Expert Systems With Applications 171 (2021) 114573

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

A neural network approach for traffic prediction and routing with missing data imputation for intelligent transportation system

Robin Kuok Cheong Chan, Joanne Mun-Yee Lim *, Rajendran Parthiban

School of Engineering, Monash University Malaysia, Selangor, Malaysia

ARTICLE INFO

Keywords:
Neural network
Traffic prediction
Traffic modelling
Data imputation
Rerouting system

ABSTRACT

A robust traffic rerouting system is important in traffic management, alongside an accurate traffic simulation model. However, missing data continues to be a problem as it will inevitably cause errors in predicting the congestion levels, resulting in a less efficient rerouting. The lack of a realistic traffic simulation also serves to hamper the development of a better traffic management system. As such, this paper aims to address both problems by proposing three solutions: (i) a traffic simulation that would model a live-traffic, (ii) a pheromone-based, neural network traffic prediction and rerouting system, and (iii) a missing data handling method utilising weighted historical data method named Weighted Missing Data Imputation (WEMDI). The traffic simulation model was benchmarked using Google Maps rerouting system. WEMDI was tested by comparing the performance of the rerouting system with and without WEMDI's integration for various levels of missing data. The results showed that the traffic simulation model displayed a high correlation to that of Google Maps, and the WEMDI-integrated system displayed 38% to 44% improvement in the related traffic factors, when compared to a situation with no rerouting system in place, and up to 19.39% increase in performance compared to the base rerouting system for missing data levels of 50%. The WEMDI system also displayed robustness in routing other locations, displaying a similarly high performance.

1. Introduction

As the population of the world increases, so does the number of vehicles used by that population. This increase in the number of vehicles leads to a rise in traffic congestions which in turn causes an increase in CO_2 emissions as shown in a study by Bharadwaj, Ballare, Rohit, and Chandel (2017) based the impact of traffic congestion in Mumbai.

In order to combat this issue, there is a need to further develop existing Intelligent Transport Systems (ITS). Various studies have been done with that in mind — such as traffic modelling (Idrissa, 2017; Mustapha & Nik Hashim, 2016; Valente, Avram, Machado, & Astilean, 2018), traffic forecasting and rerouting (Chon, Lim, Lun, & Yong, 2019; Soon, Lim, Parthiban, & Ho, 2019; Tan, Wu, Shen, Jin, & Ran, 2016). A traffic simulation software (Pell, Meingast, & Schauer, 2017) would be used to validate such studies as it provides a safe, inexpensive, and flexible environment for the user to test the proposed methods.

Traffic simulation continues to grow in importance in traffic studies related to road design and traffic light planning, which leads to a need to further improve the realism of the simulation. What comes to mind is to provide a simulation where the current state of the traffic of the desired location is copied and initialised within the traffic simulator. While there are studies into providing a real-time traffic situation for driving simulators as proposed by Maroto, Delso, Félez, & Cabanellas, 2006, and it is likely possible simulate a live traffic situation with the help of physical sensors and floating car data, to the knowledge of the authors, there are presently no attempts found to utilise live input data from available online traffic data sources. It is believed that a simulation that models a live traffic condition at a given time would help improve the reliability of proposed methods due to its increased realism in data imputation application. As traffic APIs such as Google Maps and Waze become more prevalent in society, the authors believe that such APIs would be able to provide the required traffic data that would help model a current traffic situation.

In order to reduce the travel time of vehicles, a lot of effort have been placed into researching ways to forecast traffic congestion and provide drivers with a more efficient route. Studies in recent years have seen many methods with regards to machine learning of which neural networks play a big part of (Qu, Li, Li, Ma, & Wang, 2019; Sun, Dubey, &

* Corresponding author.
E-mail addresses: rkcha5@student.monash.edu (R.K.C. Chan), Joanne.Lim@monash.edu (J.M.-Y. Lim), Rajendran.Parthiban@monash.edu (R. Parthiban).

https://doi.org/10.1016/j.eswa.2021.114573
Received 10 December 2019; Received in revised form 28 July 2020; Accepted 2 January 2021
Available online 6 January 2021
0957-4174/

White, 2018; Zhang & Zhang, 2016). With the advancement in ITS, data is continually obtained in real-time from various sources, resulting in a large amount of data. Neural networks will then face a problem of scalability when faced with such scenarios, this research proposes an online deep neural network that utilises a sliding window technique to learn a certain period's worth of data as a small batch and adapt accordingly. This congestion prediction system is paired with a routing mechanism and utilises a pheromone approach similar to the one introduced by Chon et al. (2019) to provide a complete package from traffic prediction to rerouting.

Meanwhile, as ITS relies heavily on data obtained from vehicles and the surrounding infrastructures, it is bound to face the problems of missing data. Missing data occurs when traffic sensors break down or if there is a transmission error between the vehicle and the traffic management system and there are many recent studies in traffic prediction aimed to resolve this issue as described in the literature review. Many proposed traffic prediction methods also face other problems such as requiring manual input of certain traffic information such as events (Sun et al., 2018) or a lack of computational power, making the scalability of the system an issue (Tan et al., 2016). While many methods have been introduced, the historical average method seems to have been ignored despite its potential. The proposed solution towards the missing data imputation problem is to pre-process the data with a weighted historical average method prior to being passed into a neural network for traffic congestion prediction.

As such, the main contributions of this paper are as follows. First, the traffic simulation is modelled with a live traffic situation using the data obtained from traffic APIs and benchmarked against Google Maps to ensure realistic traffic initialisation. Second, a pheromone-based, multi-factor online deep neural network traffic congestion prediction system is proposed that handles missing data by utilising a weighted average between the present and historical traffic data. This traffic prediction system would be used to determine the viability of the missing data handling method. Third, a rerouting system which uses the predicted congestion pheromone to reroute the vehicles to the route with the lowest overall cost is proposed to include a complete system inclusive of traffic prediction and rerouting.

The resultant rerouting system is tested using the developed traffic simulation for 3 different cases — No rerouting, standard rerouting, and rerouting with missing data handling.

The map used for the simulation is Singapore City Centre and the traffic simulation software used is Simulation of Urban MObility (SUMO) (Krajzewicz & Rossel, 2007). To further verify the robustness of the developed system, an urban map of Bukit Bintang in Malaysia was also used.

The following sections are as follows: Section 2 reviews literature related to the current work, Section 3 explains the principals and theory used in this paper, which is the car-following theory and Greenshields' model, Section 4 covers the methodology and the setup of the systems, Section 5 discusses and analyses the results, and Section 6 concludes the paper.

2. Literature review

The following subsections details the literature review of various topics researched by this paper.

2.1. Traffic prediction and routing

There have been many studies done in regards to traffic prediction, one of the most popular methods is the use of neural networks due to the capability of the neural network to identify and learn traffic behavioural patterns given enough variables to inspect. Qu et al. (2019) proposed a deep neural network that utilises contextual factors such as day of the week, weather, and seasons in order to predict the traffic flow. The proposed method showed a higher overall prediction accuracy than the conventional methods they compared it to but does not perform as well for low-demand periods, not to mention that gathering many contextual factors such as the weather may not always be feasible depending on the location.

Swarm-Intelligence method has also been proposed such as the one by Shang, Lin, Yang, Bing, and Zhou (2016) who proposed a particle swarm optimisation-based system. Such methods however, suffers from a large computation time due to the number of iterations that are required. This can be seen in a rendition of particle swarm optimisation by Zhang, Zou, and Shen (2018) which has shown that the time complexity for their algorithm is $O(M*N)$ where M is the population and N is the problem dimensions.

Methods utilising support vector machines (SVM) as shown by Chen, Li, Tian, Chen, and Wang (2012) are also another popular method in predicting traffic flow. However, SVM requires trial and errors in determining the necessary parameters to be effective, which poses a problem of uncertainty when developing the system.

Many of these methods use a static historical traffic data to train their model but in reality, it is not always possible to have the traffic data for training, so a traffic prediction system that is able to learn on-the-go using live traffic data is required.

There are just as many traffic studies done with regards to vehicle routing, which by itself has many different scenarios besides typical traffic situations such as for distribution logistics (Sarathi Barma, Dutta, & Mukherjee, 2019) which focuses on a Multi-depot vehicle routing problem or routing which focuses on the reduction in greenhouse gas emissions (Soon, Lim, Parthiban, et al., 2019). The common end goal of these studies is always to reduce the overall costs required for an agent (i.e. vehicles) to reach its destination. These costs are attached to the path taken and are calculated according to each studies' focus.

The proposed design utilises an online deep neural network that utilises the sliding window strategy for traffic prediction and routes vehicles through roads that are weighted based on two factors — predicted road travel time and the presently detected mean vehicle speed. These two features provide a scalable prediction system and a routing mechanism that takes into account the both the current and future condition of the road that can adapt to the traffic patterns with little to no prior training.

2.2. Missing data imputation

Recently, there have been studies involving the use of matrix/tensor completion with regards to imputation or recovery of missing data (Fan & Cheng, 2018; Yokota, Erem, Guler, Warfield, & Hontani, 2018; Yokota, Zhao, & Cichocki, 2016), where one such paper by Tan et al. (2016) uses dynamic tensor completion in predicting traffic flow and has demonstrated a higher accuracy when compared to the ARIMA model but faces a problem with scalability due to the large amount of data required.

Zhang and Zhang (2016) has done a comparative study between vector autoregression (VAR), general regression neural network (GRNN), and historical average (HA) forecasting methods with GRNN displaying the highest performance and robustness towards missing data. Meanwhile, Lan, Xu, Ma, & Li, 2020 proposed a missing data imputation method utilising a Bayesian network for imputing incomplete credit data which contains multiple variables.

