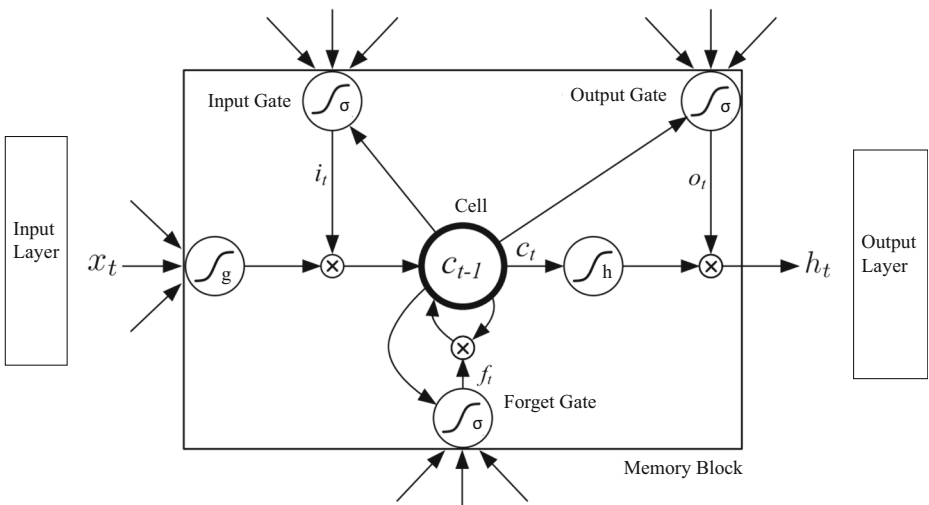


Figure 2 An LSTM model having one memory block

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \quad (5)$$

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \quad (6)$$

$$c_t = f_t c_{t-1} + i_t g(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \quad (7)$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \quad (8)$$

$$h_t = o_t h(c_t) \quad (9)$$

Where W and b represent the weight matrix and bias vector respectively and $\sigma(\cdot)$ denotes a standard logistic sigmoid function defined as:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (10)$$

$$g(x) = \frac{4}{1 + e^{-x}} - 2 \quad (11)$$

$$h(x) = \frac{2}{1 + e^{-x}} - 1 \quad (12)$$

Where $g(\cdot)$ and $h(\cdot)$ are the respective transformations of the sigmoid function above. The variables $i, f, o,$ and c are the input gate, forget gate, output gate, and cell activation vector respectively.

LSTM deep neural networks have been widely adopted in traffic prediction studies, for instance in [28], where an LSTM-NN model was used for traffic speed prediction, and the results compared to other non-parametric algorithms (Support Vector Machines (SVMs), Kalman Filter, and ARIMA). The findings showed superiority in the prediction accuracy of the LSTM model. Furthermore, [23] presented an LSTM and deep belief network (DBN) deep learning model to predict short-term traffic speed using traffic and rainfall data in Beijing, China. The results of the experiment revealed that fusing weather and traffic data sources improved the prediction performance of the models.

2.2 Traffic prediction using twitter information

Although Twitter has become a popular social media platform, there are still opportunities for employing data from its massive user base towards improving traffic prediction. [21] presented a study about social interactions on Twitter, which revealed that the driving process for its widespread adoption could be attributed to the fact that it represents a hidden network, with most of the messages describing meaningless interactions. This constitutes an essential antecedent for the skepticism – evident in the sparse number of studies – observed in the

literature about traffic prediction using twitter messages. Due to the open-source, public nature of the Twitter platform, data obtained from tweets can be subjective, context-specific, contain nuances, or statements that are intended to express sarcasm or irony.

Despite these drawbacks, many studies have attempted to incorporate tweet information in traffic prediction model training. For instance, [1] presented a Kalman Filter model trained using integrated twitter traffic information and traffic data in order to predict public vehicle arrival time. The study utilised real-time tweets related to road traffic information and performs semantic analysis on the retrieved dataset. The findings of the results showed remarkable improvement when compared to traffic-only data sources. Similarly, [15] presented a deep crowd flow prediction model trained using semantically-mined twitter traffic datasets. The study adopted an existing crowd flow prediction model – Spatiotemporal Residual Network (ST-ResNet), presented in [44] – as the baseline model for comparison. The end-to-end prediction model was configured to take tweets as additional inputs towards model training in order to predict future traffic crowd flows in an urban environment. The findings from the study reveal a positive correlation with tweet data and traffic flow, which resulted in improved prediction accuracy in comparison to the baseline model. Likewise, [41] presented a decision tree model for prediction of road traffic congestion severity and tested the model on real-time traffic networks in Bangkok, Thailand. The C4.5 decision tree-based model accepted tweets from various road-traffic broadcasting twitter user accounts and correlated the same to large crowds in certain areas. The findings of the study revealed that the inclusion of tweet road-traffic information improved the predictive model performance. Although the above listed studies used Twitter-obtained datasets for traffic prediction, they however, used tweet analysed data as the only non-traffic input dataset for the model training. Although the findings showed improvements in prediction accuracy, we, however, empirically show that the inclusion of weather and tweet data in addition to traffic datasets would yield a significant improvement in prediction accuracy of the deep learning predictive model.

3 Methodology

This section introduces basic concepts in neural networks and their use as autoencoders. We begin with a description of autoencoders, including their underlying logic and functions, before concluding with the proposed deep bi-directional LSTM prediction model. The proposed model is an enhanced version of the one presented in [11], where we employed stacked autoencoders for model training in addition to tweet messages being included as additional inputs to the model training dataset. The main advantage of an autoencoder is to learn a compressed representation (encoding) of a set of input data vectors. In other words, the autoencoder, which doubles as a dimensionality reduction technique in time series data and a data compression tool in image analysis, can be trained using very large datasets in a short time [16]. For this reason, high dimensional and large data can be processed in significantly shorter timeframes using LSTM autoencoders than ordinary deep LSTM neural networks.

3.1 Autoencoders

An autoencoder is a feedforward neural network that takes an input vector x and transforms it into a hidden representation or ‘latent’ space h . In other words, autoencoders compress the

input vector into a lower-dimensional ‘code’ and attempt to reconstruct the output from this given representation. The autoencoder consists of three major components: the encoder, the code, and the decoder. The input vector transformation, or encoder function, is achieved using equation (13):

$$h = \sigma(Wx + b) \quad (13)$$

Where W and b represent the hidden weight and bias components respectively, and σ represents the sigmoid function. The decoder transforms the resulting hidden representation h into a reconstructed feature space y using the following equation:

$$y = \sigma(W'h + b') \quad (14)$$

Where W' and b' denotes the weight and bias respectively. The stacked autoencoder is a stack of autoencoders and, just like autoencoders, learns in an unsupervised manner. The learning process involves layer-wise training in order to minimise the error between input and output vectors. The subsequent layer of the autoencoder is the hidden layer of the previous one, with each of the layers trained by gradient descent algorithm using an optimisation function, which is the squared reconstruction error J of the individual autoencoder layer. This is described in (15).

$$\operatorname{argmin}_{W_1, b_1, W_2, b_2} [J] = \operatorname{argmin}_{W_1, b_1, W_2, b_2} \left[\frac{\sum_{i=1}^m \|x_i - x'_i\| + J_{wd} + J_{sp}}{2} \right] \quad (15)$$

Where J represents the squared reconstruction error of the single autoencoder layer, x_i and x'_i respectively represent the i th value of the input vector as well as its corresponding reconstructed version. m represents the training dataset size, corresponding to the length of the input time series.

