Deep-PRESIMM: Integrating Deep Learning with Microsimulation for Traffic Prediction

DOI:
10.1109/SMC.2019.8914604

Document Version
Accepted author manuscript

Citation for published version (APA):
Essien, A., Petrounias, I., Sampaio, P., & Sampaio, S. (2019). Deep-PRESIMM: Integrating Deep Learning with Microsimulation for Traffic Prediction. In *IEEE International Conference on Systems, Man, and Cybernetics (IEEE SMC 2019).* (pp. 4257). (2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)). IEEE Computer Society . https://doi.org/10.1109/SMC.2019.8914604

Published in:
IEEE International Conference on Systems, Man, and Cybernetics (IEEE SMC 2019).

Deep-PRESIMM: Integrating Deep Learning with Microsimulation for Traffic Prediction

Aniekan E. Essien, *Member, IEEE*, Ilias Petrounias, Pedro Sampaio, and Sandra Sampaio.

Abstract—**This paper presents a proactive model and tool for traffic analysis and management that integrates deep learning for traffic parameter prediction with microscopic traffic simulation providing traffic analysts with the ability to visualise the traffic network state ahead of time, generate traffic control measures, and visualise the consequences of the applied measure(s). The model adopts an integrated assess-forecast-simulate approach in which traffic flow characteristics are applied to deep Convolutional Neural Network and Long Short-Term Memory (CNN-LSTM) stacked autoencoders in order to forecast traffic flow and speed, which is subsequently passed on to a traffic microsimulation tool – Simulation of Urban Mobility (SUMO) – where the predicted parameters are used to generate a traffic future state simulation. We test our model using sensor-collected traffic and weather data from the geographical area of Greater Manchester, United Kingdom. The empirical results show the benefits of the model for urban traffic analysis.**

I. Introduction

Due to economic and demographic growth in urban areas, traffic congestion has reached critical levels. This is partly due to limitations in the ability to expand the physical capacities of road networks in big cities. As a result, advanced traffic management systems (ATMS) have been developed in the past few decades to efficiently manage the existing road network capacity. These systems aim to minimise the negative consequences of intense traffic in urban areas, such as increases in air pollution levels, road network congestion, and noise by providing traffic network managers with capabilities for traffic analysis and forecasting [1].

In this context, research in Intelligent Transportation Systems (ITS) and short-term forecasting has emerged, leveraging recent advances in deep learning prediction methods and techniques. There is a strong focus on traffic prediction research applied to offline systems using historical traffic data to make forecasts and assess the impact of the observed changes/control measures [2]–[5]. However, there is a dearth of studies that apply historical and real-time traffic data for traffic prediction, as well as provide decision-support capabilities via the visualisation of the forecasted traffic network state. Modelling approaches developed to provide such proactive traffic management combining traffic parameter forecasting, traffic state estimation, and subsequent visualization of the future traffic network state have been proposed, such as in [2], [6]–[8], but are restricted in their approach towards the future traffic state estimation during a particular time interval. For instance, the ability to provide an answer to the question: ‘if there is a lane closure on a motorway due to an accident, snow or flood, what will be the traffic situation 30 minutes from now?’ In other words, while it is possible to estimate/predict the future traffic state via a faster-than-real-time simulation run of the current traffic network state, it is essential to include a vast range of operational conditions that might affect the transport network, such as weather information, significant events such as football matches or musical concerts, road construction works, etc., which can significantly affect the traffic demand within an urban network. Changes in the conditions mentioned above affect commuters regularly, which in turn affects the traffic situation. For example, if there is an adverse weather warning, travellers are likely to cancel, modify their trips, or delay departure to avoid severe congestion [9]. The capability of an ITS or ATMS to respond in real-time to such disruptions is essential for effectively managing urban transport networks.

This paper proposes a proactive traffic management model that integrates deep learning and traffic microsimulation to provide decision-making support capabilities relating to predicting traffic state in future time-window intervals that can be used by traffic network analysts. The system is comprised of two distinct stages: (1) a prediction module that is made of deep CNN-LSTM stacked autoencoders trained using traffic and non-traffic data sources (temperature and precipitation), and (2) a traffic microsimulation stage via a simulation tool (SUMO), which allows for simulation of future traffic states in scenarios involving changes in conditions.

The contributions of this paper are summarized as follows: (1) a novel two-stage deep learning model for proactive traffic management that incorporates prediction and simulation/visualization, (2) an approach for traffic analysts to visualize the future status of the present traffic network state for each possible alternative action that can be taken, thereby guiding choices of alternative control measures to be applied based on 1-hour traffic prediction horizons. We test the model on real-world sensor-collected data from A56 (Chester Road), Stretford in Greater Manchester, United Kingdom. The chosen study area provides a good test bed for our proposed model as it serves as a conduit between a residential area and office complexes, and also houses Manchester United Football Stadium (Old Trafford).

A. Essien is with the University of Manchester, Manchester, UK. (e-mail: aniekan.essien@manchester.ac.uk). I. Petrounias was a Senior Lecturer with the University of Manchester, UK. (e-mail: iliaspetrounias@icloud.com). P. Sampaio is in the Management Sciences & Marketing (MSM) Division at the University of Manchester. (e-mail: pedro.sampaio@manchester.ac.uk). S. Sampaio is at the School of Computing, University of Manchester. (e-mail: s.sampaio@manchester.ac.uk).

