

# Integrating data-driven and simulation models to predict traffic state affected by road incidents

Sajjad Shafiei <sup>a\*</sup>, Adriana-Simona Mihăiță<sup>a,b</sup>, Hoang Nguyen<sup>a</sup>, Chen Cai<sup>a</sup>

<sup>a</sup>*Transport Analytics Group DATA61 | CSIRO, Sydney, Australia*

<sup>b</sup>*Faculty of Engineering and IT, University of Technology Sydney, Australia*

\*corresponding: sajjad.shafiei@data61.csiro.au

Predicting the traffic conditions in urban networks is a priority for all traffic management centres around the world. This becomes very challenging, especially when the network is affected by traffic incidents that vary in both time and space. Although data-driven modelling can be considered an ideal tool for short-term traffic predictions, its performance is severely degraded if little historical traffic information is available under incident conditions. This paper addresses this challenge by integrating data-driven and traffic simulation modelling approaches. Instead of directly predicting the traffic states using limited historical data, we employ a traffic simulation reinforced by data-driven models. The traffic simulation uses newly reported incident information and the estimated origin-destination (OD) demand flows to capture the complex interaction between drivers and road network, and predicts traffic states under extreme conditions. Because accurate real-time OD flows cannot be directly measured in large-scale areas, we propose a rolling-horizon OD demand estimation problem to estimate demand flows based on the most recent measured link volumes. We showcase the capability of the proposed data-driven enforced traffic simulation platform for incident impact analysis in a real-life sub-network in Sydney, Australia.

Keywords: demand estimation and prediction; micro-simulation; machine learning; incident management

## 1. Introduction

Short-term traffic forecasting is a necessary step for efficient traffic network operations and is an integral part of intelligent transportation system (ITS) applications. The abundance and recent increase of various traffic data sources have led many researchers and data scientists to employ a wide range of data-driven models to predict future traffic conditions. Various parametric and non-parametric methods are used for the short-term forecast of speed (Xu et al., 2018; Yao et al., 2017), travel time (Wang et al. 2016; Oh et al. 2015) and traffic volume (Polson and Sokolov 2017), which offer predictions from a few minutes to several hours into

the future. However, two challenges are highlighted in the previous studies as the main critical limitations of the majority of data-driven models (Tedjopurnomo et al., 2020; Vlahogianni et al., 2014). Firstly, most data-driven forecasting models have been applied to freeways or arterial corridors rather than urban networks. The complex spatial configuration of all network connections and the dynamics of the travel demand make traffic forecasting in urban networks challenging, particularly for large suburban networks. Secondly, a wide variability of traffic incidents can occur at different times of the day and rarely in the same location or with the same severity. These incidents can range from temporary lane closures due to car breakdowns and small-scale accidents to more complicated incidents such as sudden weather changes and train system breakdowns (Tercan et al., 2020). All these varying characteristics of traffic disruptions increase forecasting complexity and make it difficult to find similar patterns in any historical dataset. To address this issue, some studies focus only on a particular type of incident, for example, highway lane closures due to roadway reconstruction projects (Du et al., 2017; Karim and Adeli, 2003). However, in most cases, there are limited recorded data for each particular type of incident, and thus, forecasting traffic measurements such as traffic volume, speed, or travel time can result in inaccurate outcomes.

On a parallel research track, the traffic state is estimated using real-time traffic simulation models (Oh et al. 2018; Li et al. 2015). In these models, each traveller attempts to minimise their travel time/cost, and their decision impact other travellers' decisions in the network. By considering this essential principle, the intricate traveller route decisions can be modelled in the traffic network. In addition, the propagation of traffic along the network is replicated by traffic flow theories that determine the traffic flows and the associated travel times on network links (Cascetta, 2009). However, the traffic simulation models require several parameters that should be well-calibrated for operational applications (Balakrishna et al., 2012).

In this paper, instead of directly predicting the link traffic features using only limited historical data (which does not have recorded correlations of specific anomalies or past accidents), we used data-driven models to reinforce a traffic micro-simulation by providing the origin-destination (OD) information as one of the most important required inputs. In our proposed framework, once an incident is categorised as severe by an artificial intelligence (AI) engine (Nguyen et al., 2017), a summary of incident characteristics such as the location and the number of affected lanes is transferred to the traffic simulation. Moreover, a machine learning method predicts short-term OD demand flows and feeds them into the traffic simulation. With

knowledge of the incident information and predicted OD demand flows, the simulation applies traffic flow principles to predict the traffic state under non-recurrent conditions.

For this study, we assumed that network commuters hardly cancel their short-term trips even if affected by disruptions. This assumption holds for morning peak hours in which a large proportion of trips like home-to-work or home-to-education still need to continue. Some travellers respond to a new prevailing bad network condition by updating or changing their route trips. Such behavioural phenomena can be modelled accurately in traffic simulations. We propose a rolling-horizon bi-level optimisation model to keep the traffic simulation model calibrated based on the most recent measured traffic data. Finally, we investigate the impacts of traffic incidents on a real-life application in an urban sub-network and showcase the benefit of the integrated approach. To summarise, we present our main contributions for this paper as:

- proposing a prototype for an incident management platform using integrated data-driven and dynamic traffic simulation modelling;
- estimating day-to-day OD flows for the OD demand prediction module when an incident occurs. We propose several machine learning models for OD demand prediction to reinforce the traffic simulation;
- deploying traffic micro-simulation modelling according to real-life adaptive signal control by applying the same controllers' logics to simulated vehicles.

This paper extends our previous study presented in (Shafiei et al., 2020) by adding travel time validation, real-life adaptive signal control integration, and improving OD flow prediction methodology. The rest of the paper is organised as follow: Section 2 describes the methodology applied over a real sub-network in Sydney, Australia and Section 3 showcases the results. Finally, Section 4 provides the concluding remarks and outlines some research extensions for future studies.

## **2. Methodology**

This section explains the general data flow used for building an operational incident management platform tailored to the needs of the Traffic Management Centre (TMC) in Sydney, Australia. Next, OD demand estimation and prediction methodology are discussed as the main focused components of the incident management platform in this paper.

## 2.1 *Incident management platform*

Figure 1 presents the proposed incident impact analysis platform using integrated data-driven and traffic simulation models. The methodological diagram treats the cases of recurrent versus non-recurrent traffic conditions differently by triggering various modules detailed below.

Our proposed framework is using various data types, including:

- **measured link traffic count:** link traffic counts are essential inputs for the OD demand estimation module as well as to validate the framework's output.
- **Network and demand data:** which contains the primary information of the links of the network, public transport lines and their frequency, and signal configuration. This data is exported and used to construct the baseline simulation model in an offline matter.
- **Incident data logs:** includes the incident location ([x,y] coordinates), the number of affected lanes, the start time of the incident, the incident duration and more details about the incident severity.

By having such information, the proposed framework predicts the traffic state in two streams of recurrent and non-recurrent traffic conditions using the following modules:

- **Incident severity classification:** this module includes raw data processing and uses machine learning techniques to classify reported incidents into severe and non-severe. When an incident is categorised as severe, its data is transferred to the traffic simulation model for further impact analysis. For non-severe incidents, the data-driven traffic modelling is used to obtain the short traffic prediction; this module is not a focus of this study and readers can refer to our previous work published in (Nguyen et al., 2017).
- **Data-driven traffic prediction:** this module is based on our previous study (Nguyen et al., 2019), which proposed a deep learning methodology for travel speed prediction involving feature generation, model development, and model deployment. The proposed neural network model is used for recurrent traffic conditions.
- **OD demand estimation:** this module aims to adjust a priori demand data by using link traffic observations. We estimate the OD flows through a bilevel optimization framework in which the a priori demand flow is updated based on the latest measured traffic in several links of the network.