3.2 Deep bi-directional LSTMs

A deep LSTM network is a recurrent network that has many (hence the term deep) layers. It replaces conventional recurrent units with LSTM memory cells. The particular advantage of applying deep LSTM networks is the fact that it can learn long-term dependencies in complex data structures hierarchically [16]. When compared to single or double-layer (i.e. shallow learning/networks) LSTM networks, deep LSTMs can hierarchically extract temporal dependencies in complex time series or sequential datasets [18].

The structure of a bi-directional LSTM is one that is composed of two unidirectional LSTMs – stacked in opposite directions. Therefore, both previous and future vectors of the time series are applied in a bi-directional LSTM training cycle. In this way, the data is processed in both directions using two separate hidden layers, which are then fed forward to a single output layer. Figure 3 shows the structure of a bi-directional LSTM. As can be seen, the network computes both the forward hidden sequence \vec{h} and the backward hidden sequence h . The output is then computed by iterating the backward layer in reverse chronological order (i.e. from $t = T$ to 1), while the forward layer is iterated from $t = 1$ to T . A deep bi-directional LSTM is, therefore, a deep bi-directional LSTM network, which is a critical component of the success recorded in deep learning architectures. As previously stated, deep learning networks can hierarchically build-up layer representations in complex datasets. Deep bi-directional LSTMs are created by stacking multiple layers of bi-directional LSTMs

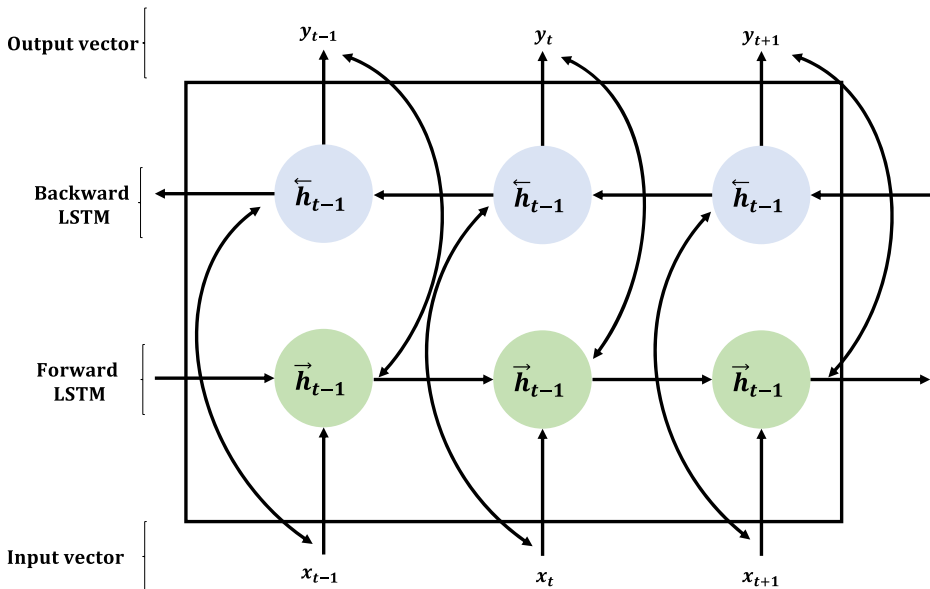


Figure 3 The architecture of a bi-directional LSTM

vertically. In this way, the output sequence for one layer serves as the input sequence for the next layer.

3.3 Data fusion

Data fusion refers to a multi-layered process for handling the automatic detection, integration, prediction, and combination of data from several sources. According to [4], five main categories of data fusion techniques exist: (i) Data in-data out (DAI-DAO), (ii) Data in-feature out (DAIFEO), (iii) Feature in-feature out (FEI-FEO), (iv) Feature indecision out (FEI-DEO), and (v) Decision in-decision out (DEIDEO). Within this study, we adopted the DAI-DAO data fusion technique, similar to the one presented in [11], where the combination of traffic and weather information was utilised. As stated, the advantage of fusing data at this level leads to more reliable outputs, as errors that can be introduced during fusion at the feature or prediction/decision level are avoided.

3.4 Deep bi-directional LSTM model

Figure 4 depicts the overview of the architecture of the model proposed in this paper. The model aims to predict the next twelve time steps of the traffic flow at time t (twelve 5-min predictions, which is the predicted traffic flow up to the next hour). Our approach towards traffic prediction follows the use of bi-directional LSTMs to learn in an unsupervised manner. The model comprises four main elements: the encoder, repeat vector, decoder, and fully connected (FC) layers respectively.

The first component represents the input layer, which takes in the input vector, comprising traffic, weather, and tweet data as a 2-dimensional $m \times n$ vector, where m represents the number of samples in the training dataset, and n represents the number of features (5 in this case). The second set of