The remainder of this paper is organised as follows. Section II presents a review of related studies, while details about the Deep-PRESIMM model and tool are presented in Section III. Section IV briefly describes the input data used, while Section V discusses the experimental setup. In Section VI, we present the case study and discuss the results and findings in Section VII. In Section VIII, we conclude the paper and outline future work.

II. Related Work

According to [1], the extant literature on short-term traffic forecasting is voluminous and has largely considered single data points, mainly employing univariate predictive models. Many methods and techniques, such as time series analysis, neural network models, simulation-based models, and Bayesian methods have been applied towards the development of traffic forecasting models, leveraging advances in algorithm design, data storage capacity and computational processing power. The advent of short-term prediction saw the application of statistical methods like Auto-Regressive Integrated Moving Average (ARIMA), which mainly ignored the spatial dependency of traffic evolution, until in the early 2000s [2]. The second generation of prediction methods saw the rise of non-linear predictive models such as neural networks, and kernel-based algorithms, which exposed the inherent vulnerabilities in classical predictive models. According to [3], parametric models such as ARIMA tend to assume linearity of the mean and variance of input data, which makes them show weaknesses in traffic data, which often contain peaks (during rush hour traffic and accidents/events).

Furthermore, simulation-based methods for traffic prediction have been explored in academic literature within the last two decades, such as in, [4]–[7]. The success of such approaches led to an increase in research towards simulation-based traffic prediction models. For instance, online traffic network simulation models have been integrated with real-time decision support systems for integrated corridor management (ICM) [8].

However, many studies involving simulation-based traffic prediction focus on traffic parameter prediction at an individual or microscopic level, rather than at the network (macroscopic) level [8]. Table I presents a summary of existing studies in simulation-based traffic prediction. As the table shows, only two models have decision-support functionalities. The two simulation-based real-time traffic management systems (DYNASMART-X [5] and DynaMIT [9]) that provide real-time traffic short-term prediction and have decision support capabilities for traffic management, however, sacrifice traffic prediction (i.e. network state estimation) accuracy for simulation latency, thereby ignoring non-traffic input factors capable of significantly affecting traffic state, such as rainfall and temperature [10]. Moreover, a limited effort is put into the development of real-time proactive simulation-based traffic systems with decision support capabilities that incorporate robust and accurate non-linear predictive algorithms, such as deep learning networks, due to the computational and data demands of such models.

The intent of providing a degree of proactive-ness involves creating accurate representations of the future traffic state, which is not adequately represented by a simulated current network state. We, therefore, propose a system that accurately captures the present traffic state using traffic and non-traffic input data sources on a deep learning model for short-term prediction of the network state.

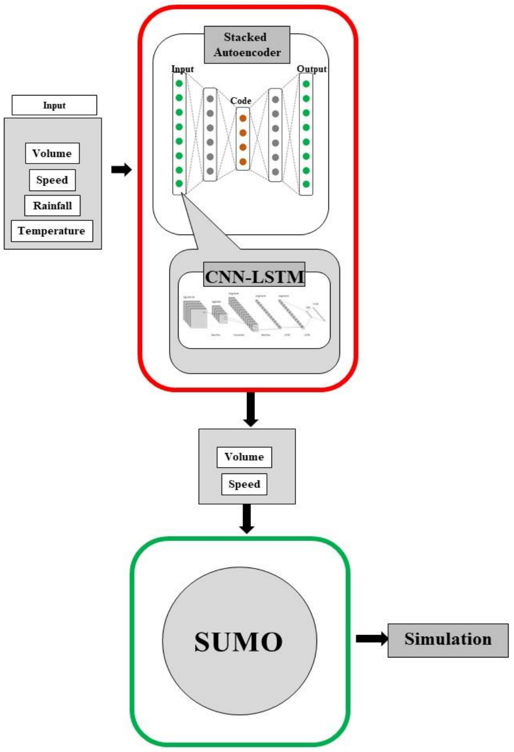

Fig. 1: Flowchart of Deep-PRESIMM

TABLE I. Summary of Related Extant Literature

Paper	Area [1]	Decision Support[2]	Prediction Model	Spatio-temporal	Real-Time?
[4]	M	N	Entropy Maximization	N	N
[11]	M	N	Kalman Filter	Y	Y
[12]	M	N	RNN	N	N
[13]	U	N	Fast Simulation	N	F
[5]	U	Y	Kalman Filter	Y	Y
[14]	U	N	Neural Network	Y	Y
[15]	U	N	ARIMA	N	Y
[9]	M	Y	Fast Simulation	Y	Y

III. Methodology

Fig. 1 presents the flowchart for the proposed Deep-PRESIMM model for proactive traffic management. The model comprises two distinct stages, prediction and simulation respectively. In the next subsections, we will present further details about the underlying concepts of the respective stages.

A. Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM)

Recurrent Neural Networks (RNN) [16] are a class of traditional neural networks that extend the functionality of traditional neural networks by considering the temporal

[1] M: Motorway/Highway, U: Urban

[2] Y: Yes, N: No

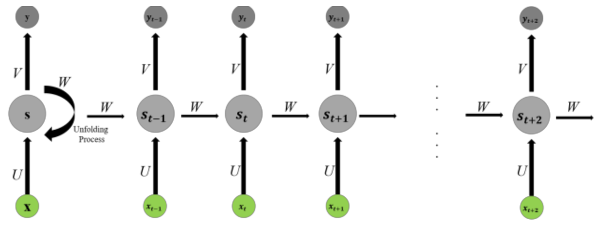

Fig. 2: Structure of the RNN

dimension of time series data. The basic functioning of RNN networks incorporate loops or ‘recurrent’ components to connect the neuron to itself and unfold it many times such that it is able to preserve the sequential dimension found in time series 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 Fig. 2.