From what was found so far, the historical average method has been deemed as underperforming in most cases, thereby not receiving much attention, which resulted in not many studies concerning it. To the best of the authors' knowledge, there has not been any research combining the use of weighted historical averages which utilises historical and current data together with neural networks. This paper intends to develop and improve upon a pheromone-based, multifactor neural network vehicle rerouting (MVR) algorithm inspired by Chon et al. (2019) and Soon, Lim, Parthiban, et al. (2019) by proposing a novel historical average method of weighing its input data based on the amount of missing data and averaging it out with its historical data. The proposed model utilises a deep neural network with sliding window technique instead of the LSTM approach to provide an online learning neural network that begins learning as the simulation begins — The neural network is initially untrained.

2.3. Significance of online traffic information

While there are many studies made in the effort to predict traffic accurately, many of these studies utilise traffic data obtained directly from physical sensors such as loop detectors or cameras as detailed by Panichpapiboon & Leakkaw, 2017. These methods would limit researchers when designing their proposed method to only a handful of locations. Not to mention traffic prediction that utilises prior events' information would not work for other places as it may not be available.

However, by relying on organisations with an already established traffic monitoring framework such as Google's Google Maps (Google, n.d.) or TomTom's (TomTom Developer, n.d.) and HERE Technologies' (HERE, n.d.) Traffic API , it is possible to collect traffic data on a global scale and in a uniform manner (i.e. The data obtainable would be similar no matter the location). It is also well known that map data can also be obtained via OpenStreetMap (OpenStreetMap, n.d.) which is an open-source website providing free geographic data.

This is one of the main reasons why the research conducted in this paper focuses on the use of online traffic information. HERE Traffic API was chosen for this paper as it is able to return all available road data belonging to a portion of a map given its coordinates based on the Mercator Projection (Google, 2019) which is a way of displaying the world map in 2D from its spherical shape and breaking it up into tiles. This is unlike Google Maps which can only return the congestion level of the road in colour form, HERE Traffic API returns raw traffic data, allowing for more detailed analysis and processing to be done. While TomTom TrafficAPI returns the raw data as well, it targets specific roads based on the coordinates instead of a map tile like HERE, making it less suitable to be used for the objective of the proposed simulation model.

2.4. Use of traffic simulation software

Agent-based modelling or traffic simulation software packages are tools used in traffic studies that mainly focuses on road designs such as road layouts as well as traffic light regulations, as well as traffic studies involving the validation of car-following models (Song, Wu, Xu, & Lin, 2015).

There are various commercially popular packages such as VISSIM (PTV Vissim, 2011), CORSIM(TSIS-CORSIM, n.d.), and PARAMICSParamics microsimulation, n.d.) available on the market which provides their software at a price.

Meanwhile, there are also open-source traffic simulators as well such as SUMO (Lopez et al., 2018), Repast Symphony Repast Simphony, n.d.), and TRANSIMS (Smith et al, n.d.).

The research conducted utilises SUMO as it is active in development, is well documented, and specialises in microscopic traffic studies. There are also traffic studies using this simulator such as an Intelligent Traffic Management System Framework by Akhter, Ahsan, Jafor, Quaderi, and Forhad (2020).

By utilising online traffic information obtained from HERE Traffic API together with the SUMO Graphical Interface, it is possible to recreate — to an extent — the current traffic situation as described by the data, thus providing a novel traffic simulation that has an increased realism which will no doubt be beneficial in future traffic research.

2.5. Literature review summary

Table 1 below provides an overview of the research gap covered by the literature reviews and the contributions of the proposed method with regards to address the issues.

Table 1
Broad view of the literature review, summarised in table form.

	Traffic Simulation	Traffic Prediction	Traffic Routing	Missing Data Imputation
Research Gaps	Minimal attempts found to simulate live traffic using online traffic API for traffic simulators	Majority of traffic prediction models suffers from scalability issues due to large amount of data and trains using static historical data.	Traffic routing models only considers predicted information and not current information. As predicted information is not entirely reliable, utilising current traffic data could help increase its	Historical average method not investigated despite its advantage in having a small computational complexity compared to other methods.
Contributions of Proposed Model	Proposed model simulates traffic based off live traffic data obtained via online traffic APIs to provide a more realistic traffic simulation	Proposed model solves scalability issues by introducing a sliding window technique to train an 'online' deep neural network — The neural network trains and adapts to traffic trends in real time.	Proposed model routes traffic by using weights to adjust the routes' cost based on predicted traffic data along with present	WEMDI is a historical average method utilising weights with regards to the amount of missing data missing from the current data. The data is pre-processed prior to being input into the neural network

3. Principals and theory

This section briefly explains the basics of traffic modelling theory as well as the traffic model used as reference in this paper — Greenshield's Macroscopic Stream Model.

3.1. Traffic modelling theory

Traffic modelling is used to derive a few relationships in order to quantify the key characteristics of traffic, namely: flow rate, density, and velocity. (Ako, Yahaya, Ako, Atoo, & Yusuf, 2017) These relationships are shown below:

$$q = \frac{n}{t} \tag{1}$$

$$k = \frac{n}{l} \tag{2}$$

$$q = ku \tag{3}$$

where q is the traffic flow – the vehicles arriving at a point per unit time – and n is the number of vehicles observed over a time interval, t, for Eq. (1).

Eq. (2) depicts the traffic density k, which is the number of vehicles occupying a roadway per unit of road length, l. Eq. (3) shows the relationship between q and k whereby the traffic flow is equal to the traffic density multiplied with the average velocity u.

3.2. Greenshields macroscopic stream model

One of the most prominent traffic model is the Greenshields model developed in 1934 where the speed-density relationship is linear and is governed by the following equation as described by Rakha & Crowther, 2002:

$$v = vf - \left(\frac{vf}{kj}\right)k \tag{4}$$

where *vf* is the free flow speed of the road, and *kj* is the jam density – The density at which the vehicles are at a complete stop. Substituting Eq. (3) into this equation results in Eq. (5) below:

$$q = vf.k - \left(\frac{vf}{kj}\right)k^2 \tag{5}$$

As shown by these two equations and mentioned by Rakha & Crowther, 2002, Greenshields Traffic Model describes the speed-flow relationship in a parabolic manner. This can be visualised through the graphs shown in Fig. 1.

Although it could be said to be less representative of how a true traffic behaves, the Greenshields traffic model has the least number of parameters that are needed to be configured to work, as compared to the 4-parameter Van Aerde model for example.

As the research done in this paper relies on the usage of online traffic data, many parameters are not available and hence, the Greenshield traffic model proved to be the simplest and most suitable model for the scope of this research.

4. Methodology

The following subsections describes the development of the traffic simulation, rerouting system, as well as the missing data handling method and how the simulations will be carried out.

4.1. Estimation of simulation parameters

As the simulation requires traffic flow, *q* which is a parameter that is not available from the API, the only way is to estimate traffic flow is by using traffic modelling theory. For the following derivations, Greenshields' traffic flow model (Greenshields, 1935) is used. Greenshields' model is still used as the foundation for other traffic model research papers such as the one by Hossain and Hasan (2019) due to its simplicity and having a decently accurate representation of traffic flow despite its shortcomings — Having low goodness-of-fit at low congestion levels (de Grange, Marechal, & González, 2019).

4.1.1. Derivations

Through reverse engineering of Greenshields' traffic model's equations using the data obtained by the HERE Traffic API, along with some assumptions, the equations are derived to model the current traffic situation.

The derivations are made and calculated with the following assumptions:

- Headways are assumed to follow a linear relationship as shown in a study by Sanik et al. (2016) between the speed and the driver's reaction time, $H = V*R + H_0$, where R is the reaction time. Although the constant term is the average length of the vehicle, studies have found that when $V = 0$ (Traffic is at a standstill), H_0 is different and therefore, we represent the constant as the standstill distance between the cars.
- H_0 is taken to be 2.7 m (~8ft) following a separate study of vehicle data in a standstill situation where the mean headway at standstill is between 8.12ft to 9.6 ft. (Houchin & Houchin, 2015)
- The reaction time, R is assumed to follow a controlled study of 2.3 s. (McGehee, Mazzae, & Baldwin, 2000)
- The only vehicles on the road are assumed to be cars, hence $L_{Vehicle}$ would equal to the average family car length, at 4.5 m.
- Roads with jam factors, *JF* (Degree of traffic jam) greater than zero are assumed to have cars linearly spaced (Based on the details above) from one another, filling up the entirety of the road and moving at the same speed.
- Only roads with *JF* greater than zero are taken into consideration, as other roads are assumed to have a traffic density of zero.