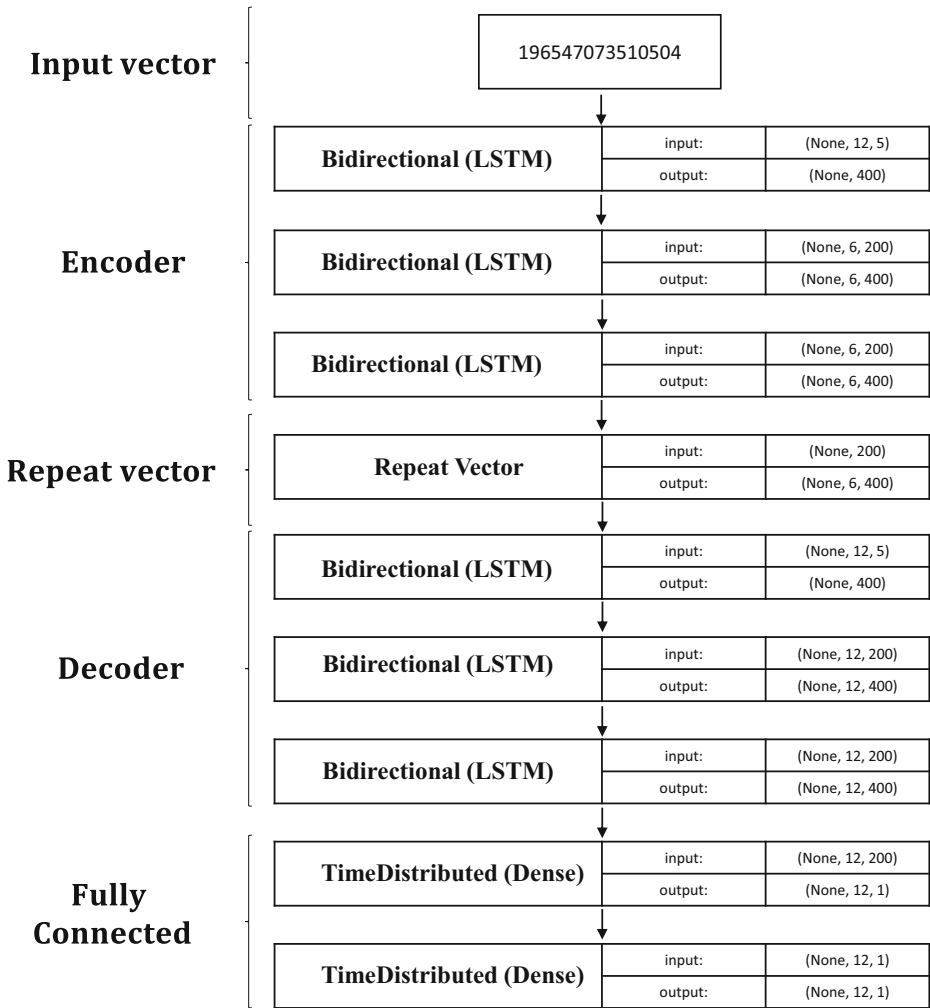


Figure 4 Overall model architecture with tweet-inclusive dataset

layers comprise bi-directional LSTM layers, which together form the encoder layer. The stack of bi-directional LSTMs reads in the input sequence of vectors. After the last sequence has been read, a repeat vector layer is applied, which –as the name implies – repeats the sequence of vectors to be reproduced by the encoder layers. The decoder layer then takes over the sequence from the repeat vector layer and outputs a prediction as a single row vector sequence. This is then passed on to a FC layer, where the target sequence is predicted. The concept of LSTM autoencoders is outside the scope of this study, but we refer to the work presented in [38] for more details.

4 Data description

For the experiments recorded in this paper, the datasets were provided by the Transport for Greater Manchester (TfGM). The weather data obtained during the study period comprised

hourly observations of temperature (Celsius) and precipitation (measured in millimeters). The tweets data was obtained from two (2) road-traffic information twitter user accounts. The accounts are the official Twitter handle of TfGM (@OfficialTfGM) and the Waze Manchester (@WazeTrafficMAN) respectively. Details about the dataset preparation are described in the following subsections.

4.1 Traffic dataset

The dataset comprised historical 5-min observations of traffic flow characteristics (speed, flow and density), collected using inductive loop detectors. The study area has ten traffic sensors, each of which is 0.3 miles apart. The study period spans from 1 April 2016 to 16 April 2017. The study area is an arterial road – Chester Road (A56) – in Stretford, Greater Manchester, UK, which lies within the coordinates of longitude and latitude between (53.46281, -2.28398) and (53.43822, -2.31394), as depicted by the pinpoint markers on the map in Figure 5. This study area was chosen as it represents an ideal test bed due to serving as one of the two main roads that allow access from South Manchester residential areas to the Manchester city centre. In addition, the road serves as the main arterial for the Manchester United Football Stadium – Old Trafford – as well as to other leisure points like shopping malls, clubs, restaurants, etc. As such, this road always has heavy traffic during rush hour (i.e. when travellers leave for and return from work), football games, weekends (due to the shopping mall and other attraction centres).

4.2 Weather data

For this study, the weather data was obtained from the Centre for Atmospheric Studies (CAS) at the University of Manchester. The dataset comprised hourly observations of precipitation (in millimetres) and temperature in degree Celsius for the same study period as stated in the previous subsection. However, given that the speed data comprised 5-min traffic parameters, in order to merge the weather data, the hourly data was estimated to be the same for each of the minutes that make up the hour. Although this might be a limitation in that there is a tendency of losing some rich information, it represents the better option when compared to aggregating the traffic and tweet data into hourly observations. Moreover, estimating temperature to remain the same for an hour does not represent a significant loss in information, as one can argue that the temperature does not significantly change within the hour. Even popular weather information mobile apps, such as the Weather Channel,⁵ WeatherProHD,⁶ and Yahoo! Weather,⁷ all present weather information in hourly observations.

4.3 Tweets data

For this study, python scripts were written to collect tweets from Twitter using the Twitter Streaming API⁸ and *tweepy*⁹ package. In order to ensure that the data obtained