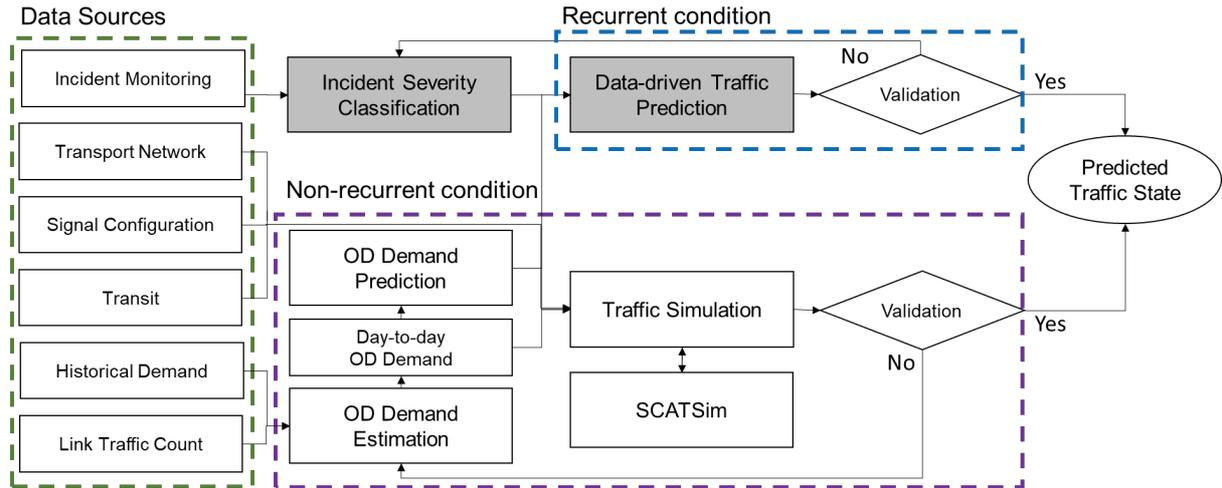


Figure 1. Data-driven and simulation platform for incident impact analysis. The grey modules are beyond the scope of this paper.

- **OD demand prediction:** the demand prediction model will forecast the OD trip flows for short time intervals into the future (up to an hour ahead) by using the OD demand flows obtained by the OD demand estimation module.
- **Traffic simulation:** this is used to understand the allocation of the predicted OD demand flows considering the freshly reported incident information. The microsimulation modelling uses various inputs such as OD demand matrices, public transit lines and timetables, traffic signal control definition for each controlled intersection, incident information on where the disruption has been affecting. The traffic microsimulation outputs are numerous, and a summary of this includes network or individual link travel times, assigned link traffic volumes, network/link delay or speed/density, etc.
- **SCATSIM:** this is a simulation plug-in control architecture that responds to the simulated traffic state by adjusting: a) the total cycle times inside each SCATS-controlled intersection, b) the ratio of the cycle time assigned to each phase, and c) the offset between adjacent signal controls. As a result, the real-life SCATS control architecture is applied to the simulated vehicles in the micro-simulation model, which offers a realistic replication of real-world traffic control conditions.
- **Validation:** The inputs of the simulation model are regulated by the updated traffic measurements observed consistently from the network at each time-interval; this is to ensure the predicted results reflect the actual real-life conditions. If the error between the predicted and the corresponding observed values is less than an acceptable threshold

defined in NSW traffic modelling guidelines (Morgan and Veysey, 2013), then we accept the outputs of the platform as the final predicted traffic state.

## **2.2 Demand estimation and prediction**

The success of the traffic simulation relies on the quality of this fundamental input and how well it captures the travellers' movement in the city from one time interval to another. Accurate demand flow information is difficult to obtain directly and is normally estimated using link traffic measurements. The main objective of the OD demand estimation problem is to minimise the error between the simulated and the observed traffic measurements. The OD demand estimation problem can be expressed as a system of equations in which the unknown parameters are the OD flows, and each equation represents the observed link flow. Many studies apply a bi-level optimisation formulation where demand flows are estimated at the upper-level and the feedback of estimated demand in the network is evaluated by a lower-level traffic assignment model (Antoniou et al., 2016). Some relevant research on dynamic OD estimation problems include a) using advanced traffic surveillance data to improve the accuracy of estimated OD flows (Kim et al., 2018; Rao et al., 2018), b) proposing methodological enhancements to deal with non-linearity problems in congested networks (Frederix et al., 2011; Shafiei et al., 2017), and c) applying simultaneous adjustment of network and demand parameters to consider the complex interactions of demand and network components (Liu et al., 2020). Work on the OD estimation problem has been ongoing for decades, and interested readers can refer to the comprehensive survey studies by (Antoniou et al., 2016; Ziliaskopoulos, 2001).

Dynamic OD demand estimation based on traffic measurements is performed for offline and online applications. The offline demand flow estimation problem relies on historical measured traffic data, while in the online applications, the model should be able to update OD flows based on real-time data and predict the traffic data actively for the near future. Therefore, offline dynamic OD estimation can be considered a complementary module in which a reliable initial OD demand is estimated for online models. Kalman-filter and least-squares estimators are two common approaches used in generating real-time OD flows with a reasonable computational burden (Ashok, and Ben-Akiva, 2000; Zhou and Mahmassani, 2007). Wen et al. (2006) present an overview of the online OD estimation model and other functional requirements, which need to be considered when planning for the online deployment of traffic model-based systems.

In this study, we used the traditional bi-level optimization problem in which the OD demand flows are estimated by solving the system of equations at the upper level and then evaluated at the lower level. The OD demand estimation problem is mathematically expressed as follows:

$$\min F(X) = \omega \cdot \sum_{i=1}^I \sum_{t=0}^T f(\hat{x}_i^t, x_i^t) + (1 - \omega) \cdot \sum_{a=1}^A \sum_{t=0}^T f(\hat{y}_a^t, y_a^t) \quad (1)$$

$$\hat{y}_a^t = \sum_{\tau=1}^t \sum_{i=1}^I p_{a,i}^{\tau,t}(X) \hat{x}_i^{\tau}$$

where each parameter is explained here below:

- $f$  Euclidean distance function,
- $a$  Link index,  $a \in [1, A]$ ,  $A$  is the total number of observed links in the network
- $t, \tau$  Time index  $t, \tau \in [1, T]$ ,  $T$  is the total number of modelling discrete times,
- $i$  OD pair index  $i \in [1, I]$ ,  $I$  is the total number of OD pairs in the network,
- $\hat{x}_i^t$  Estimated demand flow for an OD pair  $i \in I$  at time  $t$ ,
- $x_i^t$  Initial demand flow for an OD pair  $i \in I$  at time  $t$ ,
- $X$  Current estimated demand vector,  $X = [x_1^T, x_2^T, x_3^T, \dots, \hat{x}_{I-1}^T, \hat{x}_I^T]$
- $\hat{y}_a^t$  Estimated link flow in link  $a$  at time  $t \in T$ ,
- $y_a^t$  Observed link flow in link  $a$  at time  $t \in T$ ,
- $p_{a,i}^{\tau,t}$  Assignment proportion of  $x_i^t$  that passes through a link  $a$  during period  $\tau$ ,
- $\omega$  Reliability weight for the demand deviation.

The lower level of Eq.1 assumes the link flows are linearly related to OD flows by defining the assignment proportion ( $p_{a,i}^{\tau,t}$ ). This assumption simplifies the OD demand estimation problem and makes it practical for offline and online large-scale applications. The assignment proportion includes useful information on the link between network and demand components of the traffic models. However, link flows ( $y_a^t$ ) are indicative of demand ( $\hat{x}_i^t$ ) only if there is no saturated bottleneck between the origin and this link flow location. In the case of the existence of congestion, the linear models may respond to forming congestion by decreasing the corresponding OD demand among the congested links (Frederix et al., 2011). In other words, the estimated demand results in the link flow which either indicates congestion or free-flowing traffic link condition. Therefore, minimizing the only difference between measured and simulated link flow can significantly affect the reliability of the solution found (Shafiei et al., 2015). To alleviate this issue, in addition to reducing the Euclidean distance between

simulated and observed flow rates, the second term of the objective function seeks to keep the estimated demand as close as possible to the initial demand. In this way, the solution does not explore the local optima, which presents different link condition with the initial one. Moreover, the result must be evaluated by other road characteristics such as speeds and travel times to ensure how well the simulation itself mimics traffic dynamics.