In Fig. 2, 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 (1)

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

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. RNNs display vulnerabilities when applied to time series datasets with long lags [17]. From (1), the recurrent hidden state h_t approaches infinity or zero as the time interval increases, which leads to a diminishing or exploding gradient [18], [19]. This is a vulnerability that is encountered when using RNNs for long-term time series modelling, which was resolved by the work of German engineers Hochereiter and Schmidhuber [20] – Long Short-Term Memory RNNs – 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 atomic component of the LSTM neural network with one memory block is shown in Fig. 3. The memory block contains input, output, and forget gates, which respectively represent write, read, and reset functions on each cell. The memory cell contains a node that is connected to the recurrent edge of the fixed weight node. The multiplicative gates allow the model to store information over long periods, thereby ensuring that the gradient survives longer time steps without vanishing or exploding [20], and eliminating the vanishing gradient problem commonly observed in traditional neural network models [18].

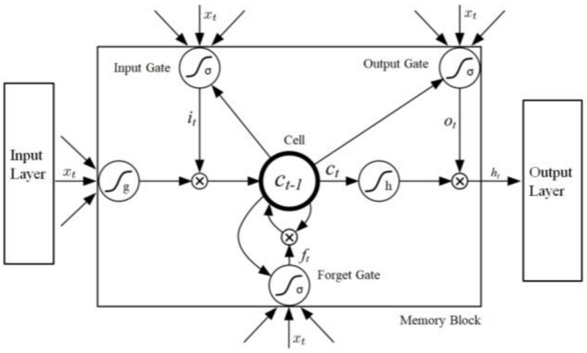

Fig. 3: LSTM RNN having one memory block

B. Stacked autoencoders (SAE)

An autoencoder is a type of feedforward network that has its input the same as its output. Autoencoders compress the input vector to a lower-dimensional ‘code’ and attempt to reconstruct the output from this given code. Fig. 4a shows the basic structure of an autoencoder. As can be seen, the autoencoder consists of three major components: the encoder, the code, and the decoder. The functions of these elements are as easy as their respective names suggest. The encoder compresses the input to a ‘latent space’ to produce the code, which is then decoded by the decoder. In other words, the output is reconstructed from the code using the decoder. Fig. 4b shows the architecture of the SAE used within the prediction stage in Deep-PRESIMM. As can be seen, the SAE comprises four hidden layers, including two CNN-LSTM layers and a fully connected layer tied to a linear output layer.

The learning process involves layer-wise training to minimize the error between the input and output vectors. The activation function applied is the Rectified Linear Unit (ReLU), which is mathematically defined by (2).

$$g(z) = \max\{0, z\} \qquad (2)$$

The subsequent layer of the autoencoder is the hidden layer of the previous one, with each of the layers trained by gradient

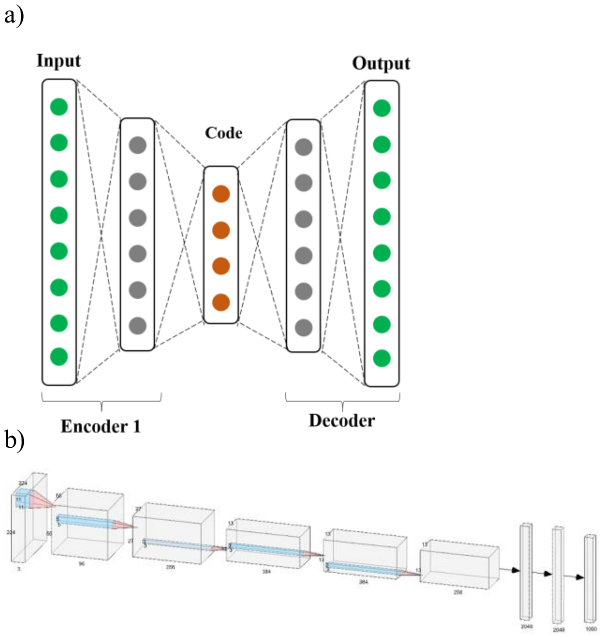

Fig. 4. a) Basic structure of the stacked autoencoder b): Proposed architecture of SAE layer in Deep-PRESIMM prediction module comprising CNN-LSTM layers $F^{2\quad 4} = 128$ for feature and sequential learning with a linear fully-connected layer.

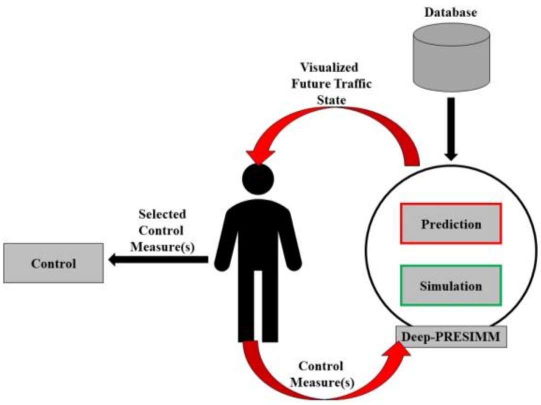

Fig. 5: Conceptual model of Deep-PRESIMM

descent algorithm using an optimization function, which is the squared reconstruction error J of the individual autoencoder layer.

$$\operatorname{argmin}_{W_1,b_1,W_2,b_2}[J] = \underset{W_1,b_1,W_2,b_2}{\operatorname{argmin}}\left[\frac{\sum_{i=1}^{m}\|x_i - x_i'\| + J_{wd} + J_{sp}}{2}\right] \qquad (3)$$

Where J represents the squared reconstruction error of the single autoencoder layer, x_i and x_i' respectively represent the ith 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.