⁵ Can be found at <https://weather.com/apps>

⁶ Can be found at <http://www.weatherpro.eu/home.html>

⁷ Can be found at <http://uk.mobile.yahoo.com/weather>

⁸ Can be found at <https://developer.twitter.com/>

⁹ Can be found at <https://www.tweepy.org>

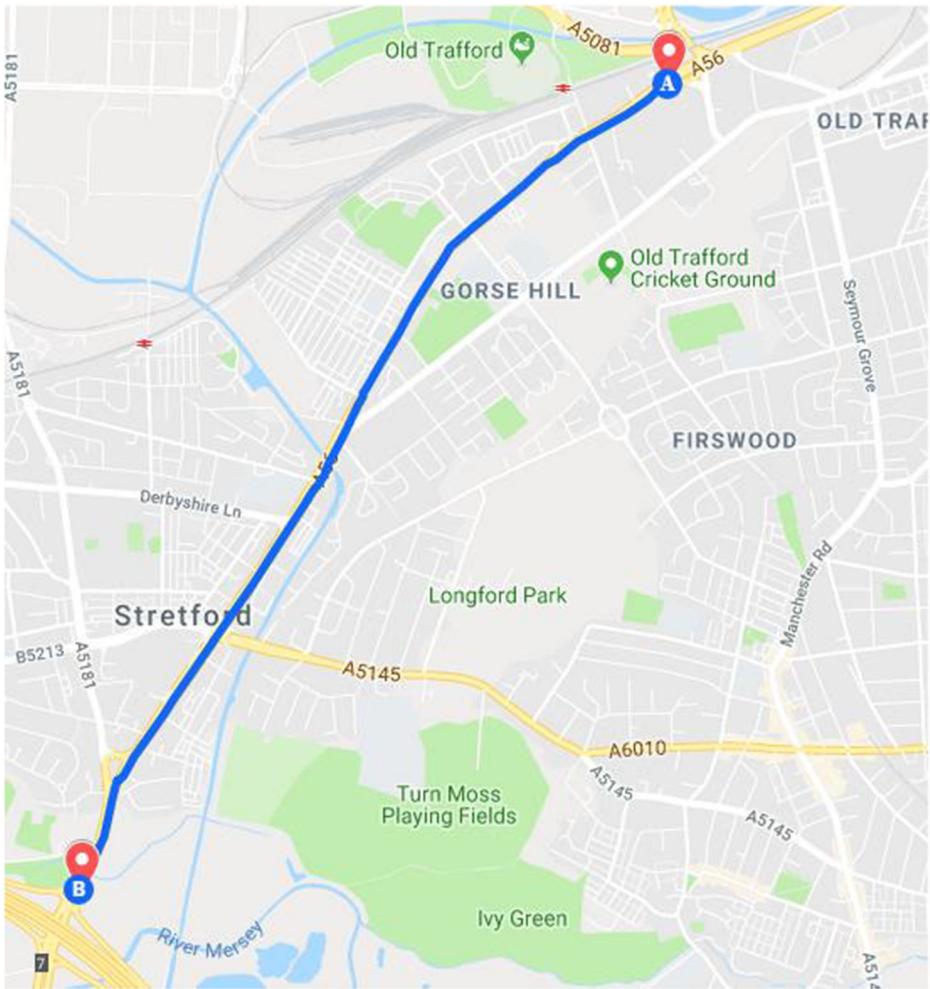


Figure 5 Study Area

from twitter were authentic and geo-specific, we opted to only use tweets from road-traffic information users – TfGM and Waze. The first script was written to extract all tweets from the two user accounts. In this way, we passed the Twitter user account IDs of *@OfficialTfGM* and *@WazeTrafficMAN* as *follow* parameter in the API. This step resulted in 102, 675. Second, we filtered the result set to only include tweets about the given road segment under consideration (A56 Chester Road). In this way, the keywords of 'A56', and 'Chester Road' were used as the track parameters to the endpoint in the python script. This resulted in a dataset comprising 9, 275 tweets from both users. The fourth step included extracting the timestamp from each of the tweets in order to get its date-time format (i.e. *dd-mm-yyyy hh:mm*). The final step of the script involved merging the timestamp from the tweets with the merged traffic and weather dataset. The merging process involved encoding the dataset at the respective timestamps of the tweets as 1 to reflect a tweet about the traffic condition, or 0 where

no tweets were recorded. The combined dataset contained 109,728 observations of 5-min aggregated traffic flow, speed, rainfall, temperature, and tweet variables. The dataset was then split using a train-test ratio of 70:30. A summary of the descriptive statistics for the dataset used in this study is presented in Table 1.

5 Experimental setup

This study adopted an overlapping sliding window approach for reconstructing the input multivariate time series dataset into a supervised learning format, similar to the procedure described in [11]. Thus, the predictive model was developed to learn the features in the historical time series dataset in order to make multi-step 1-h-ahead (12 steps of 5-min prediction horizons) traffic flow prediction, having been trained using the aggregated dataset presented in Table 1.

5.1 Model description

The proposed framework incorporates an eight-layer bi-directional LSTM stacked autoencoder architecture. For all of the inter-connected layers (except the output layer), the activation function used was the Rectified Linear Unit (ReLU), which introduced non-linearity to the learning process. The performance of deep learning networks is dependent on critical parameters that must be predetermined using a process known as *hyperparameter optimisation* or *hyper-parameterisation*. For this study, in order to identify the optimal set of hyperparameters, we applied a grid search framework. This presented a reproducible and flexible method for arriving at the optimum set of parameters. The overall prediction algorithm is presented in Algorithm 1. We applied an enhanced deep bi-directional LSTM traffic prediction model from the one presented in [11] to predict traffic flow 12 time steps ahead (i.e. 12 time steps of 5-min prediction horizons, equalling the traffic in the next hour). The model used in [11] was urban traffic speed prediction model trained using traffic and weather data. Therefore, we adopted the following deep learning methodology for this study (see Table 2).

5.2 Model performance evaluation

For this study, we adopted a prediction evaluation technique referred to as *walk-forward* validation or *backtesting*. Conventional evaluation methods like *k*-fold cross-validation are unsuitable for use in time series data because they do not consider the temporal or sequential

Table 1 Descriptive Statistics of Dataset

Statistic	Temp (°C)	Rainfall (mm)	Flow (veh/h)	Density (veh/mi)	Speed (mph)	Tweet Count
Mean	11.4477	0.0966	60.6928	26.8031	25.9123	0.0081
St. Dev.	5.2988	0.4963	41.3550	24.6022	4.1803	0.0899
Variance	28.0776	0.2463	1710.22	20,783.08	25.135	0.0080
Minimum	-4.6558	0	0	2	14.6	0
Maximum	32.6163	29.329	186	193.7	35.6	1
Count	109,728	109,728	109,728	109,728	109,728	109,728

order/dimension of the input dataset. We apply three statistical prediction accuracy evaluation metrics – Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Symmetrical Mean Absolute Percentage Error (sMAPE), which are defined by the equations below.

$$MAE = \frac{1}{n} \sum_{i=1}^n |e_i| \tag{16}$$

Algorithm 1 Speed Prediction using LSTM and Data Fusion

Input: observed sequences $X = x_1, x_2, x_3, \dots, x_T$

Output: predicted traffic speed sequence $Y = y_1, y_2, y_3, \dots, y_T$

1. Split X into training and testing data with 70:30 ratio
 2. In training data, select look back steps/batch size b , and at time t , generate lookback observations as x_1, x_2, \dots, x_b as input, and single-step ahead x_{b+1} as predicted value y_t .
 3. Set random initialisation for model parameters weight w_t and bias c .
 4. Train model using forward greedy layer-wise algorithm using bi-directional processing, and update model parameters
 5. Update model learning by backpropagation algorithm using optimiser (Adam in this case), minimising loss function $L = \frac{1}{n} \sum_{i=1}^n |y^i - \hat{y}^i|$, where \hat{y}^i is the predicted sequence in the i^{th} layer.
 6. Use testing data for model test prediction, while another batch of training data is used for another retraining process
 7. Reiterate until training set is exhausted
 8. Return prediction output sequence Y .
-

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n |e_i^2|} \tag{17}$$

$$sMAPE = \frac{200}{n} \sum_{i=1}^n \left| \frac{x_i - y_i}{x_i + y_i} \right| \tag{18}$$

Where $e_i, i = 1, 2, \dots, n$ represents n samples of modal errors, x_i and y_i respectively represent the input and output values.

5.3 Baseline models

We compared the performance of the proposed model against selected state-of-the-art baseline machine learning models. Using the performance evaluation metrics described in the preceding section, we compared the proposed model to the following baseline models. (1) Support Vector Regressor [32], (2) Extreme Gradient Boosting (xGBoost) [6], and RandomForest regressor [27]. For each of the baseline models, the identical training dataset was used to ensure fairness and objectivity in the model evaluation process.

Table 2 Experimental methodology for urban traffic flow prediction

Stage	Description	Output
1	Data wrangling and pre-processing, gathering and cleaning of traffic, weather, and tweet datasets. Separation into train-test datasets	Training and testing datasets respectively
2	Data fusion stage using DAI-DAO approach	Merged dataset
3	Train deep bi-directional LSTM network using traffic-only dataset	Output vector
4	Train deep bi-directional LSTM network using traffic and weather dataset	Output vector
5	Train deep bi-directional LSTM network using traffic, weather, and tweet datasets	Output vector

5.4 Implementation environment

The experiment environment used for this study was on a single GPU node with Intel® Xeon® E-2146G CPU @ 3.50GHz, 32-GB Memory, and NVIDIA Tesla V100-PCIE 16GB GPU. The GPU is used for accelerated model training due to significant computation demand in deep learning models. The development was performed using Python 3.6.8, R version 3.5.1, and TensorFlow 1.12.0.