To solve the problem in Eq.1, the partial derivative of function F with respect to the demand flow for the OD pair  $\eta$  at time  $h$  ( $x_\eta^h$ ;  $h \in [0, T]$ ,  $\eta \in I$ ) is determined as follows:

$$\begin{aligned} \frac{\partial F}{\partial x_\eta^h} &= \frac{\partial}{\partial x_\eta^h} \left( \omega \sum_{t=1}^T \sum_{i \in I} (x_i^t - \hat{x}_i^t)^2 + (1 - \omega) \sum_{a=1}^A \sum_{t=1}^T (y_a^t - \hat{y}_a^t)^2 \right) \\ &= 2\omega(x_\eta^h - \hat{x}_\eta^h) + 2(1 - \omega) \\ &\quad \left( \sum_{a=1}^M \sum_{\tau=h}^T p_{a,\eta}^{\tau,h}(X) \left( \sum_{t=1}^{\tau} \sum_{i \in I} p_{a,i}^{\tau,t}(X) x_i^t - y_a^\tau \right) \right) \end{aligned} \quad (2)$$

In addition to determining the gradient, we also need to calculate the step size ( $\lambda$ ). To do so, the sub-optimization problem in Eq.3 is solved in each iteration:

$$\min_{\lambda} F\left(X + \frac{\partial F}{\partial x} \lambda\right) \quad (3)$$

We use the golden section algorithm as a line search minimization solution for solving the sub-optimization problem defined in Eq.3. The algorithm includes the following steps:

Step 1: if  $\lambda^* \in [0,1], \zeta = 0, \varrho = 1, \lambda_1 = \psi, \lambda_2 = 1 - \psi$  where  $\psi = \frac{\sqrt{5}-1}{2}$

Step 2: if  $F\left(X + \frac{\partial F}{\partial x} \lambda_1\right) > F\left(X + \frac{\partial F}{\partial x} \lambda_2\right)$

$\zeta = \lambda_2, \lambda_2 = \lambda_1, \varrho = \varrho, \lambda_1 = \psi$

if  $F\left(X + \frac{\partial F}{\partial x} \lambda_1\right) < F\left(X + \frac{\partial F}{\partial x} \lambda_2\right)$

$\zeta = \zeta, \varrho = \lambda_1, \lambda_1 = \lambda_2, \lambda_2 = \varrho - \psi$

Step 3: if  $\zeta - \varrho < \varepsilon$  then:  $\lambda^* = \frac{\zeta + \varrho}{2}$ ; otherwise, go to step 2.

With the knowledge of the gradient  $\left(\frac{\partial F}{\partial x_{\eta}^{\tau}}\right)$  and the step size ( $\lambda$ ), the demand flow ( $\hat{x}_{\eta}^{\tau}$ ) is then updated as:

$$\hat{x}_{\eta}^{\tau} = x_{\eta}^{\tau} + \lambda \frac{\partial F}{\partial x_{\eta}^{\tau}} x_{\eta}^{\tau} \quad (4)$$

The proposed OD demand estimation model is performed in two stages. First, we execute the procedure offline to estimate OD demands for a typical weekday. In this stage, an initial static OD demand obtained from the strategic model is used to estimate sixteen 15-minute OD demand matrices for a 4-hour simulation during peak hours. Second, the time-dependent OD demand matrices estimated from the first stage are adjusted in a rolling-horizon estimation procedure based on the most recent observed traffic counts. As a result, the simulation model is kept updated in real-time using the newly observed data. Moreover, the estimated OD demand flows are archived for different days of the week.

The archived demand is considered as a reliable training demand set for any demand prediction module. Let the current time be ‘ $t$ ’. Then, the demand prediction model has to predict demand flow for the next time intervals ( $\tilde{x}_i^{t+1}$ ) given the value of the ‘ $m+1$ ’ previous demand flow values ( $\hat{x}_i^{t-m}, \dots, \hat{x}_i^{t-1}, \hat{x}_i^t$ ). Thus, the prediction model is expressed as:

$$\tilde{x}_i^t = P(\hat{x}_i^t, \hat{x}_i^{t-1}, \dots, \hat{x}_i^{t-m}, t) \quad (5)$$

We use different predictors for building this OD demand prediction module, which are presented in the following sections.

### 2.3 Data-driven models

In this section, we provide a brief description of various algorithms that we used for demand flow prediction, among which we cite: support vector regression (SVR), decision trees (DT), extreme gradient boosting and autoregressive moving average (ARMA).

#### 2.3.1 Support Vector Regression

Support vector regression (SVR) in statistical learning theory has been widely applied for solving classification and regression problems (Geron, 2015). In general, SVR is a well-established prediction method in complex systems that can deal with noisy databases (Wu et

al., 2003). The SVR model is formulated as a constrained optimization problem specified in Eq.6. The problem consists of two conflicting objectives: minimizing the Euclidean distance of  $\omega$  to increase the margin and reducing the instance margin violations ( $\xi_i$ ). Constant  $C$  makes a balance between the two objective terms. A small value for hyper-parameter  $C$  allows a higher generalization ability while a big  $C$  value enforces serious penalty to limit instances (in our application demand flow  $x_i^t$ ) that violate the determined margin.

$$\begin{aligned} & \underset{\omega, b}{\text{minimize}} \quad 0.5 \omega^T \cdot \omega + C \sum_{i=1} \xi_i \\ & \text{so that:} \end{aligned} \tag{6}$$

$$\xi_i = \begin{cases} |\omega \cdot \Gamma(x_i^t) + b - x_i^t| - \varepsilon & |\omega \cdot \Gamma(x_i^t) + b - x_i^t| \geq \varepsilon \\ 0 & |\omega \cdot \Gamma(x_i^t) + b - x_i^t| < \varepsilon \end{cases}, \forall i$$

where  $\Gamma$  is a transformation function that can be replaced with a kernel function. Different kernel functions such as linear, polynomial, Gaussian radial basis function (RBF), and Sigmoid are commonly used in SVR modelling (Geron, 2015). Regardless of the kernel functions used, the main goal is to estimate the value of coefficients of  $\omega$  and  $b$  in the Eq.6. To do this, the parameters  $C$ ,  $\varepsilon$ , and  $\xi$  should first be defined. Once they are determined, there will be a global optimal solution found for the convex problem to obtain  $\omega$  and  $b$  values. In Section 3.2, we explore different values to find the best combination for our application.

### 2.3.2 Decision Trees

Decision Trees (DT) are versatile non-parametric supervised ML algorithms that are capable of fitting complex datasets. Decision Trees are considered as white-box models because their decisions are intuitive and easy to interpret. This algorithm is also widely used in both classification and regression (Geron, 2015). The following cost function is minimized to determine the subsets and their thresholds:

$$\begin{aligned} J &= \frac{m_{\text{left}}}{m} G_{\text{left}} + \frac{m_{\text{right}}}{m} G_{\text{right}} \\ G &= 1 - \sum_{k=1}^n p_k^2 \end{aligned} \tag{7}$$

$G_{\text{left/right}}$ : the impurity measurement of the left/right subset,

$m_{\text{left/right}}$ : the number of examples in the left/right subset.

where  $p_k$  is the ratio of subset  $k$  among the training data. If all training instances belong to the same subset, then the impurity measurement would equal to zero ( $G=0$ ). In this study, the Classification and Regression Tree (CART) algorithm is employed (Pedregosa et al., 2011). The algorithm splits various instances into several subsets and defines a threshold. The CART algorithm investigates greedily for the optimum split thresholds from the top level and continues the process to lower levels. The algorithm stops splitting once it reaches the maximum depth for the decision (regression) tree, or the impurity measurements are not reduced by splitting the instances.

### 2.3.3 Extreme Gradient Boosting (XGBoost)

Extreme gradient boosting is the specific implementation of gradient boosting methods (GBM) for classification and regression problems. The concept and formulation of XGBoost and GBM are similar, while the main feature of the XGBoost is the robust calculation of the gradient to minimise the loss function. The XGBoost (and GBM in general) works based on the ensemble learner principle and iteratively combines a set of weak learners  $h_m(x)$  into a single learner as follows:

$$F(x) = \sum_{m=1}^M \gamma_m h_m(x) \quad (8)$$

The aim of the approach is to weight model's outcome at the iteration of  $m$  based on previous iterations  $m-1$ . The predictions generated by this base learner are weighted by a constant  $\gamma_m$  and minimises the average value of the loss function  $L(y, F(x))$  on the training set ( $i \in (1, n)$ ).

$$\gamma_m = \operatorname{argmax} \sum_{i=1}^n L(y_i, F_{m-1}(x_i) + \gamma h_m(x_i)) \quad (9)$$

The XGBoost model needs parameter tuning to enhance and fully leverage its power. We use and compare the performance of two commonly used boosters, tree and linear models.