C. Simulation using SUMO

[21] refer to two main open-source agent-based traffic simulation software: Multi-Agent Transport Simulation Toolkit (MATSIM) [22], a toolbox for implementing largescale agent-based simulations, and Simulation of Urban Mobility (SUMO) [23], a portable microscopic road traffic multi-agent simulation package designed to handle large road networks. Apart from the two mentioned above, a number of multi-agent simulation packages exist, such as VISSIM, VISUM, etc. However, after careful consideration, we decided to use SUMO as the traffic simulation tool. This stems from SUMO being an open-source tool, with space continuous and time discreet capabilities, providing individual routes for vehicles start and end times and positions and most importantly, SUMO allows for import of maps or networks from OpenStreetMap[3], an open-source online map.

The first task of the simulation stage using SUMO involves the generation of the road network, which was downloaded from OpenStreetMap. Once imported to SUMO, the map, alongside the predicted speeds and volumes are used to create the simulation of the road network in SUMO using a built-in application package *DFROUTER* - which uses inductive loop values to compute respective vehicle routes using a variant of the classic car-following model [24] developed by German transport company DLR [25]. SUMO allows for customizations to be performed on the virtual road network such as lane closure/opening, traffic light signal alteration, etc. which would offer the possibility of performing detailed analyses on the future or predicted traffic network state. The conceptual control loop showing the traffic control operator/analyst, the Deep-PRESIMM model and the consequent actions are represented in Fig. 5. As depicted, the choice of the 'final' intervention/control measure to be applied by the control personnel is made easier by the iterative feedback control cycle in PRESIMM, where the control personnel can have a number of 'what-if' situations, before arriving at the intervention that is most effective.

IV. Data Description

The traffic data used in this study was made available by Transport for Greater Manchester (TfGM) authority. The dataset comprised real-time 5-minute-aggregated observations of traffic flow characteristics (i.e. average speed, flow, density), collected using inductive loop sensors. The study period spans from 1st January 2012 to 2nd May 2014. The study area is an urban arterial road in Stretford, Greater Manchester, UK. The weather data for the study period comprised hourly observations of temperature (Celsius) and precipitation (measured in millimeters). The weather data was obtained from the Centre for Atmospheric Studies (CAS), University of Manchester. The weather station is located within a 3-mile radius of the study area.

For this study, the traffic data obtained was aggregated into hourly intervals in order to match the weather data aggregation level, which were hourly observations. The combined dataset comprised hourly observations of traffic flow characteristics (speed, flow, density) and weather data (rainfall and temperature), resulting in 20,448 observations of five variables or features. The dataset was split using a training-testing ratio of 70:30. Table I presents the descriptive statistics for the dataset used in this study.

TABLE II. Descriptive Statistics of Dataset

Statistic	Temp °C	Rainfall (mm)	Volume (veh/hr)	Density (veh/mi)	Speed (mph)
Mean	11.1740	0.1043	813.9798	26.8031	26.9
Standard Deviation	5.2877	0.5056	555.5547	24.6022	4.1803
Variance	27.9600	0.2557	308641.0	20783.08	25.135
Minimum	-4.5140	0	7	2	15.9
Maximum	32.6163	5029.67	2627	193.7	35.6
Count	20448	20448	20448	20448	20448

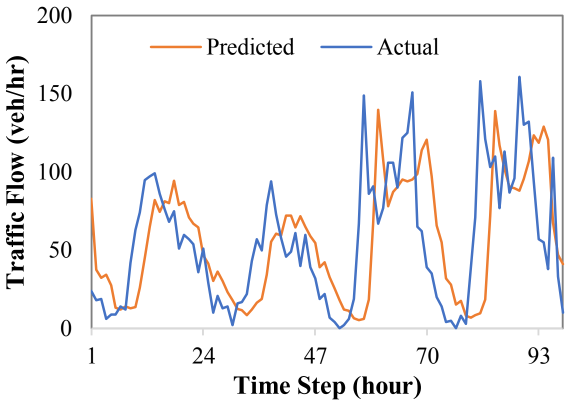

Fig. 6: Prediction performance of the Deep-PRESIMM prediction module

[3] OpenStreetMap can be found at: https://www.openstreetmap.org/

V. Experimental Setup

Within this study, we adopt the prediction setup as described in [26], where a rolling/overlapping sliding window approach is adopted for reconstructing the input time series data. The predictive model is designed to learn the features provided within the historic traffic and weather datasets and make multivariate (i.e. speed and volume) prediction. This will subsequently be passed as input to the next stage of the model, which is the simulation stage.

A. Model Description

The prediction stage within our proposed model incorporates deep CNN-LSTM stacked autoencoders. The LSTMs used are bi-directional (the hidden layer of the previous layer serving as the visible/input layer of the next layer), with each having 200 hidden units. For all of the inter-connected layers (except the output layer), the activation function utilized is the Rectified Linear Unit (ReLU), which introduces non-linearity to the learning process, and defined according to equation (2). The overall algorithm for the Deep-PRESIMM model is presented in Algorithm 1.