6 Results

The test dataset used for this study spans from 23 December 2016 to 16 April 2017. Table 3 shows the performance evaluation results of the proposed and baseline models. As the table shows, there is a significant improvement in the predictive accuracy when tweet messages, weather, and traffic datasets are utilized (i.e. the shaded row). It can also be seen that the model proposed in this paper outperformed the traditional machine learning baselines, however with a higher training time. Figure 6 presents bar plots of the MAE of the respective models trained using the three (3) input data combinations. As can be seen, the models trained using tweet, weather, and traffic data recorded the lowest MAE. Similarly, from Table 3, it can be seen that the SVR model performed competitively in comparison to the results from the proposed model. The boosting algorithm returned the worst set of results, however, with the shortest training time. In summary, the results clearly demonstrate a significant improvement (reduced error) when compared to the model trained using traffic and weather datasets only. Although the traffic-only models had the shortest training time, they, however, resulted in higher predictive errors (MAE, RMSE and sMAPE). This enables the conclusion that the inclusion of tweet analysis significantly improves the prediction accuracy when trained on combined traffic, weather and tweet data sources.

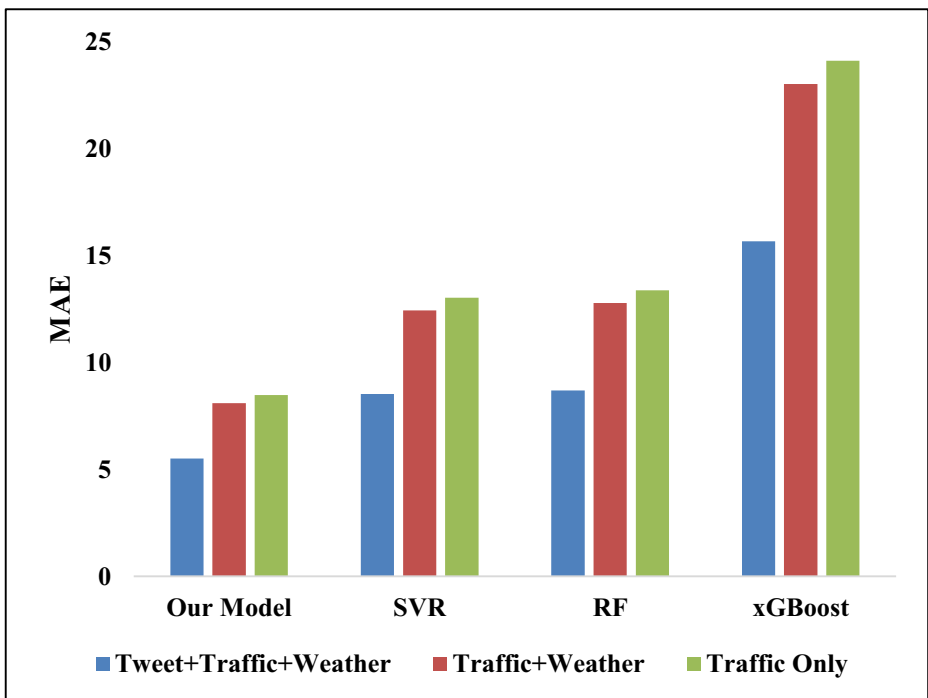
Although the gain in terms of accuracy improvement observed may appear to be marginal, it is worth mentioning here that the cost benefit realized by a reduction in the MAE (i.e. from 8 veh/h using traffic and weather datasets to 5.5 veh/h with the tweet-inclusive dataset) is significant. For instance, if the time loss of road users can be quantified and costed, as can the cost of the impact of congestion on the environment due to emissions and sound pollution, then realising an improvement in accuracy in traffic prediction need not be rationalised. Furthermore, improvement in accuracy can directly equate to financial/cost savings for supply chain/logistics companies, such as Amazon, DHL, etc. Therefore, a seemingly insignificant or marginal reduction in prediction error can easily result in millions of pounds' worth of savings, particularly given that the predictive model in question considers urban traffic. In addition, more accurate traffic predictions can lead to less frustration on behalf of drivers, since this is a very busy road, especially during rush hours. For instance, assume there is an accident on this road. Traffic management personnel will have to use an accurate traffic prediction and make appropriate decisions based on that, i.e. do nothing, divert traffic to a nearby road, open a bus lane if available or change frequency of traffic lights if available as an option. If the prediction is inaccurate and this leads to a wrong decision, then this will cause more frustration to drivers, more delays for deliveries, more noise/pollution for the environment.

Figure 7 shows scatter plots of the predicted and ground truth (actual) observations for the three prediction models as described in stages 3, 4 and 5 of Table 2. The x -axes in the scatter plots represents the actual (ground truth) traffic flow values, while the y -axes represent the predicted values. The sub-chart titles represent the R^2 values of the prediction models, which can be inferred from Table 3.

Table 3 Performance Evaluation

Model	Data Used	MAE	RMSE	sMAPE	R ²	Training Time (s)
Proposed model	Traffic Weather Tweet	5.5049	6.8579	0.1743	93.45%	583
SVR [32]		8.5236	11.444	0.2909	83.62%	496
RandomForest [27]		8.6830	11.633	0.296	80.93%	276
xGBoost [6]		15.650	20.301	0.5160	81.16%	78
Proposed model	Traffic Weather	8.0927	10.4178	0.2282	85.25%	599
SVR [32]		12.429	17.3845	0.3787	78.28%	366
RandomForest [27]		12.765	17.6680	0.3875	73.83%	189
xGBoost [6]		23.007	30.8391	0.6756	74.04%	43
Proposed model	Traffic	9.4763	13.2289	0.2294	82.25%	571
SVR [32]		13.018	19.0754	0.3898	75.53%	219
RandomForest [27]		13.370	22.4355	0.3895	73.47%	77
xGBoost [6]		24.098	39.1606	0.6790	71.43%	16

The bottom-right graphic in Figure 7 represents the multi-plot of the predicted and actual values for the respective models on a particular period in which there was a lane closure in the study area. The lane closure, which lasted for four days, was due to a road construction work in the study area. The y -axis in the bottom-right graphic in Figure 7 represents the traffic flow (vehicles/h), while the x -axis represents the 5-min time steps for the four days (48 time steps) when the lane closure was in effect. As can be seen from the graphic, the prediction model trained using traffic only and traffic & weather datasets were less accurate than the model

**Figure 6** MAE for predictive models using various data fusions

trained with the integrated tweet dataset. However, the model that was trained with the tweet dataset performed best in the prediction, because several tweets were broadcasted from the Twitter accounts, which provided improved accuracies of the predictive model.