In order to describe the traffic congestion of a road, HERE Traffic API returns a value between 0 and 10 called Jam Factor (JF). It is important to note that this parameter is different from the traffic density, *k* mentioned in Greenshields' traffic model, which is a value between 0 and 1. However, by assuming a linear relationship between the two, Eq. (6) below can be derived. This assumption holds as both *k* and *JF* are minimum when there is no traffic congestion and at maximum when the traffic is at a standstill.

$$JF = \frac{k}{kj}*10 \tag{6}$$

The jam density, k_j would normally have to be determined through field observation, however, as the only means of obtaining traffic data is through the API, it is not possible to obtain such data. Hence, the assumptions described above are made in order to determine the jam density based on the existing information provided where:

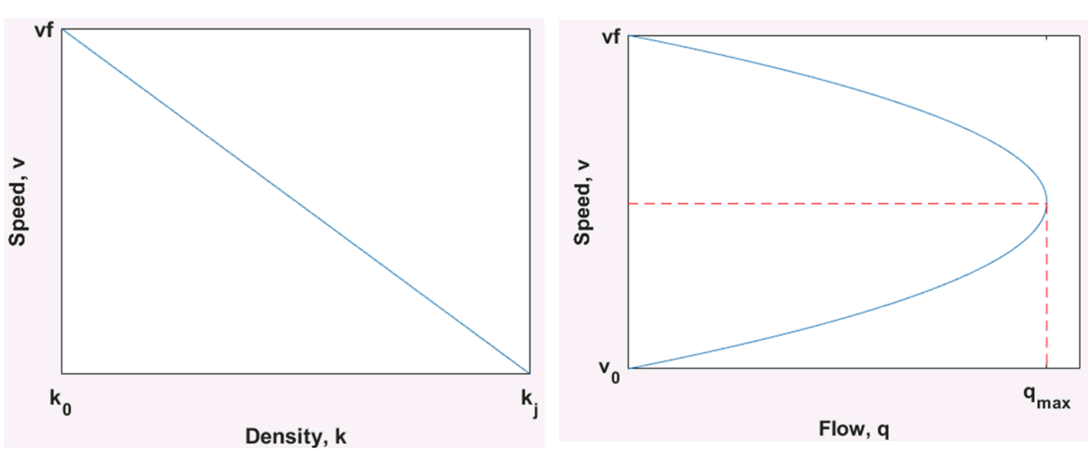

Fig. 1. Greenshields Model graphs indicating the relationship between traffic parameter. Left: Linear Relationship between speed and density. Right: Parabolic relationship between speed and flow.

$$kj = \frac{n_{Lane}}{H + L_{Vehicle}} \tag{7}$$

Based on the equation above, n_{Lane} is the number of lanes of each road, H is the distance between the front of a vehicle and the rear of the vehicle in front of it (Headway), and $L_{Vehicle}$ is the length of the vehicle.

The jam density is described by Eq. (7) by determining the maximum number of vehicles on the lane for a given speed — when a standstill occurs.

Utilising the derived equations, the traffic density, k can be found by rearranging Eq. (6). This would in turn give the value of the traffic flow, q in Greenshields' traffic flow equation, $q = k^*u$ where u is the velocity at the given time which is obtainable form the traffic API.

4.2. Congestion prediction and rerouting system

This subsection details the overall algorithm for the neural network traffic congestion prediction and rerouting system.

4.2.1. Pheromone-based multi-factor neural network (MVR) set-up

The MVR utilises a deep neural network to predict the traffic congestion of the roads for the next time step given the current state of the road. It is trained incrementally using the data obtained at each time step, otherwise known as the online training method. This method allows for a network to be deployed as-is in areas with little to no historical data as the network will be trained on-the-go.

In order to assist the neural network in adapting correctly to the given data, the sliding window technique is applied. The sliding window of size W is used to determine the horizon and size of the data batch used for the training. This ensures that the neural network captures only the most recent traffic activity and in turn adapts to the latest traffic trend.

The inputs to the neural network are descriptors of the roads of interests and are coined pheromone. The design of the deep neural network utilises 3 such pheromones, namely speed pheromone, density pheromone, and forecasted density pheromone.

4.2.2. Speed pheromone

The speed pheromone τ_{speed} describes the current mean speed of the road as a fraction of the free-flow speed as shown in Eq. (8) below:

$$\tau_{speed}(\mathrm{n}) = \frac{\overline{\mathrm{v(n)}}}{v_f(n)} \tag{8}$$

where $\tau_{speed}(\mathrm{n})$ represents the speed pheromone for a road n and $\overline{\mathrm{v(n)}}$ is the mean speed of the vehicles on the road n (Following the assumption mentioned in 4.1.1, all vehicles would be moving at this speed in the simulation, it is equivalent to the speed returned by the traffic API), and $v_f(n)$ is the free-flow speed or speed limit of the road n.

4.2.3. Density pheromone

The density pheromone $\tau_{density}$ describes the occupancy of the road, where a value of 1 means the road is fully occupied and a value of 0 means that there are no vehicles present. It is defined as shown in Eq. (9) below:

$$\tau_{density}(\mathrm{n}) = \frac{n_{vehicle}(n)^*(L_{vehicle} + H_0)}{\mathrm{L}_{Road}(n) + n_{Lane}(n)} \tag{9}$$

where $n_{vehicle}(n)$ represents the number of vehicle on the road n and $\mathrm{L}_{Road}(n)$ is the length of road n. Furthermore, in Eq. (9) above, H_0 is used instead of H as it represents the minimum space occupied by each vehicle.

4.2.4. Forecasted density pheromone

The forecasted density pheromone, $\tau_{forecast}$ was introduced by Chon et al. (2019) as the pheromone deposited by each vehicle on to the road which it intends to pass through. These pheromones carried by the vehicles are the density flow pheromone uniformly distributed for each road. As an example, for the road n with the density flow pheromone τ_{flow} shown in Eq. (10) below:

$$\tau_{flow}(\mathrm{n}) = \frac{\overline{\mathrm{v(n)}}^*\tau_{density}(\mathrm{n})^*n_{Lane}(n)}{\mathrm{L}_{Road}(n) + n_{Lane}(n)} \tag{10}$$

The pheromone carried by the vehicles along the road n, $\tau_{carried}(\mathrm{n})$ would then be $\frac{\tau_{flow}(\mathrm{n})}{\mathrm{n}_{vehicles}}$.

Each road will then have their future pheromones added or subtracted accordingly depending on if a vehicle is entering or leaving the road in the next time step. Vehicles remaining on the same road would not be required to deposit their carried pheromones. These pheromones would then be equally spread out onto the full area of the road. This is described in Eq. (11):

$$\tau_{forecast}(\mathrm{n}) = \frac{\tau_{carried\_arrived}(\mathrm{n}) - \tau_{carried\_depart}(\mathrm{n})}{\mathrm{L}_{Road}(n)^*n_{Lane}(n)} \tag{11}$$

where $\tau_{carried\_arrived}(\mathrm{n})$ are the pheromones of the vehicles entering the road and $\tau_{carried\_depart}(\mathrm{n})$ being the pheromones of the vehicles departing from the road.

4.2.5. Roads' costs

The deep neural network is trained to predict the future speed pheromone which will act as an indicator of the future congestion levels of the roads and reroute vehicles that are about to enter roads with a high congestion level as described by a congestion threshold, C. For roads which crosses this threshold, their predecessor roads or roads leading into them are looked into to find which vehicles plan to enter the road and attempt to reroute the vehicles.

The rerouting algorithm calculates the costs of each road by taking into account the weighted sum of predicted travel time and the current τ_{speed} of the road. Various weights have been tested and the best weight is found to be 0.5 for both factors. Hence, the cost of a road is calculated as shown in Eq. (12):

$$G_{cost}(n) = 0.5\left(\frac{L_{Road}(n)}{\tau_{predict}(n)^*v_f(n)}\right) + 0.5(\tau_{speed}(\mathrm{n})) \tag{12}$$

where $\tau_{predict}(n)$ is the predicted speed pheromone for a road n. The road network is then updated with the new costs and Dijkstra's algorithm (Walker & Skiena, 1992) was used to find the shortest path. Each time a vehicle is rerouted, the costs of the roads are updated to ensure that not all vehicles are directed to the same road, which would just cause another congestion.

4.2.6. Overall algorithm

The pseudo-code of the overall algorithm used is described below. The deep neural network used has 3 hidden layers with 20 neurons each. The window size used is 5 and prior to the 6th time step, all the traffic data is used for training as the simulation has less than 5 historical data to be used for training.

The simulation is run using MATLAB. Communication with SUMO is done via the TRACI (I. o. T. S. DLR, n.d.) interface using theTraci4Matlab package provided by Acosta (n.d.).

Algorithm 1: Multifactor Vehicle Routing (MVR) Algorithm

Inputs: Time step, t
Road data
Recorded roads' speed pheromone, $\tau_{speed;}$
Recorded roads' density pheromone, $\tau_{density;}$
Recorded roads' forecasted density pheromone, $\tau_{forecast}$
Window Size, W;
Vehicle Selection Parameter, L;
Congestion Threshold, C;
Outputs: Updated Road Network, G;
1. Take recorded roads' pheromone of size W up to t − 1 as training data

(continued on next page)

(*continued*)

Algorithm 1: Multifactor Vehicle Routing (MVR) Algorithm

2. Take speed pheromone time from $t_0 + 1$ to $t - 1$ where t_0 is the starting time step of the target data for the training of the neural network.
3. Train neural network using the three recorded roads' pheromone τ_{speed}, $\tau_{density}$, and $\tau_{forecast}$ of size W.
4. Use current pheromone data as input into the trained neural network
5. Obtain predicted speed pheromone, $\tau_{predict}$ from neural network
6. Update G's roads' costs using predicted travel time and speed pheromone with each weights being 0.5
7. Find predicted congested roads R_{Con} based on C.
8. **While** $R_{Con} > 0$
9.   Find road R in R_{Con} with the lowest $\tau_{predict}$;
10.   Get L predecessors of road R, R_{Pred}.
11.   **For** R' in R_{Pred}
12.   Find vehicles V along the R' which intends to pass through R;
13.   **For** V' $\in$ V
14.   Get current estimated location of V', P_{old};
15.   Compute shortest route using Dijkstra's Algorithm based on G's costs.
16.   Set new route for V'
17.   Estimate the new location, P_{new}
18.   **if** P_{new} != P_{old}
19.   Update road pheromones using line 4–6
20.   Update G
21.   **end**
22.   **end**
23.   **end**
24.   Remove R from R_{Con}
25. **end**

4.3. *Missing data handling method*

The following subsections introduces the proposed missing data handling method, as well as the method with which the missing data is detected.

4.3.1. *Missing data detection*

The simulation used in this project is working under the scenario of a Vehicle-to-Infrastructure (V2I) and Infrastructure-to-Vehicle (I2V) communication – the vehicles communicate wirelessly to the road infrastructure and vice-versa. (Namazi, Li, & Lu, 2019)

Missing data during the course of the simulation refers to the failures of the vehicles to transmit their traffic information to the infrastructures, which results in a 'No Data'. In order to detect the issue, a system must be put into place.