Algorithm 1: Algorithm for Deep-PRESIMM

```
Input: multivariate observed sequences $X = x_1, x_2, x_3, \dots, x_T$
Output: microsimulation of multi-step predicted value
Split $X$ into training and testing data with 70:30 ratio
Randomly initialize model parameters weight $w_t$ and bias $c$
for each iteration do
    Select lookback steps/batch size $b$
    Generate lookback sequence as $x_1, x_2, \dots, x_b$
    Train using forward greedy layer-wise and bi-directional processing
    Obtain actual value and compute error (i.e. MSE)
    Update model learning by backpropagation algorithm using optimizer (Adam in this case), minimizing 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.
    End for
Reiterate until training set is exhausted
Return multi-step output sequence of speed and volume $Y_v$ and $Y_s$.
Input predicted values to SUMO and create simulation using DFROUTER
```

B. Model Performance Evaluation

In terms of model evaluation, we use a technique referred to as *walk-forward validation* or *back testing* [27]. Traditional prediction evaluation methods such as k-fold cross validation or train-test hold-out data splits do not work when applied to time series data, due to the fact that in time series data, the sequential or temporal dimension needs to be preserved. Therefore, walk-forward validation refers to a method of testing the predictive model in a manner wherein the temporal dimension is preserved.

For model accuracy evaluation, we apply two prediction accuracy evaluation metrics – Mean Absolute Error (MAE) and Root Mean Square Error (RMSE), which are defined by the respective equations below.

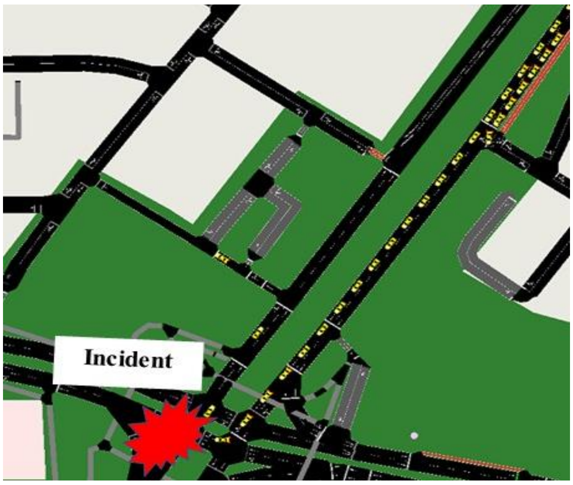

Fig. 7: Congested Junction as a result of the incident

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|e_i| \tag{4}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(e_i^2)} \tag{5}$$

C. Implementation Environment

The experiment environment used for this study was on a single machine in Windows 10 Operating System with Intel® Core ™ i7-6800K CPU @3.40 GHz, 32-GB Memory, and NVIDIA Quadro K420 GPU. The GPU is used for accelerated model training due to large computation demand in deep learning models. The development was performed using Python 3.6.8, R version 3.5.1, SUMO v1.1.0 and Tensorflow 1.12.0.

D. Prediction Results

The performance of the prediction module in Deep-PRESIMM was evaluated by carrying out single-step prediction of traffic volume using the available dataset described in the preceding section. Fig. 6 illustrates a subset (containing 100 time steps) of the results of the predictive model, which has been described in Section III above. As can be seen, the model accurately predicts the traffic volume, indicating its ability to perform adequately. Table III shows the summary of the prediction results from the predictive module of Deep-PRESIMM. The model achieved an RMSE of about 40 (veh/hr), which is satisfactory. The most significant time taken was during the model training period, which took 723s. However, the prediction time or inference time was almost negligible (0.32μs), highlighting the capacity of the predictive model to perform real-time traffic parameter prediction.

TABLE III. Summary of Results

Our Model	MAE	RMSE	Training Time (s)
CNN-LSTM-AE	29.8407	39.9207	723

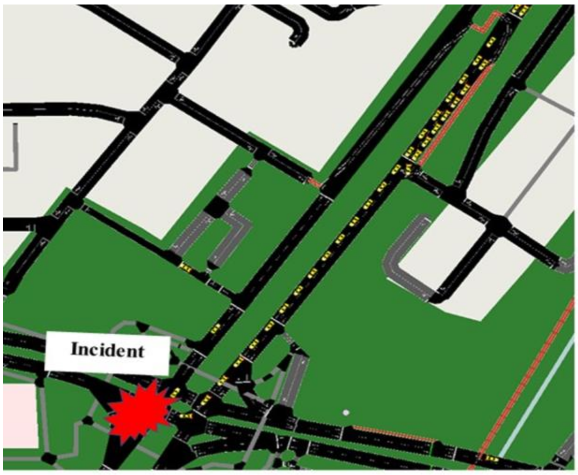

Fig. 8: Consequence of opening up a bus lane (1 hour later)

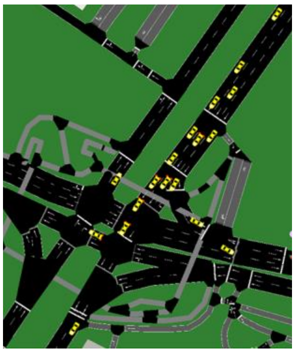

Fig. 9a: Traffic Signal alteration run

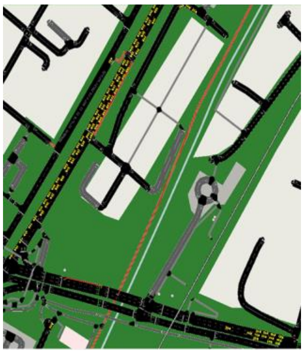

Fig. 9b: Signal alteration causes congstion at adjacent road links (1 hour later)