Similarly, Figure 8 shows the respective predictive performances for the first 400 time steps of the models trained using datasets of tweet + traffic + weather, respectively. For each of the subplots, the *x*-axis represents the time steps, while the *y*-axis represents the value of the flow (veh/h) (for the upper half of each plot) and the absolute error between the predicted and actual values (for the lower part of the subplot) respectively. From the graphic in Figure 8, it can be seen that the proposed model accurately captured the temporal patterns and significantly outperformed the machine learning state-of-the-art benchmark models. As the graphic shows, the proposed model performed best, with the SVR model being second. When these two models are compared in terms of computation time, it can be seen that the proposed model performs competitively (i.e. 583 s vs. 496 s or 17% increase in time) when compared to a reduction in MAE of about 35.4% (i.e. from 8.5 veh/h to 5.5 veh/h).

In order to test for statistical significance between the sets of results obtained, we performed analysis of variance (ANOVA) as it estimates the variance within groups (i.e. variation/error in the raw data) and between groups (i.e. the result of the effects of the experiments). In this section, we compared the result of the prediction developed from training dataset Tweet + Weather + Traffic, Weather + Traffic, and Traffic alone using ANOVA. The following hypothesis was tested using ANOVA.

$$H_0 : y = \hat{y}_{tw+tr+we} = \hat{y}_{tr+we} = \hat{y}_{tr} \tag{19}$$

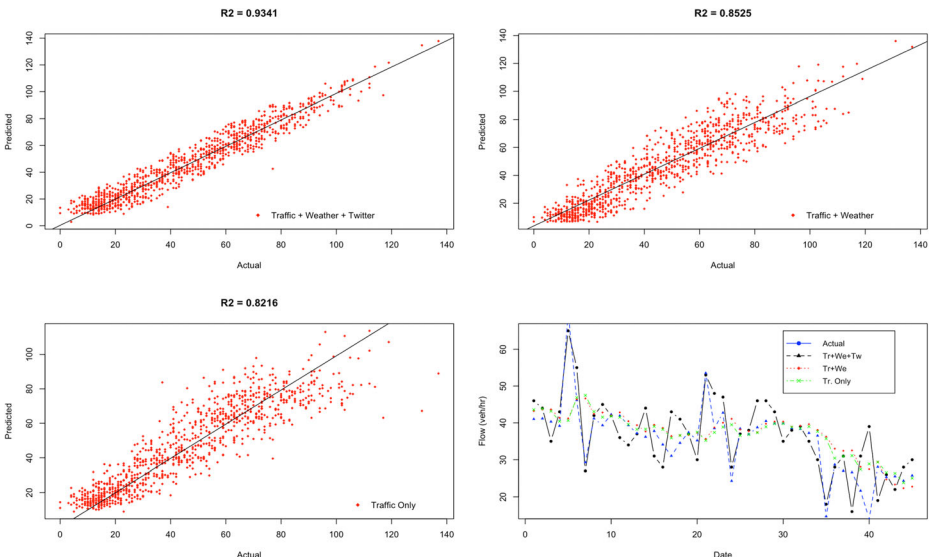


Figure 7 Performance evaluation of the proposed model. The *x*-axis represents the actual (ground truth) values of traffic flow, while the *y*-axis represents the predicted values. The bottom-right graphic represents a multi-line plot for the predicted values vs ground truth

Where y , $\hat{y}_{tw+tr+we}$, \hat{y}_{tr} and \hat{y}_{tr+we} represented the average estimation obtained from actual, model using tweet + traffic + weather, traffic only, and traffic + weather, respectively. The hypothesis statement was tested using the level of significance assumed as 0.05 ($p < 0.05$). Table 4 displays the single factor ANOVA results for the groups of predictive values for the respective training regimes (i.e. using Tweet + weather + traffic dataset, Traffic + Weather and Traffic only datasets respectively). Given that F is greater than F_{crit} , then the null hypothesis is rejected, meaning that the corresponding results are significantly different. As can be seen from the table, there is a significant difference ($p < 0.05$) between the groups, therefore, the null hypothesis is rejected, given that $F_{crit} = 2.6072$ and $F = 5.3927$, thus $F > F_{crit}$. In addition, the p value between groups displayed statistical significance, supporting the claim to reject the null hypothesis, since $p = 0.001058 < 0.05$.

7 Conclusions and future work

In this paper, we have presented an urban traffic flow prediction model that explored the effectiveness of integrating rich information obtained from tweet data for urban traffic prediction augmenting an existing urban traffic prediction model based on weather and traffic datasets. The baseline model, proposed in [11], employs a deep bi-directional LSTM

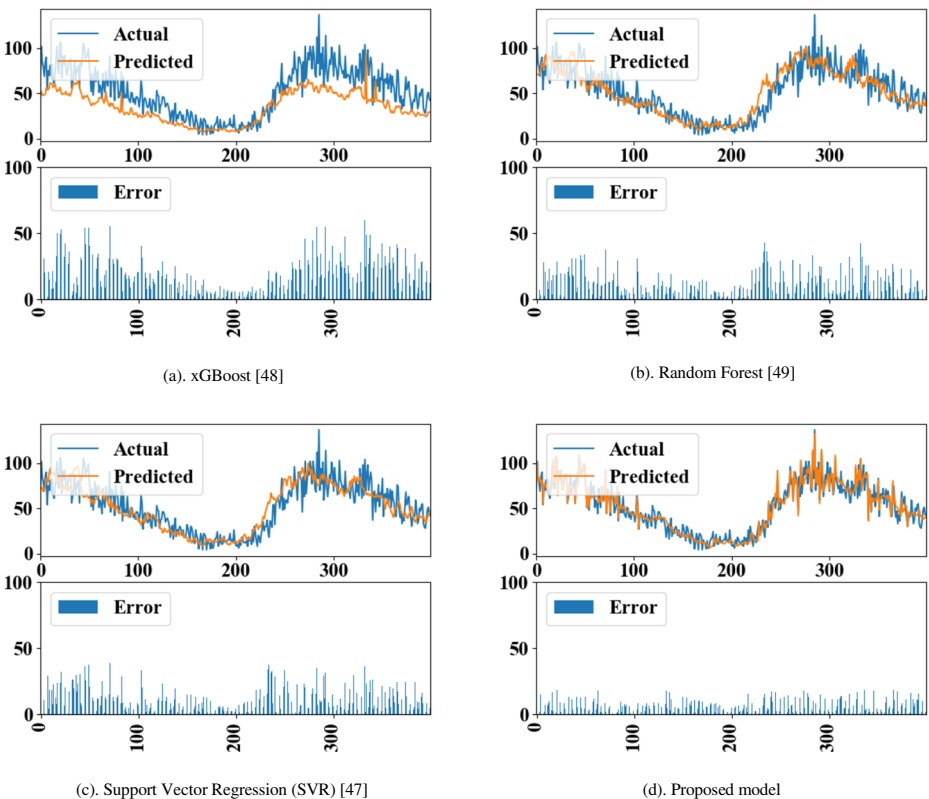


Fig. 8 Predictive performance of models on tweet + traffic + weather dataset

Table 4 Single Factor ANOVA for groups of datasets used

Groups	Count	Sum	Average	Variance		
Actual	987	45,191.66	45.7869	716.279		
Traffic + Tweet + Weather	987	41,905.30	42.4572	1.2E-27		
Traffic + Weather	987	45,081.67	45.6754	703.496		
Traffic only	987	45,486.00	46.0851	707.816		
Source of Variation	SS	df	MS	F	p value	F_{crit}
Between Groups	8605.1315	3	2868.38	5.392723	1.058E-3	2.60716
Within Groups	2,097,804.6	3944	531.898			
Total	2,106,409.6	3947				