### 2.3.4 ARMA

We used the ARMA model as a traditional time series model only for the demand prediction (not for the incident classification). The ARMA-family models include an autoregressive (AR) and a moving average (MA) part. The models predict the main variable for one or more discrete

time intervals by using the time-lagged values. Therefore, the historical variation of the variable along with subtle time-series dynamics is handled. The ARMA model with “ $p$  autoregressive terms” and “ $q$  moving-average terms” takes the following form:

$$\phi(L)x_i^t = \theta(L)\epsilon^t + \delta \quad (10)$$

where  $\phi(L)$  and  $\theta(L)$  are respectively the autoregressive operator on variables and the autoregressive operator on residuals,  $L$  is the backshift (lag) operator ( $Lx_i^t = x_i^{t-1}$ ),  $\epsilon_t$  represents the residuals between the real and the estimated variable at time  $t$ , and  $\delta$  is a constant.  $\theta(L)$  and  $\phi(L)$  can be presented as a polynomial relationship in the lag operator, defined as:

$$\begin{aligned} \theta(L) &= 1 - \theta_1 L - \dots - \theta_q L^q, \\ \phi(L) &= 1 - \phi_1 L - \dots - \phi_p L^p \end{aligned} \quad (11)$$

Consequently, if Eq.11 is replaced in Eq.10, we have:

$$(1 - \phi_1 L - \dots - \phi_p L^p)x_i^t = (1 - \theta_1 L - \dots - \theta_q L^q)\epsilon^t + \delta \quad (12)$$

Producing an ARMA model requires defining parameters  $p$  and  $q$  in order to specify  $\theta(L)$  and  $\phi(L)$ . Identification of  $p$  and  $q$  terms involves investigating a tentative formulation for the model as a starting point. After the general model is specified, the  $\theta(L)$  and  $\phi(L)$  are estimated using the least-squares method.

### 3. Numerical results

#### 3.1 Study Area

This study evaluates the proposed framework models for one of the major subnetworks in Sydney, stretching alongside the Victoria Road corridor from CBD to western city (see Figure 2). The subnetwork includes 1,310 links and 428 nodes. The General Transit Feed Specification (GTFS) data is used to import public transport information such as bus time schedules, lines, and bus stop data. There are 81 signalized intersections with the adaptive SCATS control system running. The link traffic counts obtained from the SCATS detectors are aggregated in

15-min time intervals. Most of the SCATS signals are located throughout the main corridor and near the Sydney CBD. The simulation is conducted for 4-h morning peak hours from 6:00 to 10:00 AM. using AIMSUN microscopic simulation model. AIMSUN is a discrete-event simulation tool established based on car-following and lane-changing models (Aimsun, 2013). Therefore, detailed traffic phenomena such as congestion propagation and dissipation of queues are simulated over time. We used a modified multinomial logit model as an advanced stochastic route choice model (Aimsun, 2013). Maximum five shortest paths are calculated using Dijkstra’s label-setting algorithm. The probability of choosing a path  $k$  is then calculated according to a utility function of each path.



Figure 2. Victoria corridor sub-network. Green points show signals equipped with SCATS count detectors (measured traffic data). Red line demonstrates the main Victoria corridor.

In Sydney, the signal controls are working with SCATS to reduce delay and make the transport system more efficient. Signal configurations such as signal group, phases and detector IDs are set in the model based on the SCATS configuration. Since the SCATS signals are adaptive and the adjacent signals are synchronized, it is very challenging to model such a complex system. To deal with this issue, we integrate our traffic simulation with SCATS using SCATSIM plug-in architecture as detailed in (Morgan and Veysey, 2013). We observe that our simulation model was spreading over 3 main SCATS regions (Ultimo, Rozelle and Ryde). A total of 81 controlled intersections were configured to replicate SCATS adaptive signal control in either isolated or coupled modes (sub-systems made of several intersections can be coupled together to optimise traffic signal timings). The entire architecture required several data sources such as the graphical layout of each intersection according to each region definition, the strategic inputs, signal timings, signal coordination, the action list for a particular time of the day routings, the controller information as well as the local time settings.

The SCATSIM and microsimulation model work in parallel and exchange information on a second by second loop: when cars arrive near the detectors placed for each lane in the simulation model, a message is being exchanged with the SCATSIM control architecture to communicate the occupancy of each lane. Based on incoming traffic volumes detected at each detector, SCATSIM would then adjust and modify the signal phase durations and send back to the simulation the new red/green/yellow times that the model should follow. A snapshot of a SCATSIM controlled intersection from our model is provided in Figure 3 where the Central manager and SCATS region modules are working correctly (green menu blocks on the SCATS Access interface are always activated) while detectors are detecting correctly incoming cars (see blue-highlighted detectors 1-4, 9, 10, and 7). Several verifications have been undertaken to monitor the total phase and cycle duration during the simulation. As a result, the real-life adaptive SCATS control logic is continuously applied to the simulated vehicles in our AIMSUN microsimulation model, making the results follow similar real-life traffic conditions.

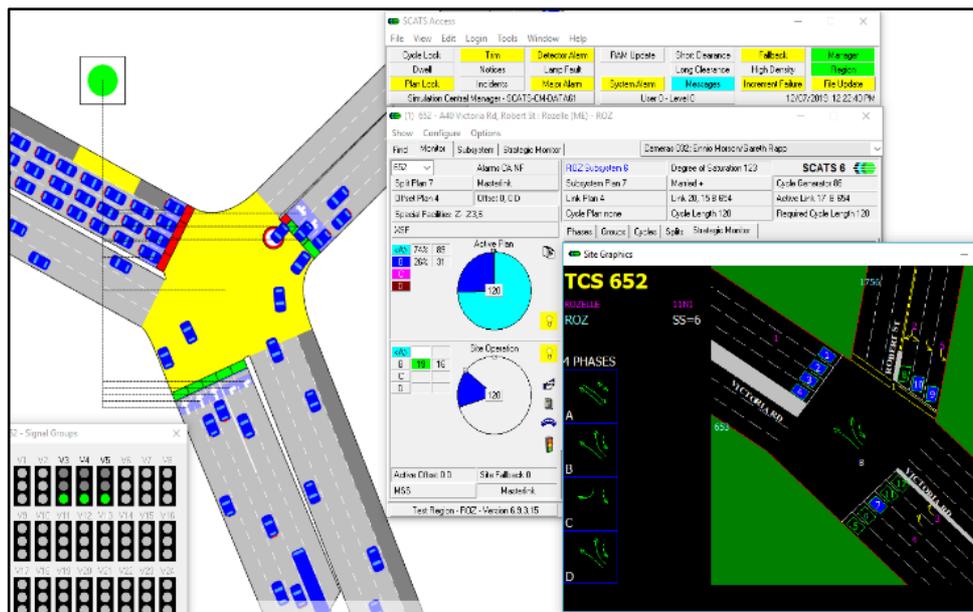


Figure 3. Example of SCATSIM controlled intersection inside the microsimulation model.

### 3.2 Demand estimation and prediction

The initial demand used in our study was obtained from the Sydney strategic model received from Transport for NSW, Australia. The extracted demand for the subnetwork contains 1,262 non-zero origin and destination pairs and around 150,000 travellers who are commuting in the area during 4-h morning peak hours on regular working days. The total number of travellers suffers daily changes and declines to less than 100,000 on weekends. We select October 2017

for this study results as the traffic count data and incident data are available almost every day of the month. To compare day-to-day traffic changes, we analyse the overall traffic counts pattern for all days in October as showcased in Figure 4. We observe that during morning peak hours, the traffic flow can reach almost 40,000 vehicles per every 15-min time interval compared to the weekend when this number can be counted in the 20,000s. Please note that the total count across the network is not a good indicator of the total number of vehicles travelling in the network because the same vehicle can be counted multiple times along its route. From Figure 4, we selected Thursday 12<sup>th</sup> October as a typical weekday.

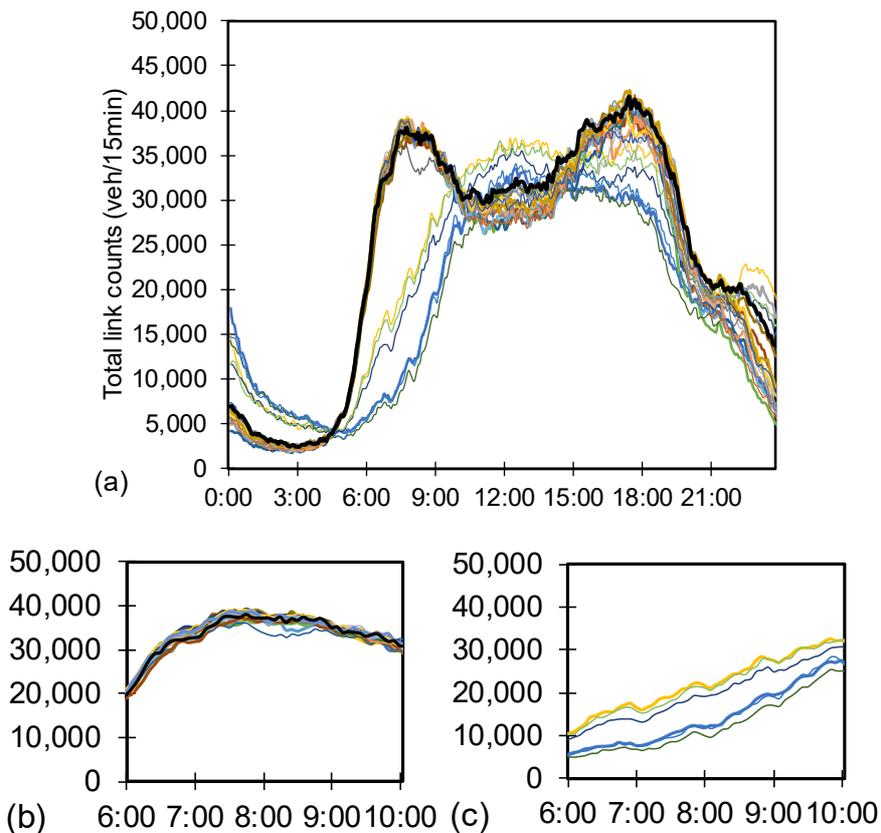


Figure 4. (a) Total 24-hour link counts obtained from loop detectors in the Victoria area in different days within a month, morning peak hours in (b) weekdays, and (c) weekends.

