



**HAL**  
open science

# A Deep-Learning Approach for Network Traffic Assignment with Incomplete Data

Yue Su, Wenbo Fan, Jakob Puchinger, Minyu Shen

► **To cite this version:**

Yue Su, Wenbo Fan, Jakob Puchinger, Minyu Shen. A Deep-Learning Approach for Network Traffic Assignment with Incomplete Data. Transportation Research Board 100th Annual Meeting, Jan 2021, Washington D.C., United States. <hal-03131516>

**HAL Id: hal-03131516**

**<https://hal.science/hal-03131516v1>**

Submitted on 4 Feb 2021

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

# TRB Annual Meeting

## A Deep-Learning Approach for Network Traffic Assignment with Incomplete Data

--Manuscript Draft--

<b>Full Title:</b>	A Deep-Learning Approach for Network Traffic Assignment with Incomplete Data
<b>Abstract:</b>	<p>The Origin-Destination (OD) data collection often relies on the questionnaire surveys which is inevitably incomplete. With incomplete input data, the traditional traffic assignment models (e.g., mathematical programming) cannot generate reasonable results. Alternatively, we propose a deep-learning approach employing Feed-Forward Neural Network (FFNN) for the traffic assignment that respects incomplete data. Experiments are conducted in the Braess's paradox network, Sioux Falls network, and Chicago sketch network. In the first two networks, training data for the FFNN is obtained by randomly generating 10000 OD scenarios and running mathematical assignment models for link flows. For Chicago sketch network, a mesoscopic tool is employed to generate the training data. The feasibility of using FFNN to learn traffic assignment mechanics is verified by using complete OD data and full link flow data with accuracy over 90% in three networks. In case of partially observed OD data, our idea is to learn the mapping between incomplete OD data and full link flow data. Experiments are conducted under different OD data incompleteness levels. The results demonstrate that the accuracy of FFNN model remains over 90% even losing 50% OD data and overwhelms that of the mathematical assignment model in three networks. Practically, the reported model can be trained for a certain network with easily-obtained partial OD data (e.g., observed cellular mobile data) and traffic flow data in the field (e.g., loop data and video data). Once well trained, when inputting voluminous incomplete OD data, the data-driven approach can provide accurate full link flows efficiently.</p>
<b>Manuscript Classifications:</b>	Data and Data Science; Artificial Intelligence and Advanced Computing Applications AED50; Deep Learning; Planning and Analysis; Transportation Network Modeling AEP40; Network Assignment; Transportation Planning Analysis and Application AEP15; Planning Data Analysis
<b>Manuscript Number:</b>	TRBAM-21-00354
<b>Article Type:</b>	Presentation and Publication
<b>Order of Authors:</b>	Yue Su Wenbo Fan, Ph.D Jakob Puchinger Minyu Shen

1 **A Deep-Learning Approach for Network Traffic Assignment with Incomplete Data**

2

3 **Yue Su**

4 Université Paris-Saclay, CentraleSupélec

5 Laboratoire Génie Industriel, Gif-sur-Yvette, France, 91190

6 School of Transportation and Logistics

7 Southwest Jiaotong University, Chengdu, Sichuan, China, 611756

8 Email: yue.su@centralesupelec.fr

9

10 **Wenbo Fan**

11 Ph.D., Associate Professor, Corresponding Author

12 School of Transportation and Logistics

13 Southwest Jiaotong University, Chengdu, Sichuan, China, 611756

14 Email: wbfan@swjtu.edu.cn

15

16 **Jakob Puchinger**

17 Ph.D., Professor

18 Université Paris-Saclay, CentraleSupélec, Laboratoire Génie Industriel

19 Gif-sur-Yvette, France, 91190

20 IRT SystemX, Palaiseau, France, 91120

21 Email: jakob.puchinger@centralesupelec.fr

22

23 **Minyu Shen**

24 Department of Electrical Engineering

25 The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong

26 Email: min-yu.shen@connect.polyu.hk

27

28 Words count: 5,236 words + 5 Tables (250 words per table) = 6,236 words

29

30

31 *Submitted for presentation at the Transportation Research Board (TRB) 100th Annual Meeting and publication in*  
32 *Transportation Research Record*

33

34 *Submission Date: Aug. 1, 2020*

1 **ABSTRACT**

2 The Origin-Destination (OD) data collection often relies on the questionnaire surveys which is inevitably  
3 incomplete. With incomplete input data, the traditional traffic assignment models (e.g., mathematical  
4 programming) cannot generate reasonable results. Alternatively, we propose a deep-learning approach  
5 employing Feed-Forward Neural Network (FFNN) for the traffic assignment that respects incomplete data.  
6 Experiments are conducted in the Braess's paradox network, Sioux Falls network, and Chicago sketch  
7 network. In the first two networks, training data for the FFNN is obtained by randomly generating 10000  
8 OD scenarios and running mathematical assignment models for link flows. For Chicago sketch network, a  
9 mesoscopic tool is employed to generate the training data. The feasibility of using FFNN to learn traffic  
10 assignment mechanics is verified by using complete OD data and full link flow data with accuracy over 90%  
11 in three networks. In case of partially observed OD data, our idea is to learn the mapping between  
12 incomplete OD data and full link flow data. Experiments are conducted under different OD data  
13 incompleteness levels. The results demonstrate that the accuracy of FFNN model remains over 90% even  
14 losing 50% OD data and overwhelms that of the mathematical assignment model in three networks.  
15 Practically, the reported model can be trained for a certain network with easily-obtained partial OD data  
16 (e.g., observed cellular mobile data) and traffic flow data in the field (e.g., loop data and video data). Once  
17 well trained, when inputting voluminous incomplete OD data, the data-driven approach can provide  
18 accurate full link flows efficiently.