architecture incorporating traffic flow parameters as well as rainfall and temperature. The augmented model presented in this paper employed a bi-directional LSTM autoencoder approach that accepts geo-specific tweets from road-traffic information Twitter accounts - @OfficialTfGM and @WazeTrafficMAN as additional non-traffic input data. The results from the empirical analysis showed that the inclusion of tweet data, in addition to traffic, rainfall, and temperature datasets reduced the MAE from 8 to 5.5 veh/h, providing a more accurate traffic flow prediction model. The model was tested using historical traffic, weather, and tweets datasets on an urban arterial road (Chester Road – A56) in Greater Manchester, UK.

The work presented in this study was restricted to a single arterial road in Greater Manchester, UK. Although this is a limitation, using the model as-is to account for additional urban roads in big cities can reuse the proposed model architecture with some additional data features. For instance, the dataset, tweet location, and possibly the twitter sources (i.e. TfGM for Manchester, TfL for London, etc.) can be modified to cater for the various locations under consideration, whilst retaining the model architecture and composition. On the other hand, it may be possible to consider the correlation between roads (i.e. how does an accident on road A affect road B?) with some additional data features. However, the extension of this model to accommodate a very wide geographical area presents considerable complexity challenges with corresponding increases in model computational resources and training time. Larger geographical areas involve additional data and higher training time requirements, and, therefore, increase the likelihood for lower accuracy, especially due to less developed traffic-sensing devices in residential streets and roads. However, given that the study area considers one of the two main conduits into the Manchester city centre, which attracts heavy traffic due to businesses having their premises and houses the large Arndale Shopping centre, the proposed end-to-end model can be replicated to other major arterial roads and express ways, which will be helpful to effectively managing traffic congestion in major geographical regions such as Greater Manchester (which has a population of about 2.8 million). Future work will consider additional filtering of tweets and the inclusion of additional keywords. However, trade-offs will also have to be taken into account, for example, the semantic analysis of the tweets may improve model performance but also increases training and pre-processing time, as has been demonstrated in [35]. Secondly, in this paper, we only included tweet data from traffic authorities (i.e. TfGM and WazeTrafficMAN). There is the potential of including tweets from other road-traffic information Twitter accounts, for instance, tweets from Manchester United FC (on football days), or event companies.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Abidin, A.F., Kolberg, M., Hussain, A.: Integrating twitter traffic information with Kalman filter models for public transportation vehicle arrival time prediction. In: *big-data analytics and cloud computing*. Pp. 67–82. Springer international publishing, Cham (2015)
2. Agarwal, M., Maze, T., Souleyrette, R.: Impacts of weather on urban freeway traffic flow characteristics and facility capacity. In: *Proceedings of the 2005 mid-continent transportation research symposium*. pp. 18–19 (2005)
3. Bin, Y., Yang, Y., Shen, F., Xu, X., Shen, H.T.: Bidirectional long-short term memory for video description. In: *Proceedings of the 24th ACM international conference on Multimedia*. pp. 436–440. ACM (2016)
4. Castanedo, F.: A review of data fusion techniques. *Sci. World J.* (2013)
5. Dasgupta, B., Liu, D., Siegelmann, H.T.: Neural Networks. In: *Handbook of Approximation Algorithms and Metaheuristics* (2007)
6. Dong, X., Lei, T., Jin, S., Hou, Z.: Short-Term Traffic Flow Prediction Based on XGBoost. In: *2018 IEEE 7th Data Driven Control and Learning Systems Conference (DDCLS)*. pp. 854–859. IEEE (2018)
7. Elman, J.L.: Finding structure in time. *Cogn. Sci.* **14**, 179–211 (1990). [https://doi.org/10.1016/0364-0213\(90\)90002-E](https://doi.org/10.1016/0364-0213(90)90002-E)
8. Essien, A., Giannetti, C.: A deep learning framework for Univariate time series prediction using convolutional LSTM stacked autoencoders. In: *2019 IEEE international conference on INnovations in intelligent SysTems and applications (INISTA)*. Pp. 1–6. IEEE, Sofia (2019)
9. Essien, A., Petrounias, I., Sampaio, P., Sampaio, S.: The impact of rainfall and temperature on peak and off-peak urban traffic. In: *International Conference on Database and Expert Systems Applications*. pp. 399–407. Springer, Cham. (2018)
10. Essien, A., Petrounias, I., Sampaio, P., Sampaio, S.: Deep-PRESIMM: Integrating Deep Learning with Microsimulation for Traffic Prediction. In: *IEEE International Conference on Systems, Man, and Cybernetics*. pp. 1–6. IEEE Xplore (2019)
11. Essien, A., Petrounias, I., Sampaio, P., Sampaio, S.: Improving Urban Traffic Speed Prediction Using Data Source Fusion and Deep Learning. In: *2019 IEEE International Conference on Big Data and Smart Computing (BigComp)*. pp. 1–8. IEEE (2019)
12. Essien, A., Chukwukelu, G., Giannetti, C.: (2019) A Scalable Deep Convolutional LSTM Neural Network for Large-Scale Urban Traffic Flow Prediction using Recurrence Plots. In: *2019 IEEE AFRICON*. IEEE Xplore
13. Fernando, S., Sethu, V., Ambikairajah, E., Epps, J.: Bidirectional Modelling for Short Duration Language Identification. In: *INTERSPEECH*. pp. 2809–2813 (2017)
14. Giannetti, C., Essien, A., Pang, Y.O.: A novel deep learning approach for event detection in smart manufacturing. In: *CIE49 proceedings, Beijing* (2019)
15. Goh, G., Koh, J., Zhang, Y.: Twitter-Informed Crowd Flow Prediction. In: *2018 IEEE Conference on Data Mining Workshops (ICDMW)*. pp. 624–631 (2018)
16. Goodfellow, I., Bengio, Y.: *Deep learning*. MIT Press (2015)
17. He, J., Shen, W., Divakaruni, P., ... L.W.-...-T.I.J., 2013, undefined: Improving traffic prediction with tweet semantics. aaai.org
18. Hermans, M., Schrauwen, B.: Training and analysing deep recurrent neural networks. *Advances in Neural information processing systems*. 190–198 (2013)
19. Hochreiter, S., Schmidhuber, J.: Long Short-Term Memory. *Neural Computation*. **9**, 1735–1780 (1997). doi: <https://doi.org/10.1162/neco.1997.9.8.1735>. Long short-term memory
20. Hochreiter, S., Bengio, Y., Frasconi, P., Schmidhuber, J.: Gradient flow in recurrent nets: the difficulty of learning long-term dependencies, <https://www.bioinf.jku.at/publications/older/ch7.pdf>. (2001)
21. Huberman, B., Romero, D., Wu, F.: Social networks that matter: twitter under the microscope. *First Monday*. **14**, (2009)