Since the uncertainty around bus frequency obtained from GTFS is low, we consider only private vehicles as a part of the OD estimation problem. We conduct our OD estimation process in two stages. At the first stage, we estimate time-dependent OD demand matrices based on the selected typical weekday link count and a priori demand data. A priori static OD demand obtained from the strategic model is time-sliced into sixteen 15-minute OD demand matrices for a 4-hour simulation during the peak period. Then, the bilevel optimization problem

formulated in Eq.1 is iteratively solved for the whole 4-hour morning peak period. This stage is done in an offline manner and demand flows at different time intervals are estimated. The only parameter in Eq.1 that must be set before executing the OD estimation problem is the demand reliability weight, which influences the accuracy of results. Previous studies proposed ranges based on the accuracy of the initial demand, the size of the network, and the simulation time window (Verbas et al., 2011). After various tests, we set this parameter to 0.95 for this case study.

We next execute our proposed OD estimation in a 15-min rolling-horizon procedure. In other words, the time-dependent OD demand matrices estimated from the first stage are adjusted every 15-min time interval based on the most recent observed traffic counts. Note that the input OD matrices for the second stage are more reliable than the inputs to the previous stage since they have already been adjusted. We increase the reliability weight ( $\omega$ ) value from 0.95 to 0.99. Adding higher weight for demand deviation also helps us to avoid overfitting the simulation model based on possible noisy count data. The microsimulation is executed for four hours of morning peak with a 15-min network warm-up (5:45- 6:00 AM). We used the coefficient of determination ( $R^2$ ), mean absolute error (MAE) and GEH (TfL, 2010) as the common goodness of fit criteria to evaluate the simulation results based on link traffic volumes. Note that GEH is tailored to hourly flows, and we aggregate every four 15-min traffic volumes to calculate GEH.

$$\begin{aligned}
R^2 &= 1 - \frac{\sum_{a=1}^A (\hat{y}_a^t - y_a^t)^2}{\sum_{a=1}^A (\hat{y}_a^t - (\frac{1}{A} \sum_{a=1}^A y_a^t))^2} \\
MAE &= \frac{1}{A} \sum_{a=1}^A |\hat{y}_a^t - y_a^t| \\
GEH_a &= \sqrt{\frac{2(\hat{y}_a^t - y_a^t)^2}{(\hat{y}_a^t + y_a^t)}}
\end{aligned} \tag{13}$$

Table 1 presents how the accuracy of the simulated traffic volumes increases through the two-stage OD estimation applications. Results show that MAE improves by about 42% and 27%, respectively, after implementing OD demand estimation process at the first and second stages. In general, a GEH value under 5 is regarded as a good fit; between 5 and 10 implies the measurement site needs further investigation, and a value greater than 10 implies a significant error (TfL, 2010). As can be seen, at the end of the OD estimation process, all links among 252 observed links have  $GEH < 10$ , and almost 90% of links have  $GEH < 5$ .

Table 1. The goodness of fit before and after the OD estimation.

| OD matrices                    | No. links<br>GEH<5 | No. links<br>GEH<10 | MAE<br>(veh/h) | R2   | Regression<br>line |
|--------------------------------|--------------------|---------------------|----------------|------|--------------------|
| Before OD estimation           | 189                | 243                 | 97             | 0.97 | Y=0.98X            |
| After OD estimation<br>stage 1 | 228                | 251                 | 56             | 0.99 | Y=1.00X            |
| After OD estimation<br>stage 2 | 231                | 252                 | 41             | 0.99 | Y=1.00X            |

Similar work is conducted for other days of October 2017 to archive day-to-day OD demand matrices. Figure 5 displays day-to-day variation in the estimated demand, which follow regular AM higher peak trends after 8 AM and lower values very early morning at 6 AM.

The scatter plots in Figure 6 shows the simulated and observed traffic flows after the OD estimation applications and we can observe very high  $R^2$  values for all hourly demands (lowest  $R^2$  is 0.98 while highest is almost close to perfection at 0.99 from 7–8 AM – see Figure 6 (a) – (d)). Figure 6 showcases that most GEH values that we have obtained are less than 5, which confirms the validity and accuracy of the model.

The estimated OD demand matrices were archived and used for the demand prediction model. As mentioned earlier, there are 1,262 OD pairs with various demand profiles in the study network. The OD flows vary from few trips between local OD pairs to hundreds of trips between two main corridor ends. Therefore, it is necessary to consider an ML model for each OD pair trained on previous historical OD pairs. We consider the corresponding OD flow values from the 5 previous working days. There is a tradeoff between the size of the database and the accuracy of the prediction. More historical data may be useful, but it increases the computational time, especially for real-time forecasting.

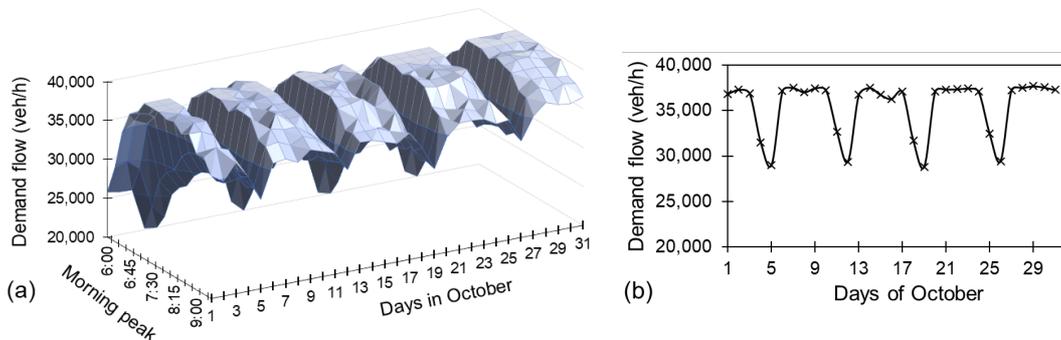


Figure 5. Day-to-day total estimated demand flow profiles: (a) in a 3D, (b) in 2D plotting.