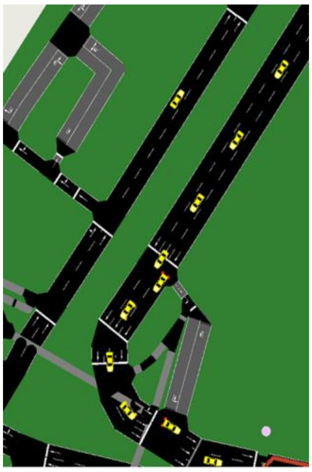

Fig. 10a: Traffic Diverted to link road

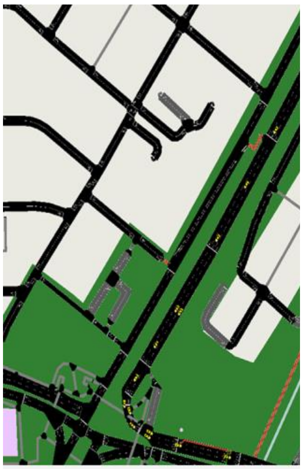

Fig. 10b: Diversion reduces congestion (1 hour after)

VI. Case Study

We evaluated the performance of Deep-PRESIMM using a case scenario, which was re-modelled from a real major road accident that caused severe traffic congestion along the A56 road within the city of Stretford, southwest Manchester on the 2nd May 2014 at 13:04hrs. Accident data was obtained from an online database[4], which made available a report containing the date and time of the incident, the geo-coordinates of the road involved, as well as the cause of the incident. The focus in the figures should be on the network of cars, rather than on the quality of the screenshots of the figures. The images presented are actual screenshots of the results from the simulations performed in Deep-PRESIMM.

Using Deep-PRESIMM, a simulation run of the predicted network state at 14:00 (i.e. 1 hour after the time of the accident) was visualised, and a screenshot is presented in Fig. 7. The figure shows a heavily congested network especially around the junction, with slow moving traffic and building queues. As a way of controlling or managing the predicted network congestion state, we are going to assume the traffic analyst (in this case) can only choose from three available traffic control measures: (1) Add an extra lane (i.e. free up the bus lane), (2) Alter the signal priority for the junction traffic light, and (3) Divert upstream traffic to a link road. We therefore created visualisations (and revised predictions) of the consequence of the applied control measures using Deep-PRESIMM. The results of the respective control measures are discussed in the subsequent subsections.

A. *Traffic Control (TC) 1: Adding an extra lane*

Most cities introduce bus lanes (Bus Rapid Transit [28]) as a means of speeding up public transport to control congestion and encourage public transport usage. By this, a dedicated lane is set aside for buses and taxis or emergency services. In this scenario, the control analyst may decide to free up the bus lane (i.e. open it up for use by private vehicles). In our network, this is equivalent to adding an extra lane. The simulation run for TC-1 showed an increase in the traffic flow (i.e. vehicles/hour), indicating that the network reaches its overload or saturation point quicker than the in original 1-hour-ahead simulation (see Fig. 7). This implies that TC-1 resulted in a more congested network 1-hour after its implementation (see Fig. 8).

B. *TC 2: Signal Alteration*

Adaptive Traffic Control Systems (ATCS) are used for traffic control by automatically adjusting traffic signals based on traffic conditions. The major aim of ATCS is to maximize road network throughput [29]. SUMO allows for signal alteration, using its interactive interface – NETEDIT. The simulation run for TC-2 is presented in Fig. 9a, with the consequence (i.e. 1-hour later) shown in Fig. 9b. The simulation showed a network that experiences free flow for a short period, accompanied by a rapid increase in traffic flow, which results in network overload and congestion in the adjacent road link.

C. *TC 3: Road Diversion*

In very congested traffic situations caused by significant reductions in road capacity (brought about by accidents or road construction works), road diversions are employed to free up the network. It is advantageous in that it frees up the congested road, but has the disadvantage of 'transferring' the congestion to the link road. The simulation run for TC-3 is presented in Fig. 10a and 10b. The visualisation of the consequence of this

[4] https://www.crashmap.co.uk/Search

action can be seen as the affected road is freed up once all traffic is diverted to the link road. TC 3 appears to be the most viable control measure, even from a non-technical 'common-sense' standpoint. This is because the accident caused a reduction in the road capacity (by 50% due to the closure of a lane), and the intuitive optimal solution would be to divert the upstream traffic to a link road.

VII. Discussion

This paper presents a proactive traffic management model that can be useful to traffic network managers by adding an extra visualization layer to typical short-term traffic prediction models. The forecasted traffic characteristics are used as input to a traffic simulation tool, which can provide guidance in decision making to traffic network controllers through visualizations of the consequent effects of the actions or interventions that are applied in order to control or eliminate traffic congestion on given road networks.