19 **Keywords:** Traffic Assignment, Deep Learning, Feed Forward Neural Network, User Equilibrium,  
20 Incomplete OD Data  
21

## 1 INTRODUCTION

2 As the last step of the traditional four-step method, traffic assignment problem (TAP) aims to  
 3 distribute the origin-destination (OD) demand over the whole network and produce the link flows under  
 4 equilibrium. Accurate link flow and travel time predictions are important in traffic planning and  
 5 management. In TAP, the final link state is characterized by the well-known deterministic user equilibrium  
 6 (DUE) condition: all the utilized paths between the same OD pair experience the same travel time that is  
 7 no greater than any of the unused counterparts' (1). Assuming that the Jacobian matrix of link cost function  
 8 is symmetric, Beckmann (2) formulated the DUE state as the solution of a convex optimization, to which  
 9 the uniqueness of the link flow solution has been proved by (3). The most popular method to solve this  
 10 optimization problem is Frank-Wolfe (FW) algorithm (4), which utilizes linear approximation method and  
 11 converts TAP into a series of shortest path problems (5). Following this fundamental work, numerous  
 12 researches have been devoted in this field and developed a wide range of improved methods (6-9).

13 The OD demand matrix is the key input to traffic assignment. However, in real-world practice, the  
 14 collected OD demand data is often incomplete. The reasons are many. Major ones may include: the survey  
 15 methods per se (e.g., the sampled households' interview) and the data processing due to hardware and  
 16 software malfunction, carelessness, and so on (10). With incomplete OD matrix, the traditional traffic  
 17 assignment approaches cannot predict link flows accurately. The incomplete data issue motivates us to seek  
 18 another approach to predict link flows. On the one hand, the development of intelligent transportation  
 19 system (ITS) may enable us to obtain the full traffic flow data at each link (e.g., from loop detectors and  
 20 videos); complete OD data is difficult to obtain but partial OD data can be inferred easily from travelers'  
 21 cellular mobile records or vehicle's Global Position System (GPS) trajectories. With these rich data, we  
 22 propose to directly map the partially observed OD data with complete traffic flow data. Once the model is  
 23 well trained, we can yield accurate and complete traffic flow data within seconds even inputting voluminous  
 24 incomplete OD data. In practical life, we can use small sized simple (e.g., size of 10000) to train the model,  
 25 and input large-scale incomplete OD to get accurate and complete link flow data in a reasonable time.

26 Before we applied data-driven method to learn the mapping between incomplete OD data and  
 27 complete link flows, we should firstly make sure whether data-driven method is capable to learn the  
 28 mapping between complete OD data and complete link flow data. In other words, the feasibility of using  
 29 deep-learning method to learn complicate traffic assignment mechanism should be verified. The biggest  
 30 challenge of learning the traffic assignment mechanism is to learn the non-linearities. The Kolmogorov-  
 31 Arnold representation theorem votes for deep learning method, which states that any continuous function  
 32  $F(x)$  of  $n$  variables can be presented by the summation and function composition (11). The formulation is  
 33 expressed in **Equation 1**:

$$34 \quad 35 \quad 36 \quad F(x) = \sum_{j=1}^{2n+1} g_j \left[ \sum_{i=1}^n h_{ij}(x_i) \right] \quad (1)$$

37 where  $g_j$  and  $h_{ij}(x_i)$  are two continuous functions, of which  $g_j$  is a function of  $h_{ij}$  and  $h_{ij}$  stands  
 38 for bias and is independent from  $F$ . In other words, any continuous function can be reformulated with one  
 39 hidden layer and  $2n+1$  activation units in a neural network. Thus, using deep-learning method to learn  
 40 complex traffic assignment seems possible.

41 More recently, in the work of Polson and Sokolov (12), the deep-learning method has been  
 42 demonstrated to be effective in learning complex traffic patterns with sudden changing traffic conditions.  
 43 The most recent research (13) again verified the feasibility of using deep-learning approach in learning  
 44 complex traffic assignment mechanism of simple networks. Based on previous studies, we further explore  
 45 a deep-learning approach to determine traffic flows in the case of incomplete OD data. To this end, we  
 46 resort to feed-forward neural network (FFNN), which is a widely used technique in deep learning (14, 15).

47 In this research, we propose a deep-learning approach for the traffic assignment with incomplete  
 48 OD data. Specifically, a Feed-Forward Neural Network (FFNN) is employed for this task. We conduct  
 49 experiments in three different networks: Braess's paradox network, Sioux Falls network, and Chicago  
 50 Sketch network, representing small-sized, medium-sized and large-sized networks. In each network, we

1 investigate several scenarios with data missing rates ranging from 10% up to 50% of total input data. For  
2 each scenario, the mapping between incomplete OD data and full link flows is learnt. Once the deep-  
3 learning model is well trained, even inputting partially observed OD data, we can obtain accurate full link  
4 flows.

5 The paper is organized as follows. Next section reviews the relevant literature. Section 3 presents  
6 the model structure of FFNN and illustrates the input/output data preparation for training/testing model.  
7 The experiments of traffic assignment using incomplete OD matrix are conducted and analyzed in section  
8 4. Section 5 wraps up the main findings and indicates future extensions.

## 9 10 **LITERATURE REVIEW**

11 Generally, there are two variants of TAP: one is the static traffic assignment (STA) model, and the  
12 other is dynamic traffic assignment (DTA) model. STA models (16-18) are usually developed under fixed  
13 OD demand assumption set to mainly reflect the long-term demand patterns. Thus, it accounts for no  
14 dynamically varying demand. DTA models (19-24) have addressed this issue by treating OD demand as  
15 functions of time. The practical implementation of DTA models is limited because of its high computational  
16 cost and complexity. Both STA model and DTA model require complete description of the transportation  
17 system (25). Any incomplete data will render biased link flow estimation. Thus, employing traditional  
18 traffic assignment methods to predict link flows under incomplete data situation seems problematic.