The system could communicate which each individual vehicle through a uniquely assigned ID, which in turn would provide the infrastructure with an idea of how many vehicles are on the road at the current time. This process is done at a set interval.

Throughout the course of the vehicle's travel, if there happens to be a failed transmission, the ID of the vehicle would still be available while the traffic data is not. This would then prompt the infrastructure to determine that there is an 'unobservable' or otherwise 'missing' data.

Through this method, it is possible to approximately quantify the number of missing data for a given time period and attempts to 'fill in' or to otherwise use such information to improve the existing system would be made possible.

A flowchart describing the process is shown in Fig. 2.

The traffic data obtained from the simulation follows the logic described above, which would then be used to determine the missing data fraction as well as the other traffic parameters such as the speed pheromone and density pheromone of the road (which information might be incorrect due to missing data).

Based on the desired amount of missing data desired for a particular simulation, Matlab would prompt TRACI for the vehicles ID and random exclude that many vehicles. For example, if 20% of missing data is desired, then out of 100 vehicles on the map, 20 vehicles' worth of data would not be included. This would then affect the speed, density and forecasted density pheromone later used for the training of the neural network.

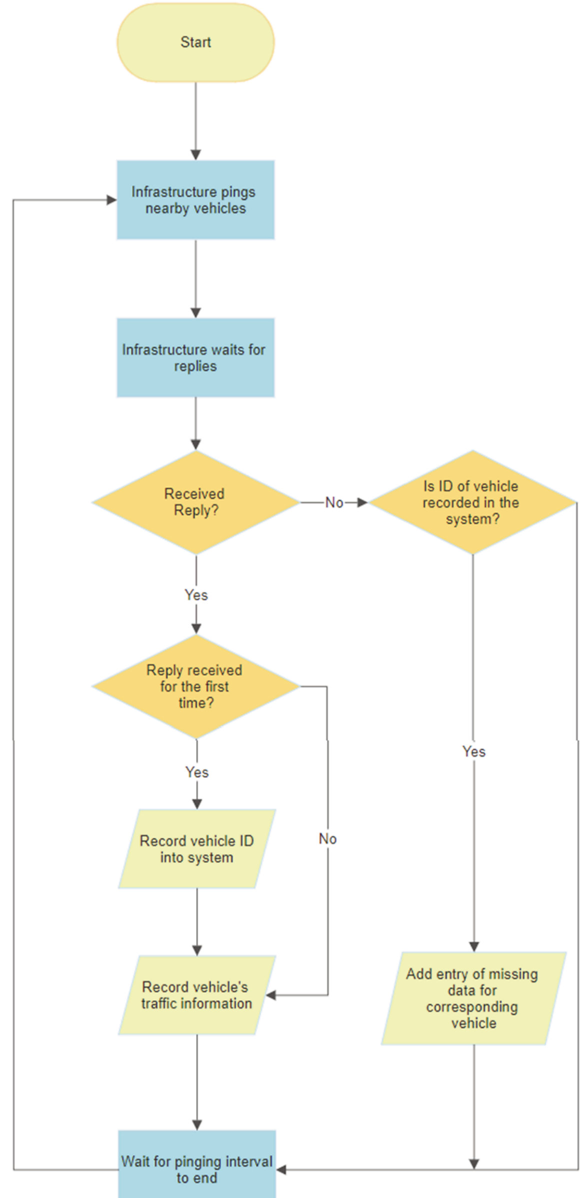

Fig. 2. Flowchart describing the communication process between the infrastructure and vehicles.

4.3.2. *Proposed missing data handling method – Weighted Missing Data Imputation (WEMDI)*

The proposed missing data handling method – WEMDI – is a data pre-processing method that assumes that the number of vehicles that fails to transmit information are known through methods such as the one mentioned in Section 4.3.1. In the case of SUMO, the ID used would be the vehicle ID used to generate the vehicle.

This method is utilised so as to complement the existing vehicle rerouting system which uses neural networks. As the inputs and targets to the neural network are the short term historical data – data belonging to a small time window before the current time step – the proposed method includes a long-term historical data – data belonging to other days of the same time window used by the short-term historical data.

Weights are imposed onto both the values in accordance to the amount of missing data detected. This is illustrated in Eq. (13) shown below:

$$input = (1 - MDF)*STH + MDF*LTH \tag{13}$$

where *input* is the data to be used as input to the neural network, *MDF* or Missing Data Fraction is the fraction of missing data detected with values in the range [0, 1], this will act as the weight between *STH* (short-term historical data) and *LTH* (long-term historical data)

STH is a matrix of size n_{Road} by W, where n_{Road} is the number of roads in the simulation and W is the window size. *STH* contains the past traffic pheromones from $t-1$to $t-1-W$.

Meanwhile, *LTH* is also a matrix of size n_{Road} by W but is an average of all the traffic records for that time collected prior to the current simulation. For example, for Wednesday's traffic, the historical data could be from Monday and Tuesday or the Wednesday of a few weeks prior to the current Wednesday.

The same data pre-processing method applies to the neural network's target output for training as well in order for the data to be consistent with one another. Through this method, the existing rerouting system would be improved in its robustness towards missing data.

The historical data used for the sake of this research is obtained by initialising the traffic simulation with traffic of different days but of the same time and using the SUMO-generated routes (The RandomTrips.py function) to determine the traffic activity throughout the simulation time. No routing was done during this phase and only the traffic parameters such as the related traffic pheromones for the neural network were collected)

4.3.3. Calibrating traffic data in rerouting system

By using the proposed missing data handling method – WEMDI – and integrating it MVR system a rerouting system which is robust against missing data can be produced.

WEMDI takes the inputs to the neural network and processes the data together with the collected historical data before training the neural network. This ensures that the inputs to the neural network are more accurate as they incorporate historical patterns to them in the events of missing data.

The flowchart shown in Fig. 3 describes the general process for the handling of the inputs to the neural network which would be affected by the missing data.

4.4. Simulation

The following subsections describes the setup of the traffic simulation to verify the performance of the proposed rerouting system and the missing data handling method.

4.4.1. Utilisation of traffic API to model live traffic situation

Through the equations from Section 4.1.1, the congestion levels for individual roads can be obtained and the necessary vehicles can be added to the map in SUMO prior to the start of the simulation – This is the setup time.

Using traffic data from HERE Traffic API, the road density of each road is calculated and an appropriate number of vehicles are initialised and added to these roads during this time. The routes given to these vehicles are obtained by matching the vehicles' starting point to existing routes generated using the in-built RandomTrips function of SUMO.

Only the initial traffic conditions are modelled due to the disjoint between reality and the simulation. Following that, the cars would begin to take a simulation-assigned route. The flow chart in Fig. 4 below illustrates the general process flow for modelling the traffic condition in the simulation.

4.4.2. Simulation of cases

In order to determine the effectiveness of the proposed method, a few simulations are done between the base rerouting system and the rerouting system which incorporates the proposed WEMDI method with varying levels of missing data — 20%, 30%, 40%, and 50%. — and are both compared to a case without any rerouting system implemented.

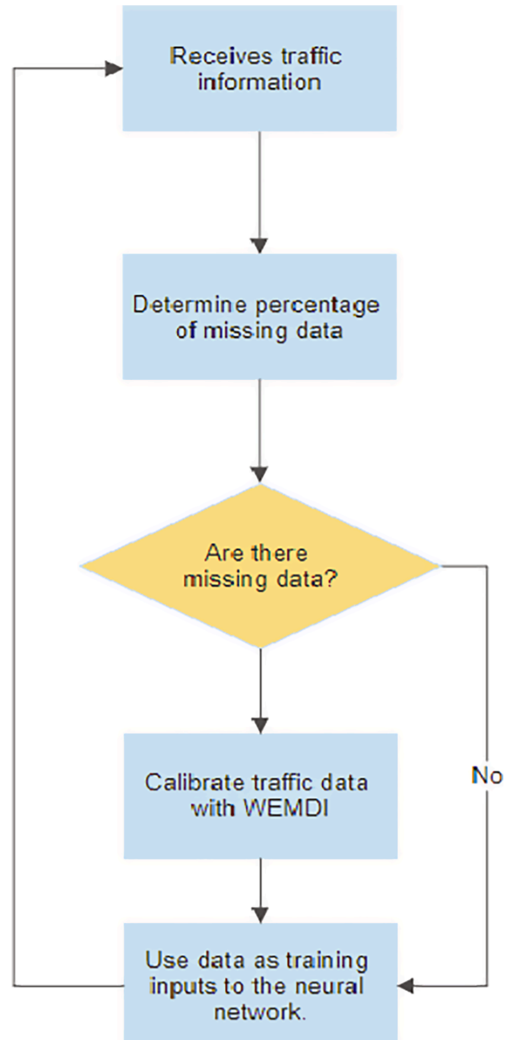

Fig. 3. Flowchart of how the missing data is handled.

4.4.3. Summarised process flow

Fig. 5 below describes the overall process flow of the system, summarising the previous subsections. The flow is split into 3 parts represented by 3 rows, each representing Collection and Implementation of Data, Running of Simulation, and Usage of WEMDI to Predict Traffic Congestion.

5. Results and analysis

The following subsections analyses the results obtain by testing the reliability of the traffic simulation as well as the performance of WEMDI.

5.1. Modelling live-traffic situation using data from traffic APIs

The following subsections display the results of the reliability of the derivations made in Section 4.1.1 through the use of HERE Traffic API.

5.1.1. Accuracy of derivations

By using the jam density, *kj* derived from Eq. (6) in Section 4.1.1 and utilising Greenshields' speed-density relationship equation: $v = vf - vf^*(k/kj)$ — where *vf* is the free flow speed on the road — the speed of the road can be found.