22. Jia, Y., Wu, J., Xu, M.: Traffic flow prediction with rainfall impact using a deep learning method. *J. Adv. Transp.* **2017**, (2017). <https://doi.org/10.1155/2017/6575947>
23. Jia, Y., Wu, J., Ben-Akiva, M., Seshadri, R., Du, Y.: Rainfall-integrated traffic speed prediction using deep learning method. *IET Intell. Transp. Syst.* **11**, 531–536 (2017). <https://doi.org/10.1049/iet-its.2016.0257>
24. Kaastra, I., Boyd, M.: Designing a neural network for forecasting financial and economic time series. *Neurocomputing.* **10**, 215–236 (1996)
25. Kim, S., Hong, S., Joh, M., Song, S.: DeepRain: ConvLSTM network for precipitation prediction using multichannel radar data. *ArXiv preprint.* 1711., (2017)
26. Lana, I., Del-Ser, J., Velez, M., Vlahogianni, E.I.: Road traffic forecasting: recent advances and new challenges. *IEEE Intell. Transp. Syst. Mag.* **10**, 93–109 (2018). <https://doi.org/10.1109/IMITS.2018.2806634>
27. Liu, Y., Wu, H.: Prediction of Road Traffic Congestion Based on Random Forest. In: 2017 10th International Symposium on Computational Intelligence and Design (ISCID), pp. 361–364. IEEE (2017)
28. Ma, X., Tao, Z., Wang, Y., Yu, H., Wang, Y.: Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. *Transportation Research Part C: Emerging Technologies.* **54**, 187–197 (2015). <https://doi.org/10.1016/j.trc.2015.03.014>
29. Nguyen, T.H., Shirai, K., Velcin, J.: Sentiment analysis on social media for stock movement prediction. *Expert Syst. Appl.* **42**, 9603–9611 (2015). <https://doi.org/10.1016/j.eswa.2015.07.052>
30. Ni, M., He, Q., Gao, J.: Using social media to predict traffic flow under special event conditions. In: The 93rd annual meeting of transportation research board (2014)
31. Petrović, S., Osborne, M., Lavrenko, V.: Streaming first story detection with application to twitter. In: 2010 Annual Conference of the North American Chapter of the Association for Computer Linguistics, pp. 181–189. Association for Computational Linguistics (2010)
32. Philip, A.M., Ramadurai, G., Vanajakshi, L.: Urban arterial travel time prediction using support vector regression. *Transportation in Developing Economies.* **4**, 7 (2018)
33. Poonia, P., Jain, V.K., Kumar, A.: Short term traffic flow prediction methodologies: a review. *Mody University International Journal of Computing and Engineering Research.* **2**, 37–39 (2018)
34. Qiao, W., Haghani, A., Hamed, M.: A Nonparametric Model for Short-Term Travel Time Prediction Using Bluetooth Data. *Journal of Intelligent Transportation Systems: Technology, Planning, and Operations.* (2013). doi:<https://doi.org/10.1080/15472450.2012.748555>
35. Qin, Z., Petrounias, I.: A semantic-based framework for fine grained sentiment analysis. In: 2017 IEEE 19th Conference on Business Informatics (CBI), pp. 295–301. IEEE (2017)
36. Rodriguez, P., Wiles, J., Elman, J.L.: A recurrent neural network that learns to count. *Connect. Sci.* **11**, 5–40 (1999). <https://doi.org/10.1080/095400999116340>
37. Smith, B.L., Williams, B.M., Keith Oswald, R.: Comparison of parametric and nonparametric models for traffic flow forecasting. *Transportation Research Part C: Emerging Technologies.* **10**, 303–321 (2002). [https://doi.org/10.1016/S0968-090X\(02\)00009-8](https://doi.org/10.1016/S0968-090X(02)00009-8)
38. Srivastava, N., Mansimov, E., Salakhutdinov, R.: Unsupervised Learning of Video Representations using LSTMs. In: International Conference on Machine Learning, pp. 843–852 (2015)
39. Wang, X., Gerber, M.S., Brown, D.E.: Automatic crime prediction using events extracted from twitter posts. In: lecture notes in computer science (including subseries lecture notes in Artificial intelligence and lecture notes in bioinformatics). Pp. 231–238. Springer, Berlin (2012)
40. Wang, X., An, K., Tang, L., Chen, X.: Short term prediction of freeway exiting volume based on SVM and KNN. *International Journal of Transportation Science and Technology.* (2015). <https://doi.org/10.1260/2046-0430.4.3.337>
41. Wongcharoen, S., Senivongse, T.: Twitter analysis of road traffic congestion severity estimation. In: 2016 13th International Joint Conference on Computer Science and Software Engineering, JCSSE 2016, pp. 1–6. IEEE (2016)
42. Yu, B., Song, X., Guan, F., Yang, Z., Yao, B.: k-Nearest Neighbor Model for Multiple-Time-Step Prediction of Short-Term Traffic Condition. *Journal of Transportation Engineering.* **142**, 04016018 (2016). [https://doi.org/10.1061/\(ASCE\)TE.1943-5436.0000816](https://doi.org/10.1061/(ASCE)TE.1943-5436.0000816)
43. Zhang, L., Liu, Q., Yang, W., Wei, N., Dong, D.: An improved k-nearest neighbor model for short-term traffic flow prediction. *Procedia-Social and Behavioural Sciences.* **96**, 653–662 (2013)
44. Zhang, J., Zheng, Y., Artificial, D.Q.: Deep Spatio-Temporal Residual Networks for Citywide Crowd Flows Prediction. In: Thirty-First AAAI Conference on Artificial Intelligence, pp. 1655–1661 (2017)

Affiliations

Aniekan Essien^{1,2} · **Ilias Petrounias**³ · **Pedro Sampaio**¹ · **Sandra Sampaio**¹

Ilias Petrounias
iliasprounias@icloud.com

Pedro Sampaio
pedro.sampaio@manchester.ac.uk

Sandra Sampaio
s.sampaio@manchester.ac.uk

¹ The University of Manchester, Oxford Road, Manchester, UK

² Future Manufacturing Research Institute (FMRI), College of Engineering, Swansea University, Fabian Way, Swansea, UK

³ UseMyData.com Ltd., London, UK