19 Many efforts have been done in the field of traffic data imputation (26-30). The objective of traffic  
20 data imputation is to estimate the missing/corrupted traffic data. Three major approaches have been mainly  
21 implemented: prediction methods, interpolation, and statistical learning. Prediction methods (31-33) use  
22 historical data collected from the same site to develop prediction model to predict incomplete/corrupted  
23 data points. The idea of interpolation methods is to directly replace the missing/corrupted data points using  
24 historical data or neighboring data points. Thus, they largely depend on historical data, and are incapable  
25 to proceed data imputation without referring to historical data. Different from the first two approaches,  
26 statistical learning methods aim to build statistical model (e.g., expectation maximization and Markov  
27 Chain) to forecast incomplete/corrupted data based on an iterative regression technique (34, 35). The  
28 incomplete/corrupted data is then estimated until the model converges. In summary, the general idea of  
29 handling incomplete data is to fill it with an estimated value. It seems that no study has been dedicated to  
30 the mechanism of traffic assignment that directly admits the incomplete dataset. To fill the gap, we propose  
31 a deep-learning-based method and employ FFNN model to learn traffic assignment principles.

## 32 33 **METHODOLOGIES**

### 34 **Fully-connected FFNN model**

35 Here we propose a deep-learning neural network, which is designed in a multi-layer FFNN fashion. A FFNN  
36 neural network is derived from the classical artificial neural network (ANN). Three types of layers are  
37 contained in FFNN, including one input layer, one or more hidden layer/layers, and one output layer. The  
38 architecture of a simple FFNN is illustrated in **Figure 1**.

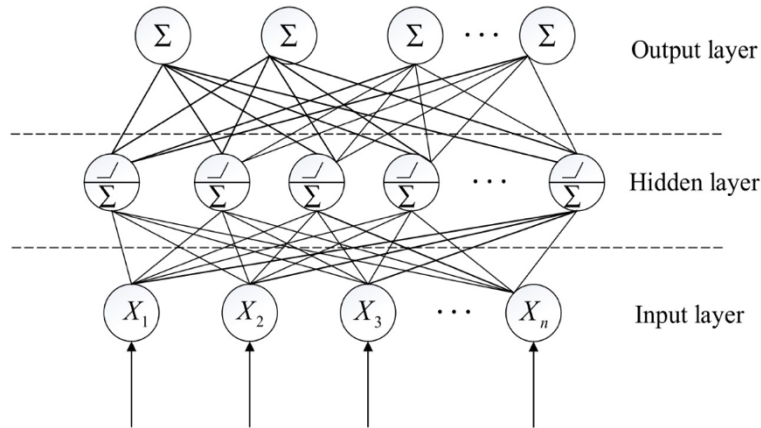


Figure 1 Fully-connected FFNN model architecture

In Figure 1, each neuron in input layer and hidden layer/layer is connected with all the neurons in next layer. Given the input dataset for training, denoted as  $\{X_1, X_2, X_3, X_4, \dots\}$ , the output of the input layer is characterized as in Equation 2:

$$Y = f(W_1 X + b) \quad (2)$$

where  $f(\cdot)$  is the layer activation function,  $W_1$  is the weight matrix,  $b$  stands for the bias vector.

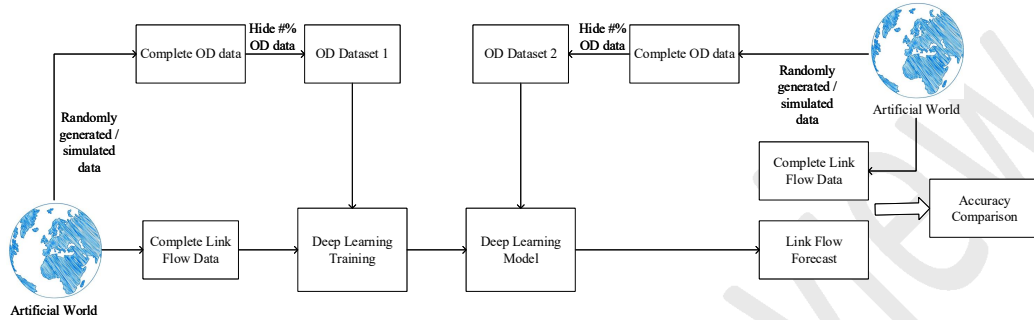
The next step,  $Y$  becomes the input for the hidden layer, and is encoded by activation function to calculate the output. Then, the output will be used as input for next layer. This process is repeated until obtaining output.

For training the input dataset, two passes are involved: forward pass and backward pass. The parameters,  $W$  and  $b$ , are randomly initialized with certain values, and the input data will propagate from network input to network output (36). The difference between desired output and model's actual output will be computed. Then, this calculated difference will be propagated backward through the network to adjust the values of weight matrix  $W_1$  and bias vectors  $b$ . This is so-called backward pass process. Those values will be adjusted iteratively until reaching their convergence (37). One iteration of the complete training dataset is called an epoch. To accomplish the model training process, many epochs are required. In each epoch, model employs backpropagation algorithm, which uses the gradient-descent minimization method to adjust the value of weight matrix and bias vectors. In this study, we have made comparison among different backpropagation algorithms in terms of model performance, and we chose ReLU (Rectified Linear Unit) as activation function.

### Modeling Framework for Learning Traffic Assignment Principle

To mimic the real world, we use both numerical experiments and simulations on three types of roadway networks, i.e., Braess's paradox network, Sioux Falls network, and Chicago network, representing the small, medium, and large-sized ones, respectively. For the first two, we randomly generate OD data and employ mathematical assignment models to calculate link flow data. As for Chicago sketch network, the training data is generated from a mesoscopic simulation tool, i.e., DTALite, which is capable of reflecting real-world traffic phenomenon, e.g., queuing at intersections and traffic lights (38). The benefits for using artificial data is twofold: Firstly, it allows us to conduct experiments regarding various levels of data incompleteness. Secondly, the randomly generated OD data and simulated data further illustrate the reliability of proposed model as we can set OD data with arbitrary value. The proposed FFNN model doesn't rely on the value of OD data nor the completeness of OD data. Once the OD matrices are entered, it can provide full link flows.