Using a deep CNN-LSTM stacked autoencoder architecture applied on historical sensor-collected traffic and non-traffic datasets, multivariate (speed and volume) traffic prediction was made on a 1-hour prediction horizon. The forecasted traffic parameters were used as inputs to SUMO, where subsequent simulations of the predicted traffic network state enabled the visualisation of the future state of the network. We re-modelled a case scenario involving a road accident on A56 Chester Road in Greater Manchester, UK. Deep-PRESIMM was used to create three 'what-if' scenarios a control operator could apply in a bid to managing the congested network. For this study, the number of traffic control measures were limited to three (3), and these were actualised in Deep-PRESIMM, as well as the visualizations of the consequences of the applied control measure, making the decision-making process easier.

However, although the case study presented a scenario with the control analyst having only three (3) possible TC alternatives, this is not the case in real-world traffic control centres, where there are many possible control alternatives. Deep-PRESIMM can accommodate any number of possible TC alternatives. Traffic analysts will be aware of the control measures available to ease congestion. Deep-PRESIMM examines these known control measures and evaluates them. Deep-PRESIMM performs analysis using real-time data collected from traffic sensors, offers 1-hour ahead traffic predictions, and therefore is the most accurate representation of the predicted traffic state. As such, even though the options examined are known to the analyst, Deep-PRESIMM presents/visualises the best available one, thereby making the choice of traffic control measure easier.

This model can be an effective traffic management tool, by presenting to the control operator information about the future state of the network, as well as the likely consequence of the actions or interventions applied on the network. For instance, when there is traffic congestion due to a temporary issue such as accidents, traffic light failure, start/end of a football match, or city events such as concerts, etc., traffic analysts will be aware of what measure(s) can be applied towards ameliorating the situation. With this model, the traffic management and control process changes from a 'reactive' to a 'proactive' one, where the future state is visualized, alongside with the consequences of the interventions. This would make the decision-making process for the traffic control personnel a lot easier and more objective, rather than on the basis of 'gut instinct', or subjective experience. It is important to mention that our research does not claim that experience or domain knowledge is irrelevant in traffic analysis and control, nor are we proposing a replacement to existing traffic control systems. However, the proposed tool can serve as a complementary guidance aid towards proactive traffic management and control. This model explicitly differs from existing popular mobility information provision systems like Waze[5], Garmin[6], and Google Maps[7] in that the stakeholders for Deep-PRESIMM are traffic network controllers/authorities, as opposed to road users (end users) serviced by the other systems mentioned.

This study is limited in a number of ways. Firstly, the model can be modified to include a self-learning feedback loop that is updated by the traffic analyst. This can contain information about the feedback obtain from the traffic analyst's traffic intervention, which, in effect, would include the tacit knowledge of the analyst as well as the feedback of the chosen intervention. Secondly, the research study was constrained to Greater Manchester, UK dataset. However, generalizing it to accommodate for other locations is possible, and achievable. The underlying structure and architecture would still remain the same, with minimal changes made in terms of geophysical information (maps), input datasets, and additional nodes on the simulation side. Thirdly, the traffic network used in this study was restricted to vehicular traffic only, excluding rail, cycle, and pedestrian traffic modes. This has the potential to limit the realism of the simulation due to omitting the interactions between the other travel modes. Finally, the model was constrained to unidirectional (i.e. only the affected lane) on the incident road.

VIII. Conclusion And Future Work

This paper presented Deep-PRESSIM, a proactive traffic management model comprising two distinct stages – prediction and simulation respectively. This makes it a unique tool aimed at providing assistance to traffic network control personnel, by proactively managing traffic flow using a number of 'what-if' scenarios and consequently visualizing the effect of the intervention on the virtual future traffic state. The model presented in this paper can be further trained to perform on more complex networks and accommodate real-time traffic inputs. Furthermore, additional test case scenarios can be performed, such that the effects of the interventions by traffic control officers can be simulated and compared against real-time traffic. For instance, actually altering a traffic signal in real-time to compare against how this is simulated in SUMO.

Finally, this work can be developed to contribute towards ushering in smart cities, where Deep-PRESIMM can be integrated into existing ITS such that a number of preset interventions on the ITS are made available to the operator (for

[5] https://www/waze/com
[6] https://www.garmin.com
[7] https://www.google.com/maps

instance, lane closure, signal alterations, etc.), so that these interventions are also simulated to see their respective consequent effects on the traffic situation. This is a step in the right direction for the introduction of driverless cars and could serve as a stepping stone to effective traffic control and management. A proactive traffic management model, such as Deep-PRESIMM, could be a step in the direction of providing holistic control and proactive management of traffic with potential for facilitating the coordination of driverless cars by traffic authorities.

REFERENCES

[1] E. I. Vlahogianni, M. G. Karlaftis, and J. C. Golias, "Short-term traffic forecasting: Where we are and where we're going," *Transp. Res. Part C Emerg. Technol.*, vol. 43, pp. 3–19, 2014.

[2] M. G. Karlaftis and E. I. Vlahogianni, "Statistical methods versus neural networks in transportation research: Differences, similarities and some insights," *Transp. Res. Part C Emerg. Technol.*, vol. 19, no. 3, pp. 387–399, 2011.

[3] B. L. Smith, B. M. Williams, and R. Keith Oswald, "Comparison of parametric and nonparametric models for traffic flow forecasting," *Transp. Res. Part C Emerg. Technol.*, vol. 10, no. 4, pp. 303–321, 2002.

[4] A. Dombalyan, V. Kocherga, E. Semchugova, and N. Negrov, "Traffic Forecasting Model for a Road Section," *Transp. Res. Procedia*, vol. 20, pp. 159–165, 2017.