To verify the validity of the relationship between the jam density and

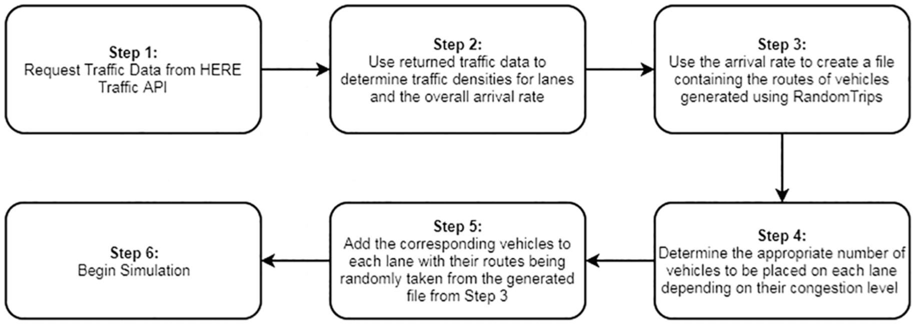

Fig. 4. Flow Chart for the Live Traffic Modelling.

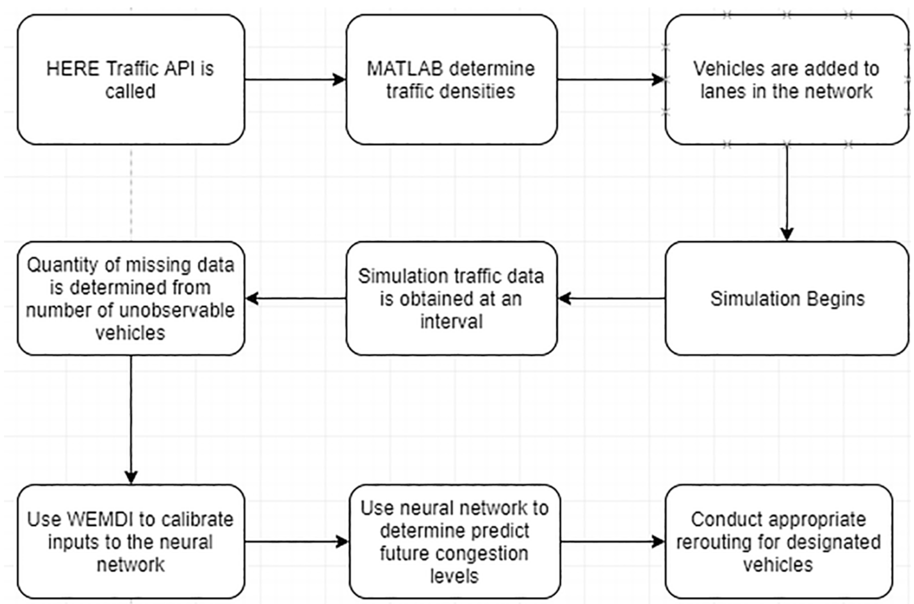

Fig. 5. System's overall process flow.

jam factor returned by the API, the speed is calculated using the API data's free-flow speed *vf*, and jam factor *JF* and compared to the API data's speed for different data points.

The calculated and API-returned mean speed for each data point is obtained and then compared in order to determine the validity of the derivation.

Table 2 displays the tabulated results for the comparison over for some of the data points.

From the table, it can be seen that the values are very close to one another, showing a maximum error of 3.704%. This shows that the derivations in Section 4.1.1 work well and that Eq. (6) is a suitable method of approximating the traffic density.

The small errors are likely due to the average speed being higher than the free-flow traffic (JF less than 0), resulting in a higher actual speed than the estimated one as the estimated speed assumes a JF of zero as the minimum, meaning the vehicles are assumed to follow the free-flow traffic stated by the API.

Table 2
Example of errors between mean measured and mean calculated speed.

Measured Mean Speed (m/s)	Calculated Mean Speed (m/s)	Error (%)
9.708	10.007	3.077
9.663	10.014	3.636
9.554	9.905	3.675
9.552	9.906	3.704

5.1.2. *Realization of real-time simulation*

After verifying the accuracy of the derived equations, the next step would be to take the traffic congestion data returned by the API and model it in SUMO. The simulation will aim to recreate the reflected Jam Factor for each given road at the initial time of the simulation.

However, due to the nature of traffic simulations, the routes of the vehicles has to be decided beforehand. The same goes to vehicles generated based on the arrival rate calculated for the map. This causes the real-time simulation to only reflect the traffic situation at the moment the traffic simulation is started. After the initial start of the simulation, everything else follows the predetermined routes and vehicles will enter from other parts of the map.

The objective is to create a simulation where the initial starting time resembles that of the current traffic.

A comparison between Google Maps and the resultant simulation is shown in Fig. 6. Some matches are circled and shown as well in the figure.

It can be seen from the figure that Google maps is more specific as to the congestion of parts of a road, whereas for the simulation in SUMO, the traffic is spread out evenly across the entire length of the road instead. Although there is such a discrepancy, it can be seen that the vehicles are being added at the proper locations based on the traffic data obtained from HERE Traffic API.

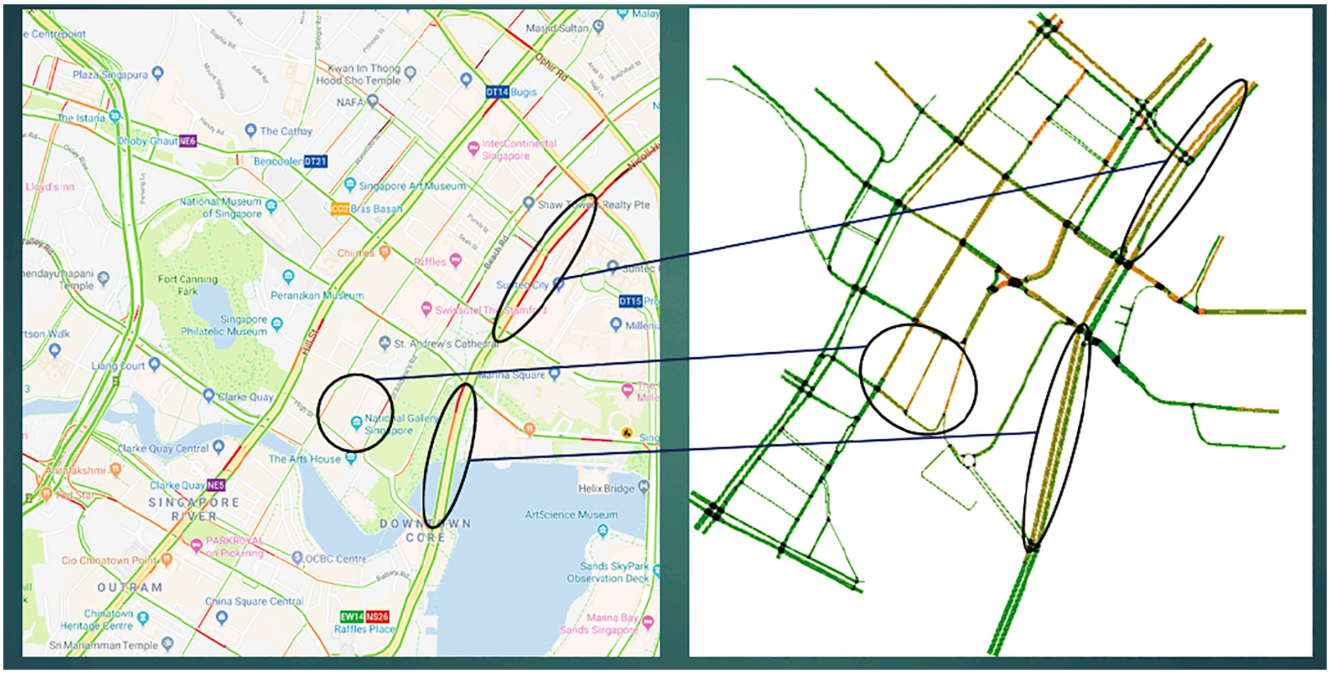

Fig. 6. Comparison between Google Map and the initialised Simulation Map at time t = 0.

5.2. *Performance of missing data handling method — WEMDI*

Using the proposed missing data handling method described in Section 4.3.2, the inputs and target outputs to the neural network has been calibrated to an appropriate value and fed into the neural network for training.

To test the effectiveness of the proposed missing data handling method – WEMDI – the rerouting system – Multi-Factor Vehicle Rerouting (MVR) – is integrated with it and is compared to its base form.

The simulation is run for missing data levels of 20%, 30%, 40%, and 50% with the required vehicle data pertaining to the factors in question collected throughout the simulation and tabulated as shown in Tables 3–5. From the table, it can be seen that while both MVR and WEMDI displayed improvements when compared to a situation where no control is placed, there are larger improvements shown by WEMDI over the MVR system as the missing data increases. This is due to the fact that as missing data increases, the standard rerouting system's error will increase as well, which results in a larger improvement by WEMDI as it is more robust towards such an event.

Moreover, it can be seen that the performance of WEMDI is consistent throughout the different cases of missing data as compared to MVR system where the performance fluctuates greatly between cases.

As shown in Figs. 7–9 above, it can be seen that the proposed system which utilises the proposed WEMDI method displayed significantly better results in all the tested situations with the MVR system only showing slightly competitive results in the case of 20% missing data which is a fairly low amount of missing data albeit still outperformed by WEMDI.

Table 3
Table showing the improvements of the referenced case (MVR) and WEMDI compared to the base case – Mean travelling Time.