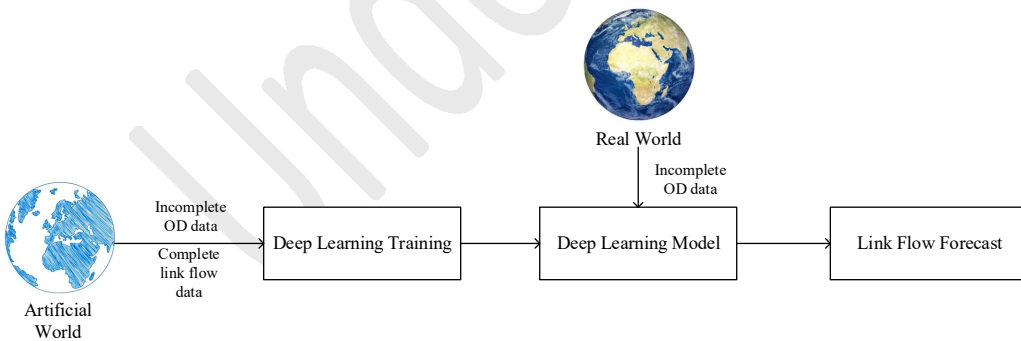
1 **Figure 2** presents the deep learning-based traffic assignment forecast evaluation process. To  
 2 evaluate the performance of proposed FFNN model of mapping the incomplete OD data with complete link  
 3 flow data, we conduct a series of experiments. The artificial world in the figure represents the world using  
 4 artificial OD data which is either randomly generated or simulated. The complete OD data is used to  
 5 calculate the corresponding link flow data. Then we hide certain percentage OD data to create incomplete  
 6 OD dataset (denoted as OD dataset 1). With the link flow data and OD dataset 1, we input them into deep  
 7 learning model for training. Then we input another incomplete OD data into the well-trained model. This  
 8 incomplete OD data (denoted OD data 2) is created from another complete OD data by hiding the same  
 9 percentage of data. Then the link flow forecast obtained from the well-trained deep-learning model and the  
 10 exact link flow data (i.e., calculated from complete OD data) are compared to assess the model accuracy.  
 11



12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22

**Figure 2 Deep learning-based traffic assignment forecast evaluation framework**

16 The proposed traffic assignment forecast is depicted in **Figure 3**, where in artificial world,  
 17 incomplete OD data and complete link flow data are inputted into deep learning training process. The deep  
 18 learning model are trained and validated by the process described in **Figure 2** in order to present the  
 19 relationship between partial OD data and complete link flow data. When the deep learning model is well  
 20 trained, we can use the real-world incomplete OD data to produce link flow forecast efficiently and  
 21 accurately.  
 22



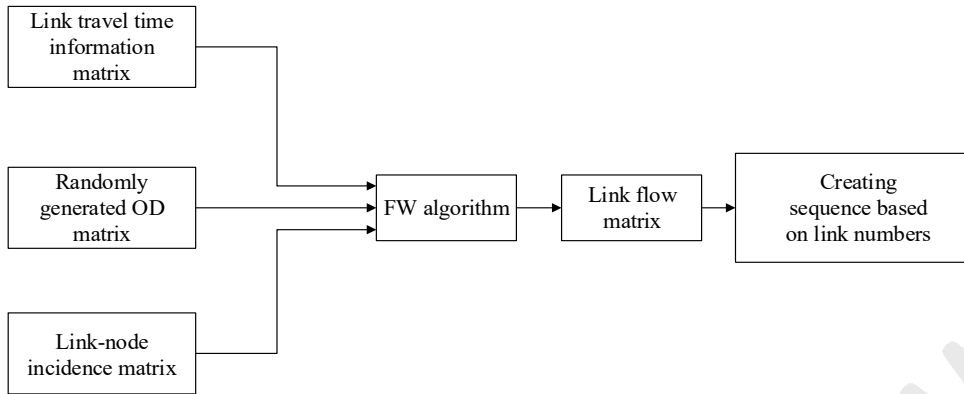
23  
24  
25  
26  
27  
28  
29  
30  
31  
32

**Figure 3 The proposal traffic assignment forecast procedure**

27 The data preparation is illustrated in **Figure 4(a)**. Given generated OD matrix and network  
 28 information, we employ FW algorithm to obtain link flows under UE principle. The calculated link flow  
 29 matrices are regarded as the target variables for training proposed model. **Figure 4(b)** exemplify the  
 30 network information in the Braess's paradox network. The link information matrix and link-node incidence  
 31 matrix are also plugged in the FFNN model as input features. There are three columns containing in the  
 32 link information matrix, they are: link numbers, capacity, and free flow travel time (FFT). The link travel

1 impedance is represented by the Bureau of Public Road (BPR) function. **Figure 4(c)** depicts the data flow  
 2 in training process of FFNN model.