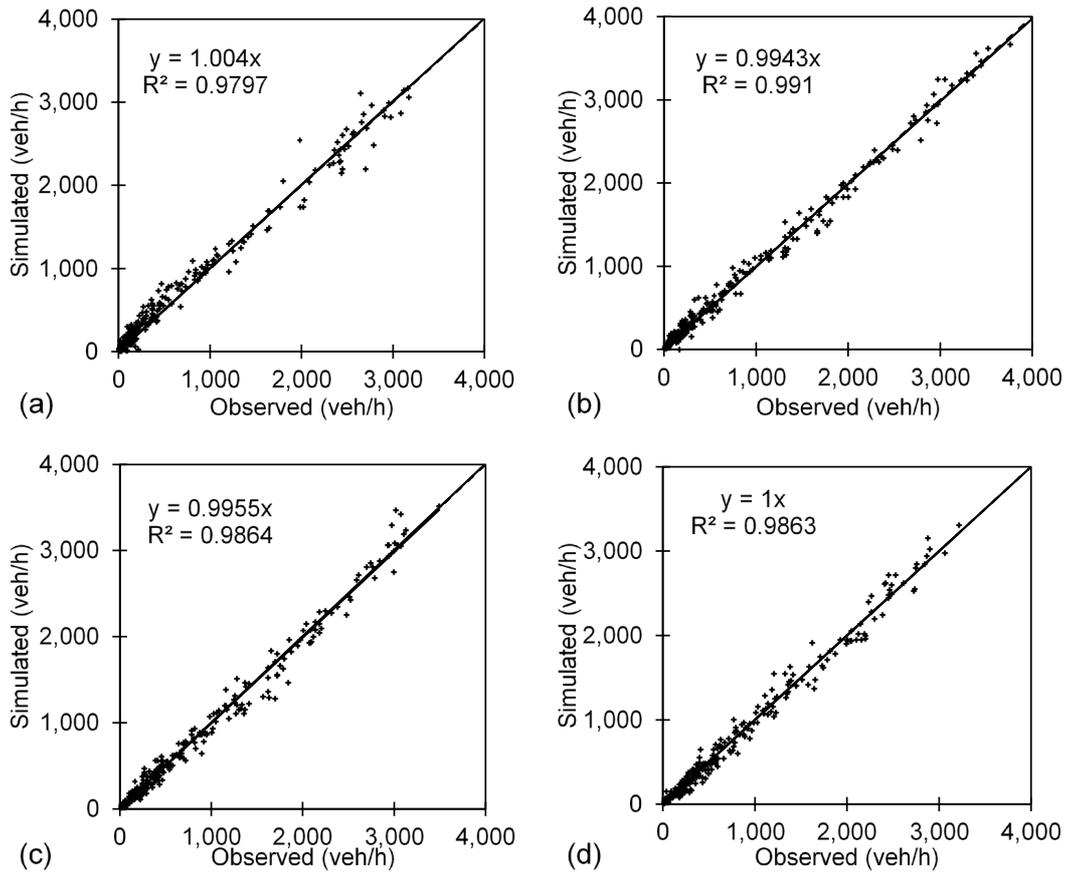


Figure 6. Simulated versus observed traffic flows after the rolling horizon OD estimation (a) 6:00-7:00 am, (b) 7:00-8:00 am, (c) 8:00-9:00 am and (d) 9:00-10:00 am

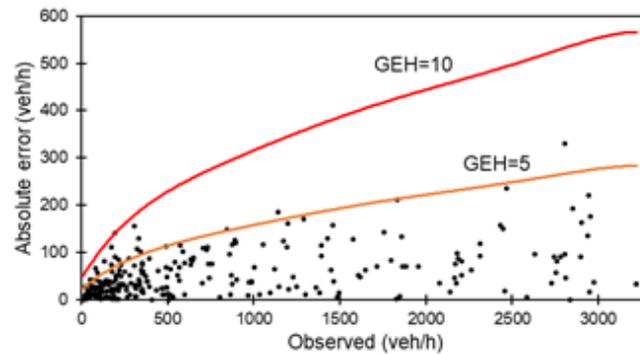


Figure 7. Absolute flow errors versus observed flows.

The features we consider for our models are: three time-lagged demand data ( $x_i^{t-2}$ ,  $x_i^{t-1}$ ,  $x_i^t$ ), the time interval of the prediction ( $t$ ) and the flow direction (inbound and outbound). We adopt DT, SVR, and the traditional ARMA models for predicting the OD flows for the next 15, 30, 45 and 60 min. The experiments were conducted using the Scikit tool and Stat Python libraries (Pedregosa et al., 2011). The results of each approach are presented in Table 2 and evaluated against the total error and MAE.

Firstly, as a baseline approach, we assume that we do not have any model to predict the demand and that previous historical demand will be the same even if disruptions may appear (this is a baseline assumption for comparison purposes). Therefore, the latest demand flow is considered for the next time intervals. As can be seen, by extending the prediction time intervals (from 15 min to 60 min), the prediction error grows quickly (from 1.37 to 1.85).

Table 2. Predicted demand using different predictors.

| Predictor       | Prediction window |      |             |      |             |      |             |      |
|-----------------|-------------------|------|-------------|------|-------------|------|-------------|------|
|                 | 15 min            |      | 30 min      |      | 45 min      |      | 60 min      |      |
|                 | Total error       | MAE  | Total error | MAE  | Total error | MAE  | Total error | MAE  |
| <b>Baseline</b> | 1283              | 1.37 | 2785        | 1.49 | 4519        | 1.61 | 6953        | 1.85 |
| <b>DT</b>       |                   |      |             |      |             |      |             |      |
| Max=2           | 762               | 0.81 | 2785        | 0.79 | 2278        | 0.81 | 3077        | 0.82 |
| Max=5           | 606               | 0.65 | 1152        | 0.61 | 1743        | 0.62 | 2533        | 0.68 |
| unconstraint    | 660               | 0.70 | 1216        | 0.65 | 1816        | 0.65 | 2645        | 0.70 |
| <b>SVR</b>      |                   |      |             |      |             |      |             |      |
| RBF (C=0.1)     | 1381              | 1.47 | 2613        | 1.40 | 3738        | 1.33 | 4797        | 1.28 |
| Sigmoid (C=0.1) | 1592              | 1.71 | 3017        | 1.67 | 4519        | 1.54 | 5545        | 1.48 |
| Linear (C=0.1)  | 832               | 0.89 | 1676        | 0.89 | 2462        | 0.87 | 3327        | 0.89 |
| Linear (C=1.0)  | 852               | 0.91 | 1693        | 0.90 | 2490        | 0.89 | 3376        | 0.90 |
| <b>ARMA</b>     |                   |      |             |      |             |      |             |      |
| (1,0,0)         | 869               | 0.93 | 1652        | 0.88 | 2459        | 0.87 | 3377        | 0.90 |
| (2,0,0)         | 883               | 0.94 | 1678        | 0.90 | 2517        | 0.89 | 3509        | 0.93 |
| (0,0,1)         | 1009              | 1.08 | 1876        | 1.00 | 2762        | 0.98 | 3864        | 1.03 |
| <b>XGBoost</b>  |                   |      |             |      |             |      |             |      |
| Tree            | 556               | 0.59 | 1033        | 0.55 | 1579        | 0.56 | 2237        | 0.60 |
| Linear          | 1141              | 1.21 | 2336        | 1.24 | 3471        | 1.23 | 4640        | 1.24 |

Secondly, we investigate the performance of the decision tree model. The maximum depth of the decision tree is one of the most important parameters that affect the prediction accuracy. Therefore, we chose three different values for this parameter to investigate the sensitivity of the model (maximum depth of 2, 5 or unconstrained). When the constraint is too strict (e.g., max depth=2), the predictor is too simplistic to predict traffic demand accurately. In contrast, with no constraint (no max depth), the prediction error increases showing the predictor overfitting the training data. This result demonstrates that the optimum depth of the tree is a critical parameter, which can be optimized.

Thirdly, the performance of the SVR model is explored by using different kernel functions. We tried several combinations (the radial basis function (RBF), the sigmoid function, and the linear function) and compared the results through a cross-validation approach. As can be seen, the first two models fail to predict the demand flows successfully, and the error is higher than the

baseline (e.g. 1.47 and 1.71 to 1.37). The best result for SVR prediction was obtained when using the Linear kernel (with the critical parameter of  $C=0.1$ ) for a 45min prediction in the future. This seems to perform similarly to predictions for the next 15 or 60min time-interval. However, overall, the accuracy results for SVR are significantly worse than those of DT prediction.

Fourthly, we present the ARMA model results, for which we need to determine the  $p$  and  $q$  parameters. Defining various OD pair parameters requires a vast amount of modelling effort. We consider the specifications of a unique model for all OD demands and estimate the factors of each OD pair. To identify  $p$  and  $q$ , we test various combinations of these parameters on all OD demand series ((1,0,0), (2,0,0) and (0,0,1)). Overall, the performance of the traditional AR and MA models show a medium level of accuracy compared to the others and the MAE errors are similar to those of SVR only for a 45min prediction.

Lastly, we used the powerful XGBoost predictor and compared its performance based on two boosters; tree and linear. Unlike the linear model, the tree-based model can model non-linear relations and shows high performance for the predictor from all our data-driven modelling approaches. The estimated MAEs for the tree-based GXBoost model in different prediction time windows range from 0.55 to 0.60. We conclude that the GXBoost method with tree booster is outperforming all other models and represent our final choice for conducting the rest of the experiments presented in this paper.

### ***3.3 Incident impact analysis***

For evaluating the potential of our proposed framework, we consider an actual reported incident along Victoria Rd with characteristics showcased in Table 3 (as received from the real-life incident stream in the TMC). The incident took place at 7:58 AM on the 11<sup>th</sup> of October 2017 and affected both directions. It also took place on the main road and according to the reported data, it impacted all lanes in both directions. No further details about how the accident affected the area exist in the database. Therefore, different interpretations of affected lanes can be considered. For example, the accident may have physically blocked all the lanes or caused some speed reduction for crossing vehicles. Since it is unlikely that all lanes are entirely blocked for half an hour in two directions, we assume only two lanes were blocked, and vehicles could slowly cross the accident area with some lane-changing near the accident. Therefore, the corresponding affected link IDs from the traffic simulation are determined based on the accident location. Then, an incident scenario containing the number of blocked lanes,

start time and incident duration is generated. Next, the Decision-Tree demand prediction module is triggered to forecast the travel demand starting from 8:00 AM for the next hour. Subsequently, the simulation model is executed with consideration of the developed incident scenario.