Missing Data (%)	Mean Travelling Time (s)			Improvements from Base – No Routing(%)	
	Base – No Routing	MVR	WEMDI	MVR	**WEMDI**
20	520.84	325.57	291.55	37.49	**44.02**
30	520.84	380.92	312.79	26.86	**39.95**
40	520.84	342.03	301.44	34.33	**42.12**
50	520.84	413.72	312.70	20.57	**39.96**

Table 4
Table showing the improvements of the referenced case (MVR) and WEMDI compared to the base case – Average CO2 Emissions.

Missing Data (%)	Average CO_2 Emissions (kg)			Improvements from Base – No Routing(%)	
	Base – No Routing	MVR	WEMDI	MVR	**WEMDI**
20	2544.6478	1660.7915	1487.0838	34.73	**41.56**
30	2544.6478	1899.4872	1558.8931	25.35	**38.74**
40	2544.6478	1723.8110	1509.0487	32.26	**40.70**
50	2544.6478	2036.7505	1557.6797	19.96	**38.79**

Table 5
Table showing the improvements of the referenced case (MVR) and WEMDI compared to the base case – Average Fuel Emissions.

Missing Data (%)	Average Fuel Emissions (litres)			Improvements from Base – No Routing(%)	
	Base – No Routing	MVR	WEMDI	MVR	**WEMDI**
20	1093.8820	713.9235	639.2510	34.73	**41.56**
30	1093.8820	816.5365	670.1219	25.35	**38.74**
40	1093.8820	741.0147	648.6948	32.36	**40.70**
50	1093.8820	875.5439	669.6009	19.96	**38.79**

5.3. *Robustness of WEMDI towards different Maps*

In order to ensure that the proposed missing data handling method — WEMDI — is robust for different locations, the system is tested once again for 2 different locations in Malaysia — Bukit Bintang and Sunway.

Bukit Bintang is a busy part of Kuala Lumpur which is the capital of Malaysia, which would serve as a good benchmark for the performance of the system. Meanwhile, the Sunway map chosen is a considerably

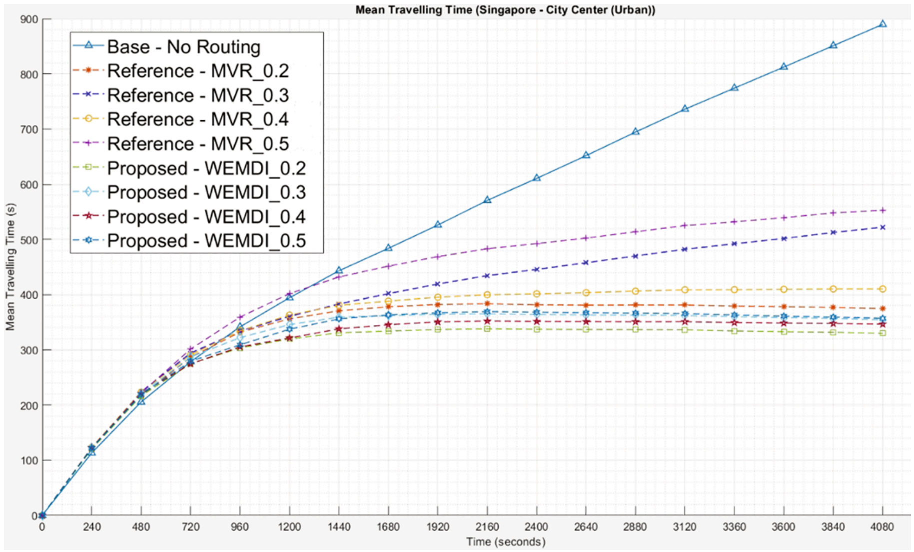

Fig. 7. Performance of Simulated Cases (Mena Travelling Time) for missing data of 0.2(20%), 0.3(30%), 0.4(40%), and 0.5(50%).

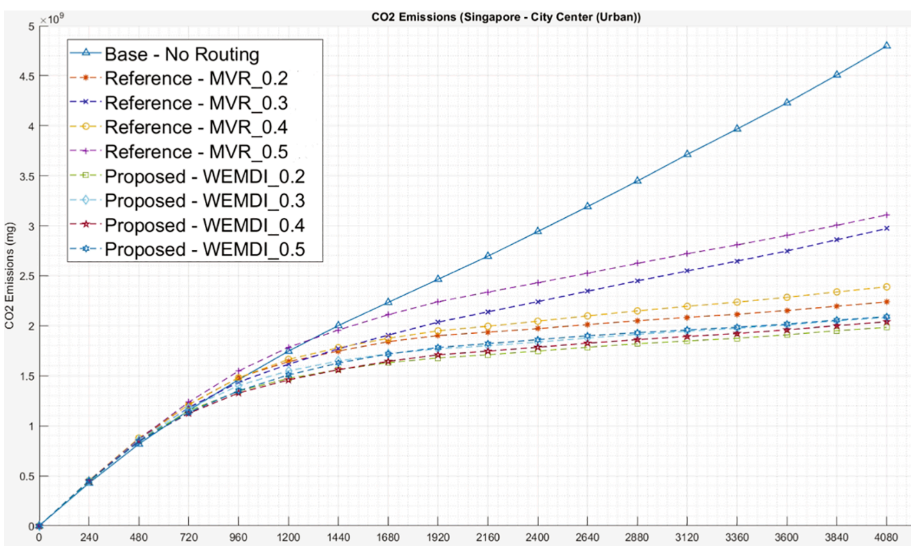

Fig. 8. Performance of Simulated Cases (CO2 Emissions) for missing data of 0.2(20%), 0.3(30%), 0.4(40%), and 0.5(50%).

larger map compared to the Bukit Bintang and Singapore city center maps, while also having less main roads and more residential roads, meaning that there is a fewer selection of roads for cars to get to their destination. This will test the system on its ability to spread the vehicles evenly throughout the available roads. Both maps are simulated with 50% missing data.

Bukit Bintang map is shown in Fig. 10 while Sunway is shown in Fig. 11 below:

Similar to the previous section, Tables 6–8 represents the results of the simulation done for the two maps, comparing the 3 cases of no routing, MVR, and WEMDI.

It can be seen that WEMDI still outperforms MVR by 7.13% for travel time and 8.47% for CO and fuel emissions for Bukit Bintang and for Sunway, outperforms MVR by 2.21% improvement in travel time and 0.34% in CO2 and fuel emissions.

5.4. Discussion

The real-traffic situation model of the simulation shows promising results when compared to Google Maps, proving that it is indeed possible to provide a more realistic traffic simulation using a simple Traffic API. While SUMO does have the ACTIVITYGEN tool as shown by Soon, Lim, and Parthiban (2019) to simulate traffic behaviour in a less linear manner, the tool itself uses high-level information such as population count, rate of unemployment, incoming and outgoing traffic and so on. These information tends to not be accurate as well as do not reflect the current traffic well enough. Hence, the proposed method would provide a more realistic traffic simulation for use, whether it be for traffic studies or driving simulators, which would in turn help advance the ITS field.

The online multi-factor deep neural-network has shown that it works even with little to no traffic data history, using only traffic data obtained from the simulation and learning on-the-go in small batches. This is encouraging as it solves the scalability problem plaguing many other existing models such as the model proposed by Soon, Lim, and Parthiban (2019) which trains the network using the whole data sequence which would then burden the system as the amount of data increases. The key is the sliding window technique introduced in this paper that allows the neural network to learn and adapt to new trends in the traffic condition rather than be confined to static-historical data as well as provide a

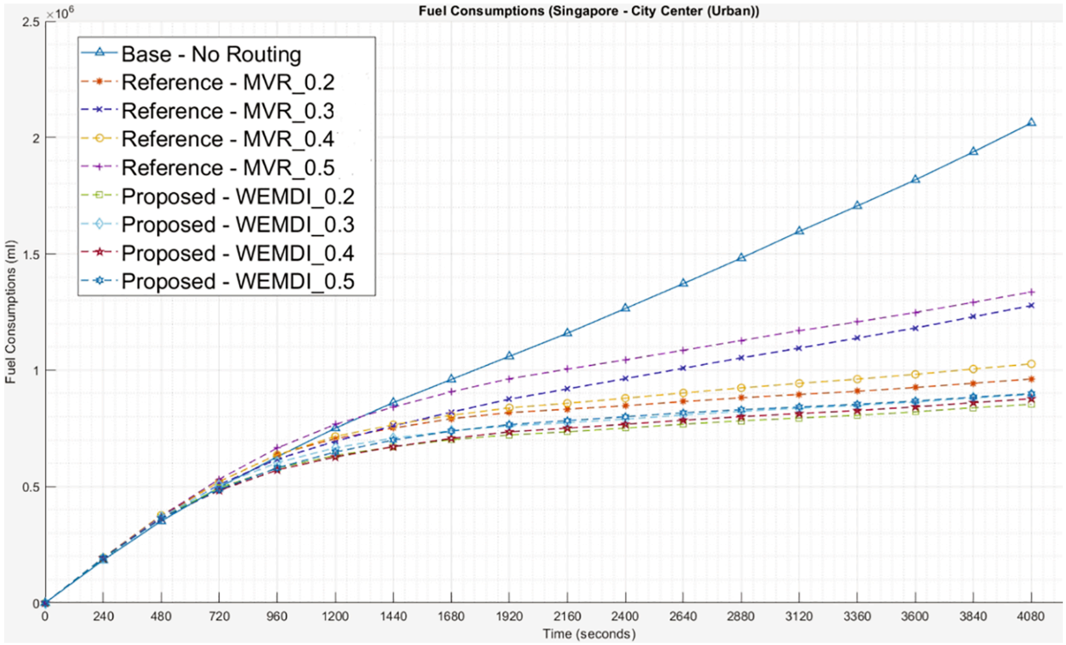

Fig. 9. Performance of Simulated Cases (Fuel Consumption) for missing data of 0.2(20%), 0.3(30%), 0.4(40%), and 0.5(50%).