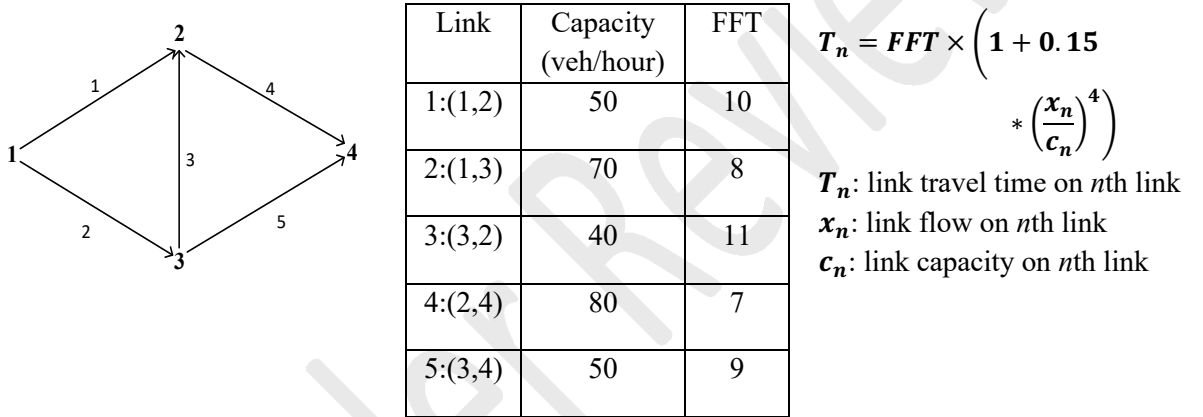
3



4

(a) Traffic assignment method for solving the network

5



Braess's paradox network structure

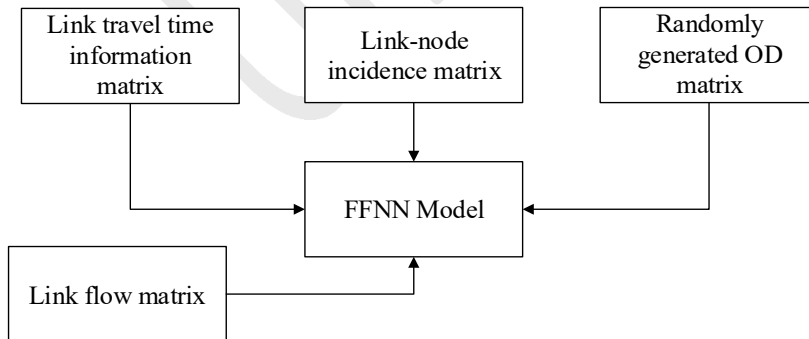
Network information matrix

BPR link travel time calculation

7

(b) Braess's paradox network

8



9

(c) Input and output data for training model

10

11

12

**Figure 4 Data preparation process for training FFNN model**

13

1 **EXPERIMENTAL RESULTS**

2 Experiments are conducted on a small-sized, a medium-sized, and a large-sized network, namely  
 3 the Braess’s paradox network, the Sioux Falls Network, the Chicago sketch network.

4  
 5 **Braess’s paradox network**

6 Firstly, we run the experiment on a Braess’s paradox network to verify whether proposed FFNN model can  
 7 learn the complex traffic assignment mechanics or not. In this case, the complete OD data and complete  
 8 link flow data is inputted to train and validate proposed model, the model accuracy is then calculated. To  
 9 begin with, we randomly generate the OD matrix, in which the OD demand values ranging from 100 to 150  
 10 units. Each time, FW algorithm is used to determine corresponding link flow matrix. By repeating this  
 11 process 10000 times, we yield 10000 OD matrices and 10000 link flow vectors. For each data point, we  
 12 simply flatten the OD matrix of shape (4,4) into a vector of size 16. Similarly, we flatten the link information  
 13 matrix and link-node incidence matrix into vector of size 15 and 20, respectively. These flattened matrices  
 14 are then concatenated into one vector of size 51. The output dimension is equal to the number of links. We  
 15 split the total dataset into 70% training data and 30% test data.

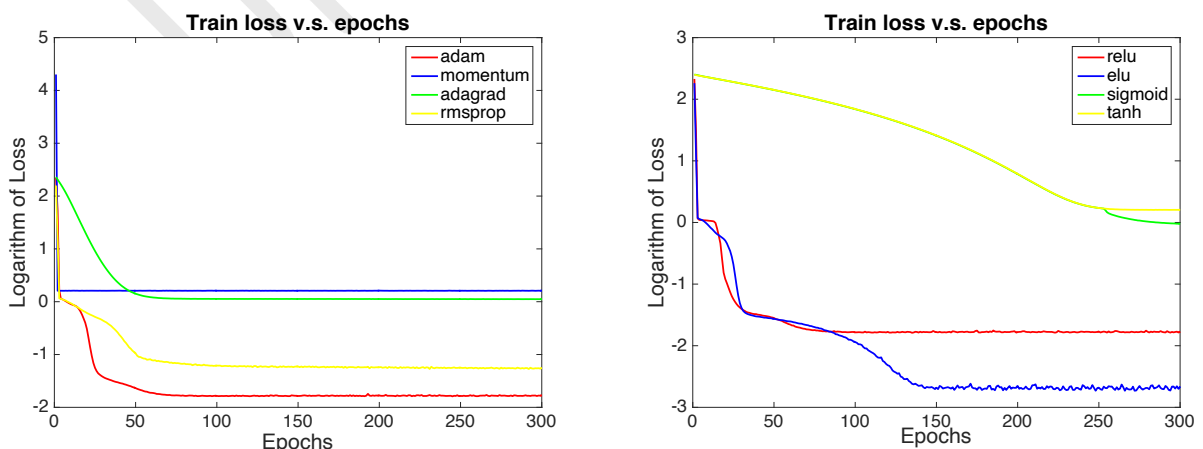
16  
 17 *Model Selection*

18 To get the best model performance, we firstly select the optimal combination of hyperparameters including  
 19 the learning rate, batch size, optimizers, and activation functions (AFs). To this end, we tune the  
 20 hyperparameters manually and train the model with different optimizers and AFs. We compare the losses  
 21 of four types of stochastic gradient descent-based optimizers (i.e., Adam, Momentum, Adagrad, RMSprop)  
 22 and four types of AFs (i.e., ReLU, Elu, Sigmoid, Tanh). To visualize the performance, **Figure 5** shows the  
 23 logarithm of training losses changes with iteration times. We further study the performance metrics when  
 24 deploying different optimizers and different AFs. Root Mean Squared Error (RMSE), Mean Average Error  
 25 (MAE), Mean Average Percentage Error (MAPE) are selected as performance metrics, and their  
 26 formulations are expressed as followings.