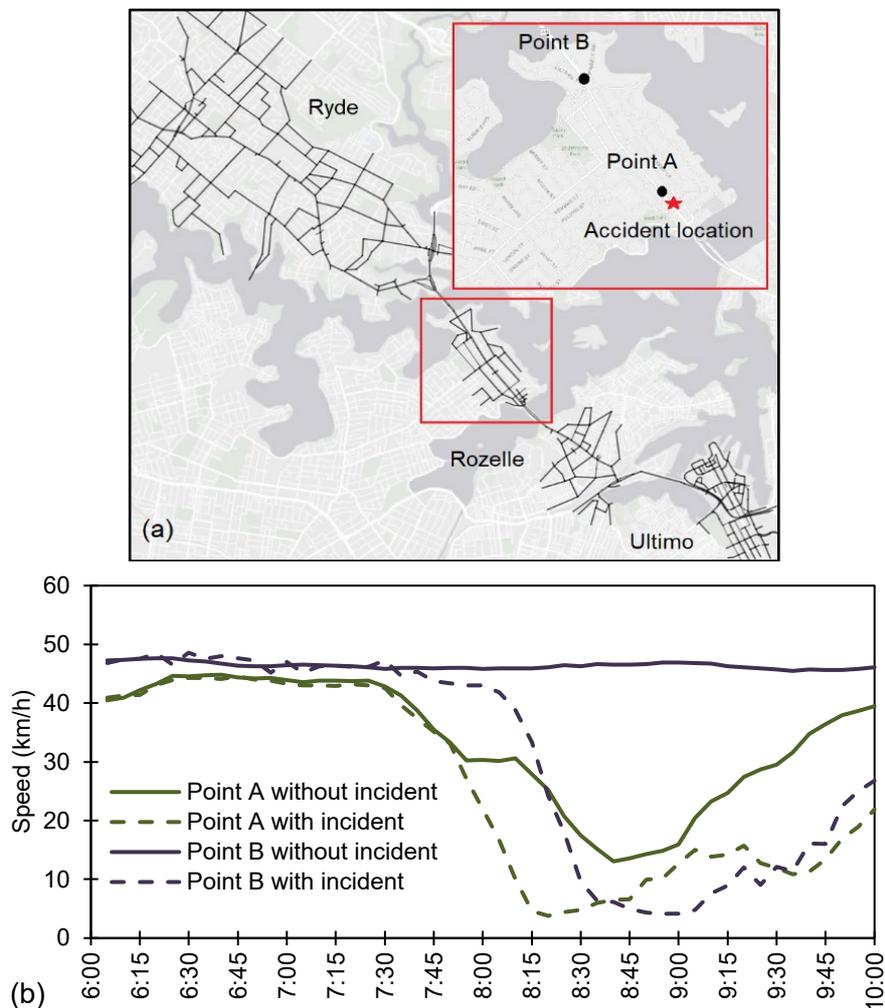
Table 3. CMCS incident data example

|                      |  |
|----------------------|--|
| (X, Y)               | (9684462, 4425168)   |
| Date                 | 11/10/2017   |
| Start Time Plain     | 7:58:19 AM   |
| End Time Plain       | 8:23:00 AM   |
| Incident Description | Accident: Accident   |
| Location Description | VICTORIA RD PARK AVE<br>DRUMMOYNE 2047 CANADA<br>BAY (LGA) NSW |
| Direction            | BOTH DIRECTIONS  |
| Affected Lanes       | ALL LANES  |
| Operator             | ----   |
| ICEMS Suburb         | DRUMMOYNE  |
| rowid                | A6C4B8F8-5617-4E12-B170-97323                                  |

We executed the simulation based on the above assumption and the simulated speeds are plotted for two different location points along the corridor: point A taken near the accident location and point B towards west, at 1,500 meters away from the accident (Fig 8 (a)). We explore the impact of the incident duration by analysing the total travel time in the network and the extra delay is caused by comparing it to what would have happened in the network if no accident would have been reported. Figure 8 (b) shows the severe drop in the speed near the incident location (Point A) and, consequently, lags further at Point B. Next, we evaluated the simulated travel time obtained for the eastbound direction along the entire Victoria Rd corridor against real-life recorded Google travel time in Figure 8 (c). We calculated experienced travel time for the entire corridor. The experienced simulated travel time (STT) and Google travel time (GTT) calculation accounts for the time required for traversing upstream links (in a duration labelled  $\Delta t$ ), and consider the downstream link  $l$  travel time ( $tt_l$ ) based on the time of entering that downstream link ( $tt_l(\tau + \Delta t)$ ). Thus, we used Eq.14 to compute the (S/GTT) at time  $\tau$  ( $\tau \in (6:00 \text{ AM}-10:00 \text{ AM})$ ):

$$S/GTT(\tau) = \sum_{l \in L} tt_l(\tau + \Delta t) \quad (14)$$

where  $L$  is the set of all links in the eastbound direction. Figure 8 (c) shows that the STT changes pattern for a typical day is similar to the associated measured GTT (which is another confirmation of simulation validation and accuracy). The corridor travel time at peak hour 8:00–9:00 AM is almost twice greater than the corridor travel time in the early morning. Moreover, one can observe that the reported incident at 07:58 AM induces an increase of almost 13 min on the eastbound direction travel time (from 31-44 min). The simulated incident follows very closely the delay reflected by Google Travel time, which confirms that our approach for prediction using data-driven and traffic simulation modelling is capable of accurately providing good insights on the impact of future incidents on the overall traffic condition in a network.



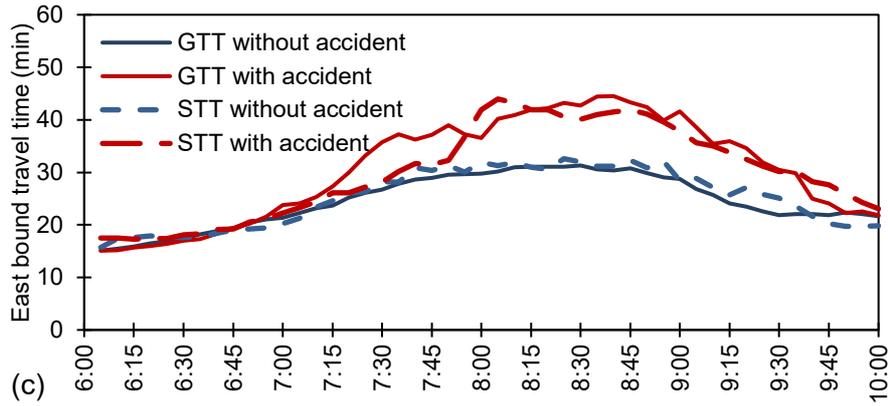


Figure 8. (a) Accident location (b) Speed profiles for Point A and B. (c) Eastbound journey times with and without the incident.

#### 4. Conclusion

Managing an incident situation effectively is one of the critical challenges that TMCs deal with daily. Methodological advances in both data-driven and computer-based simulation modelling provide a unique opportunity to predict traffic conditions accurately in real-time. This paper presents a framework for incident management by using integrated data-driven and traffic simulation models. We first introduced a generic demand estimation and prediction model that provides the essential input for the traffic simulation model to forecast the traffic features under incident conditions.

The proposed approach initially keeps the traffic model updated based on the most recent observed link traffic count and estimates OD flows through a rolling horizon bi-level optimisation. Then the updated flows are used for demand prediction for the next time interval once an incident happens. We showcased that the XGBoost method outperformed the baseline and the other tested models (SVR, DTs and ARMA) in prediction accuracy. Finally, we investigated an incident's impact by using the proposed framework outputting the predicted travel time/delay along the affected corridor.

Several limitations of this study will be addressed in our future work:

- We assumed that travellers respond to bad traffic conditions by changing routes in peak hours. However, for critical traffic disruptions, we should consider mode shifting and trip cancellation in short-term traffic forecasting.
- While the high efficiency of some well-established machine learning methods was presented in this study, recent advances in graph-based machine learning and deep

learning models could help to capture the spatial-temporal correlations of road networks in future studies.