Fig. 10. SUMO map of Bukit Bintang, Malaysia.

reasonable amount of data to train the network.

It can be seen that the implementation of the missing data imputation method WEMDI helps improve the performance of the traffic prediction system. This is especially true for Singapore city centre as well as Bukit Bintang, Malaysia. The improvements are smaller for the Sunway map, which could be due to a larger number of cars due to the size of the map coupled with a smaller selection of main roads compared to the other 2 maps but it can be seen that the improvements are present regardless. This proves that the data pre-processing method WEMDI shows that a historical average method is perfectly viable and competent in missing data imputation and has further potential in being improved upon.

6. Conclusion and future work

As the number of vehicles increases, so does the importance of an advanced intelligent transportation system (ITS). This brings into demand more realistic traffic simulation to test new traffic control methods to reduce emission of harmful substances from vehicles as well as increase the productivity level of the public by reducing travel time.

To aid the studies of such methods, a traffic simulation that models a live traffic situation is proposed in this work. The simulation recreates the current traffic based on data received by real-time traffic APIs. Although it is only able to replicate the current traffic condition and the generated vehicles would follow a simulation-provided route, the credibility of the traffic simulation would still increase due to the increase in realism of the simulation.

The proposed online multi-factor deep learning approach shows that

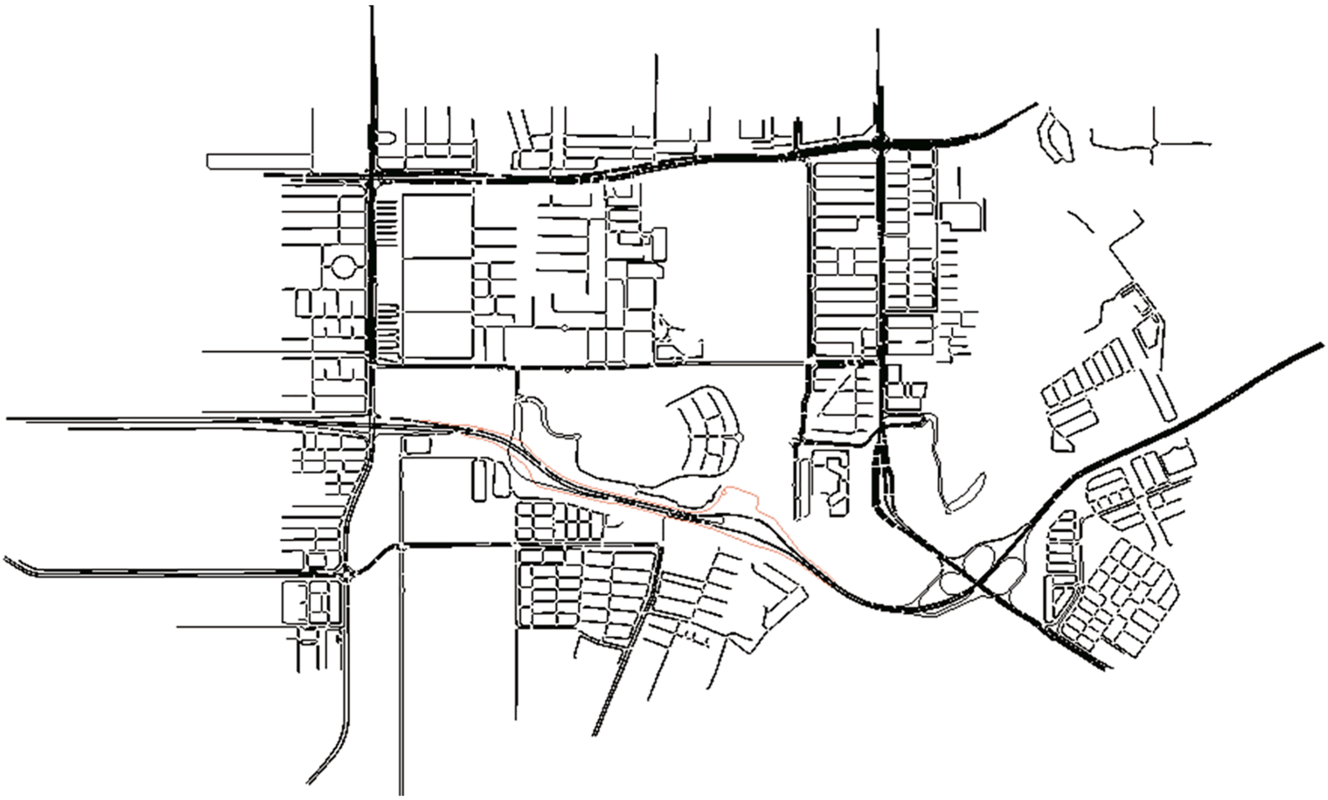

Fig. 11. SUMO map of Sunway, Malaysia.

Table 6
Table showing the improvements of the referenced case (MVR) and WEMDI compared to the base case – Mean travelling Time for Bukit Bintang Map.

Map	Mean Travelling Time (s)			Improvements from Base – No Routing(%)	
	Base – No Routing	MVR	WEMDI	MVR	**WEMDI**
Bukit Bintang	1064.19	339.75	263.92	68.07	**75.20**
Sunway	1380.98	831.86	801.32	39.76	**41.97**

Table 7
Table showing the improvements of the referenced case (MVR) and WEMDI compared to the base case – Average CO2 Emissions For Bukit Bintang Map.

Map	Average CO_2 Emissions (kg)			Improvements from Base – No Routing(%)	
	Base – No Routing	MVR	WEMDI	MVR	**WEMDI**
Bukit Bintang	4674.3722	1825.3538	1429.4603	60.95	**69.42**
Sunway	16938.1681	11733.2099	11675.6180	30.73	**31.07**

Table 8
Table showing the improvements of the referenced case (MVR) and WEMDI compared to the base case – Average Fuel Emissions For Bukit Bintang Map.

Map	Average Fuel Emissions (litres)			Improvements from Base – No Routing(%)	
	Base – No Routing	MVR	WEMDI	MVR	**WEMDI**
Bukit Bintang	2009.4305	784.6652	614.4823	60.95	**69.42**
Sunway	7281.3651	5043.7747	5019.0381	30.73	**31.07**

a newly initialised neural network can competently learn and predict traffic congestion using the spatiotemporal traffic information such as the speed pheromone, density pheromone, and forecasted density pheromone incrementally. The sliding window technique ensures that the neural network can learn effectively and respond to necessary changes in the traffic conditions. These improvements can be seen from the results above where the MVR method displayed over 37% reduction in mean travel time and more than 34% reduction in CO2 and fuel emissions.

In our proposed work, a historically weighted missing data handling method is proposed and integrated with a pheromone-based, multi-factor neural network vehicle rerouting system (MVR) called WEMDI. The implementation of WEMDI improved the existing MVR system by a further 6.5%-19.4% reduction in travel time and a further 6.8%-18.8% reduction in CO2 and fuel emission. The gap in performance continue to increase as the amount of missing data increases, thus confirming the robustness of the WEMDI system towards missing data.

In order to ensure the system is robust, it is also tested on another map — Bukit Bintang and Sunway, Malaysia. Despite using a different map, WEMDI still performs better than MVR. For Bukit Bintang, WEMDI showed a 7.13% reduction in travel time, and 8.47% reduction in CO2 and fuel emissions for a 50% missing data situation, while showing a 2.21% and 0.34% improvement in travel time and CO2 and fuel emissions for Sunway. The small improvements in the Sunway map could be attributed to the fewer main roads and larger number of residential roads, leading to fewer selection of roads for cars to take. However, the improvements in both these maps proves that the system is robust for different locations.

Future work could include non-recurring incidents such as events and accidents as these cases were not taken into account when designing the simulation and hence the performance of the neural network is not confirmed for such scenarios.

CRediT authorship contribution statement

Robin Kuok Cheong Chan: Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing - original draft, Visualization. **Joanne Mun-Yee Lim:** Conceptualization, Resources, Writing - review & editing, Visualization, Supervision, Project administration, Funding acquisition. **Rajendran Parthiban:** Writing - review & editing, Visualization, Supervision.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was funded by the Malaysian Ministry of Higher Education's Fundamental Research Grant Scheme (Grant Number: FRGS/1/2019/TK08/MUSM/03/1) under the purview of Monash University Malaysia.

References

Acosta, A. (n.d.). "TraCI4MATLAB." Retrieved from <https://www.mathworks.com/matlabcentral/fileexchange/44805-traci4matlab>.

Akhter, S., Ahsan, N., Jafor, S., Quaderi, S., & Forhad, A. Al. (2020). A SUMO Based Simulation Framework for Intelligent Traffic Management System. (June). https://doi.org/10.18178/jtle.8.1.1-5.

Ako, T., Yahaya, A. B., Ako, T., Atoo, A. A., & Yusuf, I. T. (2017). Determination of macroscopic traffic flow characteristics on an urban arterial using the moving car observer method assessment of engineering properties of asphaltic concrete produced in south western nigeria view project determination of macroscopic Traf. *USEP: Journal of Research Information Civil Engineering, 14*(2).