27  
 28 
$$RMSE = \sqrt{\frac{\sum_{t=1}^n (y_t - \hat{y}_t)^2}{n}} \tag{3}$$

29  
 30 
$$MAE = \frac{\sum_{t=1}^n |y_t - \hat{y}_t|}{n} \tag{4}$$

31  
 32 
$$MAPE = \frac{\sum_{t=1}^n |y_t - \hat{y}_t|}{y_t} \times 100\% \tag{5}$$



(a) Train loss for different optimizers (AF=ReLU) (b) Train loss for different AFs (optimizer=Adam)

**Figure 5 Train losses for different optimizers and different AFs**

While training the model, we set up the initial batch size equal to 64. This operation aims to divide the training and test dataset to small batches instead of passing the entire dataset. Then we fix the AF as ReLU (Rectified Linear Unit) and train the model with four kinds of optimizers separately. The experimental results show that, model that uses Adam optimizer has the highest model accuracy (99.35%). Then, we change our AF and find that Elu has the best system performance while comparing with others. As ReLU learns faster than Elu (39), and the system performances regarding these two AFs seem approximate. Thus, we finally select Adam (Adaptive moment estimation) as the optimizer and ReLU as AF. The optimal learning rate for proposed model is 0.001, and optimal batch size is 128. **Table 1** summarizes the FFNN model performance with different optimizers and AFs. Selecting the fine-tuned parameters, the model accuracy is 99.35% for Braess's paradox network. The accuracy of FFNN model to learn equilibrium traffic assignment mechanism has been verified.

**TABLE 1 Performance Metrics for Different Optimizers and AFs**

Optimizer/AF selected	Train loss	RMSE	MAE	MAPE%	Accuracy%
Performance metrics for different optimizers with AF = ReLU					
Adam	0.0170	0.1320	0.1404	0.65%	99.35%
Adagrad	1.1144	1.0608	0.3297	4.87%	95.13%
RMSprop	0.0557	0.2330	1.9577	1.23%	98.77%
Momentum	1.6224	1.2735	0.7508	6.26%	93.74%
Performance metrics for different AFs with optimizer = Adam					
ReLU	0.0170	0.1320	0.1404	0.65%	99.35%
Elu	0.0022	0.0470	0.1361	0.23%	99.77%
Sigmoid	0.9540	0.9881	0.8645	4.69%	95.31%
Tanh	1.6026	1.2852	0.5741	6.28%	93.72%

*Incomplete OD data as Input*

We next investigate whether a deep learning-based method can accurately accomplish traffic flow assignment task given incomplete OD data. To address this issue, we conduct two experiments. In the first experiment, we hide total demand of randomly chosen OD pairs. In the second, we randomly hide OD trips out of every trip origin according to certain percentage. The difference between two experiments is: the first reflects the fact that for certain OD pairs the demand information may be totally unknown with no samples in reality; while the second one implies that OD trips information is only partially available for each surveyed trip zones. We compare the accuracy of the results of FW algorithm and that of deep-learning algorithm. **Table 2** reports the model accuracy. It is not surprising that the accuracy of FW algorithm deteriorates fast with higher levels of data incompleteness. On the contrary, the accuracy of our FFNN model remains higher than 94% in both experiments.

**TABLE 2 Model Accuracy Comparisons with Incomplete OD Dataset: Braess's Paradox Network**

First experiment: randomly hiding OD pairs out of 6 in total			
Scenarios	Link flow results of FW algorithm	Link flow results of Deep learning	Improvement
No missing	100%	99.35%	-0.65%
Hide 1 OD pair (17%)	80.56%	97.08%	16.52%
Hide 2 OD pairs (33%)	63.68%	96.38%	32.70%

Hide 3 OD pairs (50%)	48.99%	95.27%	46.28%
Hide 4 OD pairs (67%)	33.69%	94.89%	61.20%

1

Second experiment: randomly hiding OD trips for each trip origin			
Scenarios	Link flow results of FW algorithm	Link flow results of Deep learning	Improvement
Hide 10%	90.23%	98.36%	8.13%
Hide 25%	71.40%	95.44%	25.04%
Hide 30%	64.19%	95.26%	31.07%
Hide 50%	44.90%	94.85%	49.95%

2

3

4

5

6

7

8

9

10

11

12

13

14

Results in **Table 2** demonstrate the robustness of the FFNN model. Furthermore, FFNN model also entails computation efficiency. The well-trained FFNN model will finish calculation in few hundredths of FW algorithm's running time, which is more suitable to be implemented practically.

#### Impacts of Missing Certain OD Pairs on Model Accuracy

Another interesting point is to examine the impact of missing individual OD pairs on model accuracy. The most influential OD pair may be identified. There are six OD pairs in this network: (1,2), (3,2), (1,3), (1,4), (2,4), and (3,4). We sequentially hide these OD pairs, and then train and test FFNN model. The results are summarized in **Table 3**.

**TABLE 3 Impacts of Missing OD Pairs on Model Accuracy**

Missing OD pairs	Train loss	RMSE	MAE	MAPE%	Accuracy%
No missing	0.0022	0.0470	0.1361	0.23%	99.77%
OD(1,2)	0.2792	0.5431	0.6965	2.79%	97.21%
OD(3,2)	0.1844	0.4445	0.4127	2.26%	97.74%
OD(1,3)	0.2640	0.5253	0.3103	1.70%	98.30%
OD(1,4)	0.3686	0.6230	0.9354	3.11%	96.89%
OD(2,4)	0.2807	0.5284	0.2204	1.53%	98.47%
OD(3,4)	0.2485	0.5082	0.2920	2.54%	97.46%

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

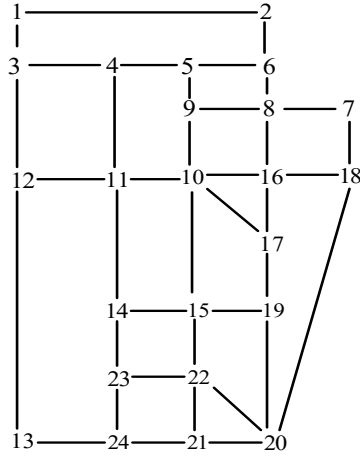
31

It is observed from **Table 3** that when OD pair (1,4) is missing, the model accuracy decreases by 2.88%, which is the biggest decrease among all the cases. The findings highlight that in real-world practice of data collection, particular OD pairs need special attention and more efforts/resources in survey to guarantee the data availability.