[5] A. Abdelghany, K. Abdelghany, H. Mahmassani, and P. Murray, "Dynamic Traffic Assignment in Design and Evaluation of High-Occupancy Toll Lanes," *J. Transp. Res. Rec.*, vol. 1733, pp. 39–48, 2000.

[6] H. Xu and D. J. Dailey, "Real time highway traffic simulation and prediction using inductance loop data," in *Vehicle Nagivation and Information Systems Conference Proceedings*, 1995, pp. 194–199.

[7] N. S. Nafi, R. H. Khan, J. Y. Khan, and M. Gregory, "A predictive road traffic management system based on vehicular ad-hoc network," in *2014 Australasian Telecommunication Networks and Applications Conference, ATNAC 2014*, 2015.

[8] H. Hashemi and K. F. Abdelghany, "Real-time traffic network state estimation and prediction with decision support capabilities: Application to integrated corridor management," *Transp. Res. Part C Emerg. Technol.*, vol. 73, pp. 128–146, 2016.

[9] M. Ben-akiva, M. Bierlaire, H. Koutsopoulos, and R. Mishalani, "DynaMIT: a simulation-based system for traffic prediction," in *DACCORD Short Term Forecasting Workshop*, 1998, pp. 1–12.

[10] A. Essien, I. Petrounias, P. Sampaio, and S. Sampaio, "The impact of rainfall and temperature on peak and off-peak urban traffic," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2018, vol. 11030 LNCS, pp. 399–407.

[11] M. Fountoulakis, N. Bekiaris-Liberis, C. Roncoli, I. Papamichail, and M. Papageorgiou, "Highway traffic state estimation with mixed connected and conventional vehicles: Microscopic simulation-based testing," *Transp. Res. Part C Emerg. Technol.*, vol. 78, pp. 13–33, 2017.

[12] M. Zhou, X. Qu, and X. Li, "A recurrent neural network based microscopic car following model to predict traffic oscillation," *Transp. Res. Part C Emerg. Technol.*, vol. 84, pp. 245–264, 2017.

[13] N. M. Abid and S. S. Hussain, "Transportation network planning using simulation: Case study\ Al Mansour city," in *2017 2nd IEEE International Conference on Intelligent Transportation Engineering, ICITE 2017*, 2017, pp. 272–279.

[14] W.-K. Lai, T.-H. Kuo, and C.-H. Chen, "Vehicle Speed Estimation and Forecasting Methods Based on Cellular Floating Vehicle Data," *Appl. Sci.*, vol. 6, no. 2, p. 47, 2016.

[15] G. Zhu, L. Wang, P. Zhang, and K. Song, "A Kind of Urban Road Travel Time Forecasting Model with Loop Detectors," *Int. J. Distrib. Sens. Networks*, vol. 12, no. 2, 2016.

[16] P. Rodriguez, J. Wiles, and J. L. Elman, "A Recurrent Neural Network that Learns to Count," *Conn. Sci.*, vol. 11, no. 1, pp. 5–40, 1999.

[17] K. Cho *et al.*, "Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation," in *2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1724–1734.

[18] S. Hochreiter, Y. Bengio, P. Frasconi, and J. Schmidhuber, *Gradient flow in recurrent nets: the difficulty of learning long-term dependencies*. IEEE Press, 2001.

[19] S. Hochreiter, "The Vanishing Gradient Problem During Learning Recurrent Neural Nets and Problem Solutions," *Int. J. Uncertainty, Fuzziness Knowledge-Based Syst.*, vol. 6, no. 2, pp. 107–116, 1998.

[20] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[21] B. Chen and H. H. Cheng, "A review of the applications of agent technology in traffic and transportation systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 2, pp. 485–497, 2010.

[22] A. Horni, K. Nagel, and K. W. Axhausen, *The Multi-Agent Transport Simulation MATSim*. London: Ubiquity Press, 2016.

[23] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz, "SUMO–simulation of urban mobility: an overview," in *Proceedings of SIMUL, The Third International Conference on Advances in System Simulation*, 2011.

[24] P. G. Gipps, "A Behavioural Car Following Model for Computer Simulation.pdf," *Transportation Research Part B: Methodological*. Elsevier B.V., pp. 105–111, 1981.

[25] "DLR - Institute of Transportation Systems - SUMO – Simulation of Urban MObility." [Online]. Available: https://www.dlr.de/ts/en/desktopdefault.aspx/tabid-9883/16931_read-41000/. [Accessed: 19-Mar-2019].

[26] A. Essien, I. Petrounias, P. Sampaio, and S. Sampaio, "Improving Urban Traffic Speed Prediction Using Data Source Fusion and Deep Learning," in *2019 IEEE International Conference on Big Data and Smart Computing (BigComp)*, 2019, pp. 1–8.

[27] I. Kaastra and M. Boyd, "Designing a neural network for forecasting financial and economic time series," *Neurocomputing*, vol. 10, no. 3, pp. 215–236, 1996.

[28] H. S. Levinson, S. Zimmerman, J. Clinger, and G. Rutherford, "Bus Rapid Transit: An Overview," *J. Public Transp.*, vol. 5, no. 2, p. 1, 2002.

[29] Q. He, K. L. Head, and J. Ding, "PAMSCOD: Platoon-based arterial multi-modal signal control with online data," *Transp. Res. Part C Emerg. Technol.*, vol. 20, no. 1, pp. 164–184, 2012.