Bharadwaj, S., Ballare, S., Rohit, & Chandel, M. K. (2017). Impact of congestion on greenhouse gas emissions for road transport in Mumbai metropolitan region. *Transportation Research Procedia, 25*, 3538–3551. https://doi.org/10.1016/j.trpro.2017.05.282

Chen, L., Li, Q. R., Tian, X. Y., Chen, X. S., & Wang, R. X. (2012). Paratactic spatial-temporal two dimension data fusion based on support vector machines for traffic flow prediction of abnormal state. *Advanced Materials Research, 532–533*, 1225–1229. https://doi.org/10.4028/www.scientific.net/AMR.532-533.1225

Chon, M., Lim, J. M., Lun, K., & Yong, C. (2019). An improved pheromone-based vehicle rerouting system to reduce traffic congestion. *Applied Soft Computing Journal, 84*, 105702. https://doi.org/10.1016/j.asoc.2019.105702

de Grange, L., Marechal, M., & González, F. (2019). A traffic assignment model based on link densities. *Journal of Advanced Transportation, 2019*, 1–20. https://doi.org/10.1155/2019/5282879

Fan, J., & Cheng, J. (2018). Matrix completion by deep matrix factorization. *Neural Networks, 98*, 34–41. https://doi.org/10.1016/j.neunet.2017.10.007

Google. (n.d.). Google Maps Platform | Google Developers. Retrieved September 4, 2019, from <https://developers.google.com/maps/documentation/>.

Google. (2019). Map and Tile Coordinates. Retrieved September 4, 2019, from <https://developers.google.com/maps/documentation/javascript/coordinates>.

Greenshields, B. d., Bibbins, J. r., Channing, W. s., & Miller, H. h. (1935). A study of traffic capacity. Highway Research Board Proceedings, 1935. Retrieved from https://doi.org/.

HERE. (n.d.). What Is the Traffic API? - Traffic API - HERE Developer. Retrieved September 4, 2019, from <https://developer.here.com/documentation/traffic/topics/what-is.html>.

Hossain, M. T., & Hasan, M. K. (2019). Assessment of traffic congestion by traffic flow analysis in Pabna town. *American Journal of Traffic and Transportation Engineering, 4* (3), 75. https://doi.org/10.11648/J.AJTTE.20190403.11

Houchin, A. J., & Houchin, A. J. (2015). An investigation of freeway standstill distance , headway , and time gap data in heterogeneous traffic in Iowa by.

I. o. T. S. DLR. (n.d.). TraCI. Retrieved from <http://sumo.dlr.de/wiki/TraCI#TraCI_Commands>.

Idrissa, K. (2017). Mathematical study for traffic flow and traffic density in Kigali roads. *International Journal of Mathematical, Computational, Physical, Electrical and Computer Engineering, 11*(3), 104–108.

Krajzewicz, D., & Rossel, C. (2007). Simulation of urban mobility (SUMO). Centre for Applied Informatics (ZAIK) and the Institute of …, pp.1–35.

Lan, Q., Xu, X., Ma, H., & Li, G. (2020). Multivariable data imputation for the analysis of incomplete credit data. *Expert Systems with Applications, 141*, 112926. https://doi.org/10.1016/j.eswa.2019.112926

Lopez, P. A., Behrisch, M., Bieker-Walz, L., Erdmann, J., Flotterod, Y. P., Hilbrich, R., … Wiebner, E. (2018). Microscopic traffic simulation using SUMO. In *IEEE Conference on Intelligent Transportation Systems*. https://doi.org/10.1109/ITSC.2018.8569938

Map, O. S. (n.d.). Open Street Map. Retrieved September 4, 2019, from <https://www.openstreetmap.org/>.

Maroto, J., Delso, E., Félez, J., & Cabanellas, J. M. (2006). Real-time traffic simulation with a microscopic model. *IEEE Transactions on Intelligent Transportation Systems, 7* (4), 513–526. https://doi.org/10.1109/TITS.2006.883937

McGehee, D. V., Mazzae, E. N., & Baldwin, G. H. S. (2000). Driver reaction time in crash avoidance research: validation of a driving simulator study on a test track. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting, 44*(20), 3-320–3-323. https://doi.org/10.1177/154193120004402026

Mustapha, N. H. N., & Nik Hashim, N. N. W. (2016). Outflow of traffic from the national capital Kuala Lumpur to the north, south and east coast highways using flow, speed and density relationships. *Journal of Traffic and Transportation Engineering (English Edition), 3*(6), 540–548. https://doi.org/10.1016/j.jtte.2016.03.007

Namazi, E., Li, J., & Lu, C. (2019). Intelligent intersection management systems considering autonomous vehicles: A systematic literature review. *IEEE Access, 7*, 91946–91965. https://doi.org/10.1109/Access.628763910.1109/ACCESS.2019.2927412

Panichpapiboon, S., & Leakkaw, P. (2017). Traffic Density Estimation: A Mobile Sensing Approach. (December), (pp. 126–131).

Paramics microsimulation. (n.d.). Retrieved July 13, 2020, from <https://www.paramics.co.uk/en/about-us/article/paramics-microsimulation>.

Pell, A., Meingast, A., & Schauer, O. (2017). Trends in real-time traffic simulation. *Transportation Research Procedia, 25*, 1477–1484. https://doi.org/10.1016/j.trpro.2017.05.175

PTV Vissim. (2011). VISSIM 5.30-05 User Manual. Retrieved from <https://www.et.byu.edu/~msaito/CE662MS/Labs/VISSIM_530_e.pdf>.

Qu, L., Li, W., Li, W., Ma, D., & Wang, Y. (2019). Daily long-term traffic flow forecasting based on a deep neural network. *Expert Systems with Applications, 121*, 304–312. https://doi.org/10.1016/j.eswa.2018.12.031

Rakha, H., & Crowther, B. (2002). Comparison of Greenshields, Pipes, and Van Aerde car-following and traffic stream models. *Transportation Research Record, 1802*(1), 248–262. https://doi.org/10.3141/1802-28

Repast Simphony. (n.d.). Retrieved July 13, 2020, from <https://repast.github.io/>.

Sanik, M. E., Prasetijo, J., Nor, A. H. M., Hamid, N. B., Yusof, I., & Jaya, R. P. (2016). Analysis of car following headway along multilane highway. *Jurnal Teknologi, 78*(4), 59–64. https://doi.org/10.11113/jt.v78.7998

Sarathi Barma, S. B., Dutta, J., & Mukherjee, A. (2019). A 2-opt guided discrete antlion optimization algorithm for multi-depot vehicle routing problem. *Decision Making: Applications in Management and Engineering, 2*(2), 112–125. https://doi.org/10.31181/dmame1902089b

Shang, Q., Lin, C., Yang, Z., Bing, Q., & Zhou, X. (2016). Short-term traffic flow prediction model using particle swarm optimization-based combined kernel function-least squares support vector machine combined with chaos theory. *Advances in Mechanical Engineering, 8*(8), 1–12. https://doi.org/10.1177/1687814016664654

Smith, L., Beckman, R., & Baggerly, K. (n.d.). TRANSIMS: Transportation analysis and simulation system. https://doi.org/10.2172/88648.

Song, J., Wu, Y., Xu, Z., & Lin, X. (2015). Research on car-following model based on SUMO. Proceedings of 2014 IEEE 7th International Conference on Advanced Infocomm Technology, IEEE/ICAIT 2014, (January), 47–55. https://doi.org/10.1109/ICAIT.2014.7019528.

Soon, K. L., Lim, J. M. Y., & Parthiban, R. (2019). Extended pheromone-based short-term traffic forecasting models for vehicular systems. *Engineering Applications of Artificial Intelligence, 82*(February 2018), 60–75. https://doi.org/10.1016/j.engappai.2019.03.017

Soon, K. L., Lim, J. M. Y., Parthiban, R., & Ho, M. C. (2019). Proactive Eco-friendly Pheromone-based Green Vehicle Routing for multi-agent systems. *Expert Systems with Applications, 121*, 324–337. https://doi.org/10.1016/j.eswa.2018.12.026

Sun, F., Dubey, A., & White, J. (2018). DxNAT - Deep neural networks for explaining non-recurring traffic congestion. Proceedings - 2017 IEEE International Conference on Big Data, Big Data 2017, 2018-Janua, 2141–2150. https://doi.org/10.1109/BigData.2017.8258162.

Tan, H., Wu, Y., Shen, B., Jin, P. J., & Ran, B. (2016). Short-term traffic prediction based on dynamic tensor completion. *EEE Transactions on Intelligent Transportation, 17*(8), 2123–2133.

TomTom Developer. (n.d.). Homepage. Retrieved September 4, 2019, <from https://developer.tomtom.com/>.

TSIS-CORSIM. (n.d.). Retrieved July 12, 2020, from <https://mctrans.ce.ufl.edu/featured/tsis/>.

Valente, E., Avram, C., Machado, J., & Astilean, A. (2018). An innovative approach for modelling urban road traffic using timed automata and formal methods. *Journal of Advanced Transportation, 2018*, 1–15. https://doi.org/10.1155/2018/6269526

Walker, R., & Skiena, S. (1992). Implementing discrete mathematics: combinatorics and graph theory with mathematica. *The Mathematical Gazette, 76*(476), 286. https://doi.org/10.2307/3619148

Yokota, T., Erem, B., Guler, S., Warfield, S. K., & Hontani, H. (2018). Missing slice recovery for tensors using a low-rank model in embedded space, 26108003. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (pp. 8251–8259). https://doi.org/10.1109/CVPR.2018.00861.

Yokota, T., Zhao, Q., & Cichocki, A. (2016). Smooth PARAFAC decomposition for tensor completion. *IEEE Transactions on Signal Processing, 64*(20), 5423–5436. https://doi.org/10.1109/TSP.2016.2586759

Zhang, X., Zou, D., & Shen, X. (2018). A novel simple particle swarm optimization algorithm for global optimization. *Mathematics, 6*(12), 287. https://doi.org/10.3390/math6120287

Zhang, Y., & Zhang, Y. (2016). A Comparative study of three multivariate short-term freeway traffic flow forecasting methods with missing data. *Journal of Intelligent Transportation Systems: Technology, Planning, and Operations, 20*(3), 205–218. https://doi.org/10.1080/15472450.2016.1147813