#### Sioux Falls Network

##### Model Selection

We further execute the FFNN model in a more complex network, as shown in **Figure 6** to verify the model effectiveness in the case of incomplete OD data. Similar to the case of Braess's network, we generate 10000 OD matrix randomly, flatten each OD matrix into a vector of size 576, and therefore obtain an input matrix of shape 10000\*576. Then we select the optimal combination of hyperparameters. Here, for the sake of simplification, we report directly the selected hyperparameters. We chose Adam as optimizer and ReLU as activation function. To avoid overfitting, we add dropout rate of 0.01 and 0.005 in the first and second hidden layer. We split the dataset into 80/20, in which 80% for training, 20% for testing. The learning rate in this scenario is 0.001. The model accuracy is 97.31% after training.



**Figure 6 Sioux Falls Network (24 nodes and 76 links)**

*Incomplete OD data as Input*

Then we do the similar experiments on Sioux Falls network. In the first experiment, we hide total demand of randomly chosen OD pairs with the missing rates of 10%, 20%, 30%, 40%, and 50%, respectively. In the second experiment, same missing rates are applied to each trip origin. The comparison of model accuracy between the traditional traffic assignment method and FFNN model is summarized in **Table 4**.

**TABLE 4 Model Accuracy Comparisons with Incomplete OD Dataset: Sioux Falls Network**

First experiment: randomly hiding OD pairs			
Scenarios	Link flow results of FW algorithm	Link flow results of Deep learning	Improvement
No missing	100%	97.31%	-2.69%
Hide 10% OD pairs	88.06%	94.40%	6.34%
Hide 20% OD pairs	77.04%	93.20%	16.16%
Hide 30% OD pairs	66.40%	92.71%	26.31%
Hide 40% OD pairs	55.70%	92.08%	36.38%
Hide 50% OD pairs	45.27%	91.52%	46.25%

Second experiment: randomly hiding OD trips			
Scenarios	Link flow results of FW algorithm*	Link flow results of Deep learning	Improvement
10%	90.89%	94.83%	3.94%
20%	78.90%	93.45%	14.55%
30%	85.43%	93.99%	8.46%
40%	69.58%	92.74%	23.16%
50%	85.88%	94.05%	8.17%

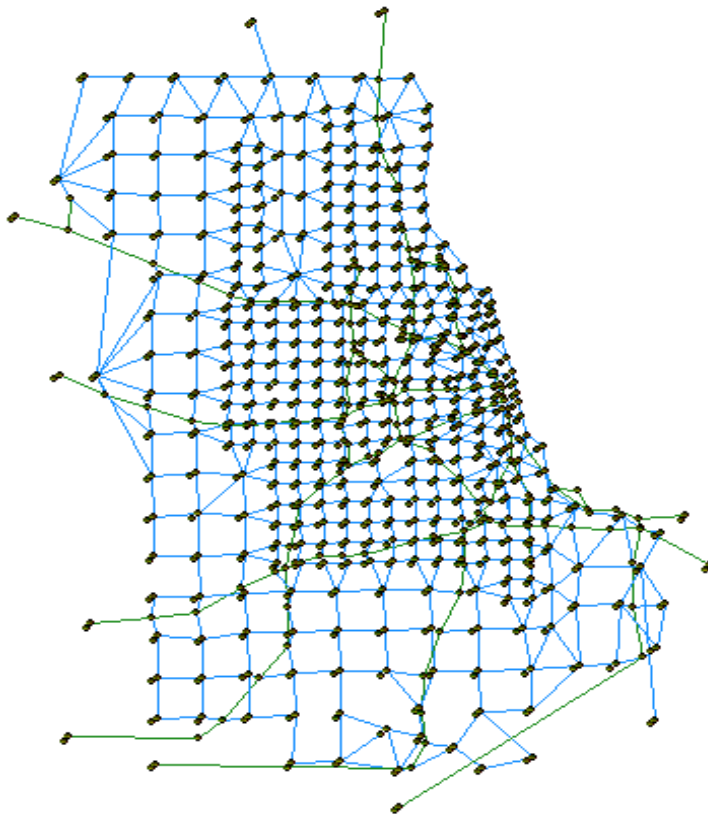
\* The accuracy of FW assignment fluctuates in this experiment due to the randomly missing OD trips, which may hurt different when they occur in different OD pairs (as confirmed by our results in Braess's network).

The results in **Table 4** are also impressive when we perform the FFNN model in Sioux Falls network. It is seen that FFNN model has an excellent performance even with half of all OD pairs missing. On the contrary, the traditional traffic assignment method is not qualified to predict link flows for any levels of data incompleteness. Significant improvements on the model accuracy are observed for the FFNN model ranging from 6.34% to 46.25%, as opposed to the results of FW algorithm.

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19

**Chicago Sketch Network**

Finally, we conduct our experiments in Chicago sketch network, as shown in **Figure 7**. Different from Braess’s paradox network and Sioux Falls network, the training data is now generated from a mesoscopic simulation tool, i.e., DTALite, which is capable of reflecting real-world traffic phenomenon, e.g., queueing at intersections (38). Similarly, we hide OD pairs randomly chosen from OD matrix with the missing rates of 10% to 50%. **Table 5** summarizes the comparisons of results by FW algorithm and our model as against the simulated results. Interestingly, we find that our model already beats the FW algorithm with higher accuracy under the no-missing scenario. The reason is that FW algorithm calculate link flows in ideal condition (e.g., no traffic light and queueing), while the link flow obtained using simulation tool considered the traffic phenomenon. At the same time, proposed FFNN model learns the mapping between OD data and simulated link flow data directly and takes these factors into account. Therefore, the predicted link flow obtained by FFNN model can better reflect practical situation than that by FW algorithm. Similar outcomes are also observed in the 2<sup>nd</sup> experiment. It should be mentioned that the accuracy of using FW algorithm to predict link flow fluctuates comparing with the first experiment. The reason is that in the second experiment, we randomly hide certain percentage of OD trips out of every trip origin rather than hiding a certain percentage of OD pairs directly out of total demand. As the consequence, the actual number of missing OD pairs fluctuates.