- Although the simulation network provides the commuters with some re-routing options, the selected area had limited major parallel routes to consider strategic re-route choices of travellers under severe incident conditions. We plan to expand our simulation modelling to larger sub-network areas.
- For the real-time incident impact application, incident time duration should be predicted based on the incident features. Another AI engine will be added to the incident management platform to provide an incident duration estimation for the traffic simulation when an incident occurs. This estimation can be updated with more incident data.
- In order for our proposed modelling framework to be used in real-life traffic management operations, it requires exact details of the incident location (inside the intersection or not and beginning and ending of road link), the length of the affected incident area and the exact number of blocked lanes. This can be challenging for operational centres to provide in real-time.
- Increasing the number of incident cases could improve our understanding of the approach. However, the quality of the incident data would need to be managed.
- The traffic model includes private cars and buses, however, other types of traffic modes such as taxi, freight and active modes can be exclusively considered. This would help us replicate the simulated congestion more accurately.

### **Acknowledgements:**

The authors of this paper are highly grateful for the data provided by TfNSW Sydney, Australia. The work has been part of the Data61-UTS-SUT project collaboration under the project ID: LP180100114.

### **5. References**

- Aimsun, 2013. Aimsun 8 Dynamic Simulators Users ' Manual.
- Antoniou, C., Barceló, J., Breen, M., Bullejos, M., Casas, J., Cipriani, E., Ciuffo, B., Djukic, T., Hoogendoorn, S., Marzano, V., 2016. Towards a generic benchmarking platform for origin–destination flows estimation/updates algorithms: Design, demonstration and validation. *Transp. Res. Part C Emerg. Technol.* 66, 79–98.
- Ashok, K. and Ben-Akiva, M.E., 2000. Alternative approaches for real-time estimation and prediction of time-dependent origin–destination flows. *Transportation science*, 34(1), pp.21-36.
- Balakrishna, R., Morgan, D., Yang, Q., Slavin, H., 2012. Comparison of simulation-based dynamic traffic assignment approaches for planning and operations management, in: *Vortrag: 91st Annual Meeting of the*

- Transportation Research Board, Washington, DC.
- Cascetta, E., 2009. *Transportation systems analysis: models and applications*. Springer.
- Du, B., Chien, S., Lee, J., Spasovic, L., 2017. Predicting Freeway Work Zone Delays and Costs with a Hybrid Machine-Learning Model. *J. Adv. Transp.* 2017, 1–8.
- Frederix, R., Viti, F., Tampère, C.M.J., 2011. Dynamic origin–destination estimation in congested networks: theoretical findings and implications in practice. *Transportmetrica* 9935, 1–20.
- Geron, A., 2015. *Hands-on Machine Learning with Scikit-Learn & Tensorflow*, Early Access Book. O'Reilly Media.
- Hawas, Y.E., 2007. A microscopic simulation model for incident modeling in urban networks. *Transp. Plan. Technol.* 30, 289–309.
- Karim, A., Adeli, H., 2003. Radial basis function neural network for work zone capacity and queue estimation. *J. Transp. Eng.* 129, 494–503.
- Kim, H., Nam, D., Suh, W., Cheon, S.H., 2018. Origin-destination trip table estimation based on subarea network OD flow and vehicle trajectory data. *Transp. Plan. Technol.* 41, 265–285.
- Li, X., Lam, W.H.K., Shao, H., Gao, Z., 2015. Dynamic modelling of traffic incident impacts on network reliability. *Transp. A Transp. Sci.* 11, 856–872.
- Liu, Y., Zou, B., Ni, A., Gao, L., Zhang, C., 2020. Calibrating microscopic traffic simulators using machine learning and particle swarm optimization. *Transp. Lett.* 1–13.
- Morgan, C.J., Veysey, M., 2013. *Traffic Modelling Guidelines*, NSW, Roads and Maritime Services.
- Nguyen, H., Bentley, C., Kieu, L.M., Fu, Y., Cai, C., 2019. Deep Learning System for Travel Speed Predictions on Multiple Arterial Road Segments. *Transp. Res. Rec.* 036119
- Nguyen, H., Cai, C., Chen, F., 2017. Automatic classification of traffic incident's severity using machine learning approaches. *IET Intell. Transp. Syst.* 11, 615–623.
- Oh, S., Byon, Y.J., Jang, K., Yeo, H., 2018. Short-term travel-time prediction on highway: A review on model-based approach. *KSCE J. Civ. Eng.* 22, 298–310.
- Oh, S., Byon, Y.J., Jang, K., Yeo, H., 2015. Short-term Travel-time Prediction on Highway: A Review of the Data-driven Approach. *Transp. Rev.* 35, 4–32.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., 2011. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* 12, 2825–2830.
- Polson, N.G., Sokolov, V.O., 2017. Deep learning for short-term traffic flow prediction. *Transp. Res. Part C Emerg. Technol.* 79, 1–17.
- Rao, W., Wu, Y.J., Xia, J., Ou, J., Kluger, R., 2018. Origin-destination pattern estimation based on trajectory reconstruction using automatic license plate recognition data. *Transp. Res. Part C Emerg. Technol.* 95, 29–46.
- Shafiei, M., Nazemi, M., Seyedabrishami, S., 2015. Estimating time-dependent origin–destination demand from traffic counts: extended gradient method. *Transp. Lett.* 7, 210–218.
- Shafiei, S., Mihăiță, A.-S., Nguyen, H., Bentley, C., Cai, C., 2020. Short-term traffic prediction under non-recurrent incident conditions integrating datadriven models and traffic simulation, in: 99th Annual Meeting of Transportation Research Board, Washington, D.C.
- Shafiei, S., Saberi, M., Zockaie, A., Sarvi, M., 2017. A Sensitivity-Based Linear Approximation Method to Estimate Time-Dependent Origin-Destination Demand in Congested Networks. *Transp. Res. Rec.* 2669, 72–79.
- Tedjopurnomo, D.A., Bao, Z., Zheng, B., Choudhury, F., Qin, A.K., 2020. A Survey on Modern Deep Neural Network for Traffic Prediction: Trends, Methods and Challenges. *IEEE Trans. Knowl. Data Eng.* 13-1.
- Tercan, E., Beşdok, E., Tapkın, S., 2020. Heuristic Modelling of traffic accident characteristics. *Transp. Lett.* 1–9.
- TfL, 2010. *Traffic Modelling Guidelines TfL Traffic Manager and Network Performance Best Practice Version 3.*, 184.
- Verbas, İ.Ö., Mahmassani, H.S., Zhang, K., 2011. Time-Dependent Origin-Destination Demand Estimation. *Transp. Res. Rec. J. Transp. Res. Board* 2263, 45–56.
- Vlahogianni, E.I., Karlaftis, M.G., Golias, J.C., 2014. Short-term traffic forecasting: Where we are and where we're going. *Transp. Res. Part C Emerg. Technol.* 43, 3–19.
- Vlahogianni, E.I., Karlaftis, M.G., Golias, J.C., 2007. Spatio-temporal short-term urban traffic volume forecasting using genetically optimized modular networks. *Comput. Civ. Infrastruct. Eng.* 22, 317–325.
- Wang, D., Fu, F., Luo, X., Jin, S., Ma, D., 2016. Travel time estimation method for urban road based on traffic stream directions. *Transp. A Transp. Sci.* 12, 479–503.
- Wen, Y., Balakrishna, R., Ben-Akiva, M., Smith, S., 2006. Online deployment of dynamic traffic assignment: architecture and run-time management, in: *IEE Proceedings-Intelligent Transport Systems*. IET, pp. 76–84.
- Wu, C.H., Wei, C.C., Su, D.C., Chang, M.H., Ho, J.M., 2003. Travel time prediction with support vector regression. *IEEE Conf. Intell. Transp. Syst. Proceedings, ITSC 2*, 1438–1442.
- Xu, D., Wang, Y., Peng, P., Beilun, S., Deng, Z., Guo, H., 2018. Real-time road traffic state prediction based on

- kernel-KNN. *Transp. A Transp. Sci.* 9935, 1–15.
- Yao, B., Chen, C., Cao, Q., Jin, L., Zhang, M., Zhu, H., Yu, B., 2017. Short-Term Traffic Speed Prediction for an Urban Corridor. *Comput. Civ. Infrastruct. Eng.* 32, 154–169.
- Zhang, Y., Liu, Y., 2009. Traffic forecasting using least squares support vector machines. *Transportmetrica* 5, 193–213.
- Zhou, X., Mahmassani, H.S., 2007. A structural state space model for real-time traffic origin-destination demand estimation and prediction in a day-to-day learning framework. *Transp. Res. Part B Methodol.* 41, 823–840.
- Ziliaskopoulos, A.K., 2001. Foundations of Dynamic Traffic Assignment : The Past , the Present and the Future. *Networks Spat. Econ.* 1, 233–265.