20  
21  
22  
23  
24  
25  
26  
27

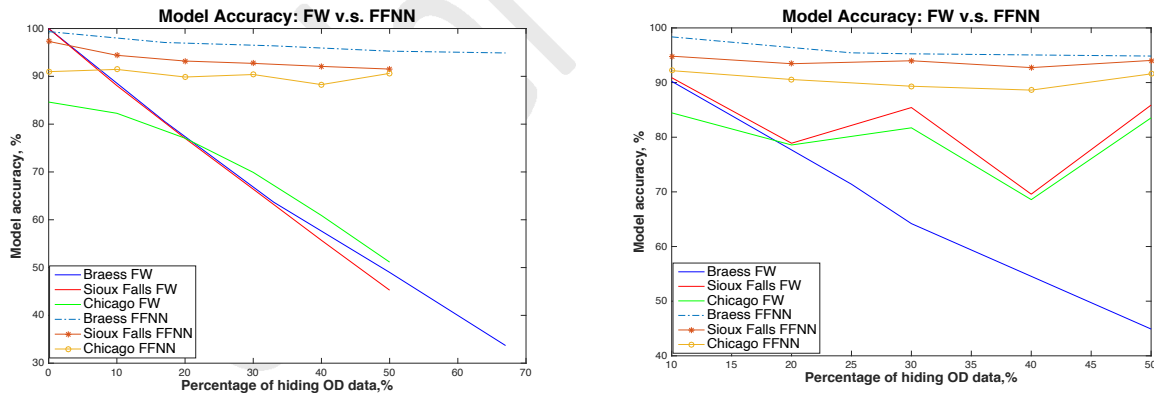
**Figure 7 Chicago sketch network (387 demand zones, 933 nodes, and 2950 links)**

**TABLE 5 Model Accuracy Comparisons with Incomplete OD Dataset: Chicago Sketch Network**

First experiment: randomly hiding OD pairs			
Randomly hide # OD pairs	Link flow results of FW algorithm	Link flow results of Deep learning	Improvement
No missing	84.62%	90.98%	6.27%
Hide 10% OD pairs	82.27%	91.45%	9.18%
Hide 20% OD pairs	77.06%	89.86%	12.80%
Hide 30% OD pairs	69.91%	90.38%	20.77%
Hide 40% OD pairs	60.90%	88.27%	27.37%
Hide 50% OD pairs	51.14%	90.62%	39.48%
Second experiment: randomly hiding OD trips			
Scenarios	Link flow results of FW algorithm*	Link flow results of Deep learning	Improvement
10%	84.46%	92.18%	7.72%
20%	78.58%	90.56%	11.98%
30%	81.72%	89.33%	7.61%
40%	68.58%	88.62%	20.04%
50%	83.54%	91.59%	8.05%

\* The accuracy of FW assignment fluctuates in this experiment due to the randomly missing OD trips, which may hurt different when they occur in different OD pairs (as confirmed by our results in Braess’s network).

In **Table 5**, we can find a significant improvement (i.e., 39.48%) in link flow prediction accuracy when hiding 50% OD pairs out of total demand in the first experiment. In the second experiment, the accuracy of deep-learning model remains stable while that of FW algorithm often experiences dramatic decrease (e.g., from 81.72% to 68.58%). The performance of using deep-learning model in large-sized network to accomplish accurate link flow prediction has been proved. To be more straightforward, **Figure 8** compares the model accuracy in three networks while conducting first and second experiment.



(a) Model accuracy in the first experiment

(b) Model accuracy in the second experiment

**Figure 8 Model accuracy: FW v.s. FFNN**

### CONCLUSIONS AND EXTENSIONS

In this research, we develop a novel data-driven approach for traffic assignment in the case of incomplete OD data without need of data imputation or sample expansion. Feed-forward neural network is embedded in a deep-learning structured model to learn the mechanism of UE traffic assignment principle. The model is trained by randomly generated OD matrix and its corresponding link flow patterns that determined by FW algorithm. Experiments are conducted in three typical networks, i.e., Braess’s paradox network, Sioux Falls network and Chicago sketch network. The FFNN model shows good performance

1 with model accuracy over 97% in small-sized and medium-sized network. For large-sized network, as  
2 Chicago sketch network in this research, the accuracy of deep learning model is still over 90%. Those  
3 results indicate that FFNN model can learn the equilibrium traffic assignment mechanism effectively and  
4 efficiently.

5 Then, we randomly hide OD pairs from each OD matrix and compare the assignment results of the  
6 traditional FW algorithm and that of FFNN model. The experimental results are promising. Under various  
7 levels of data incompleteness, the result accuracy of FFNN model remains over 87%, even with 50% OD  
8 pairs missing. In contrast, the accuracy of the traditional traffic assignment method deteriorates quickly and  
9 unsurprisingly becomes unqualified for any levels of data incompleteness. The results demonstrate that the  
10 effectiveness and robustness of the FFNN model in traffic assignment, particularly for incomplete OD data.  
11 This finding provides a new direction for traffic assignment with partial data, that is we can learn the  
12 mapping between partial observed OD data (easy to obtain from GPS but usually incomplete) and fully  
13 available link flow data (e.g., via loop detector or videos) and train the deep-learning model. Once the  
14 model is well trained, link flow results with satisfactory accuracy can be very efficiently obtained by  
15 inputting the newly-surveyed incomplete raw data.

16 Another application of FFNN model in traffic assignment is to identify the most “influential” OD  
17 pairs that have largest impacts on model accuracy while missing. Accordingly, targeted survey may be  
18 designed. We are also examining other deep-learning architectures such as convolution neural network and  
19 graph convolution network for comparison with the current one. For next extension, it is also interesting to  
20 study if the deep learning method is competent for dynamic traffic assignment task with time-varying OD  
21 demand as input, which definitely has no time for sophisticated data preprocessing.

## 22 **ACKNOWLEDGMENTS**

23 This study is funded by the National Nature Science Foundation of China (NSFC 51608455) and Sichuan  
24 Science & Technology Program (No. 2020YFH0038).

## 25 **AUTHOR CONTRIBUTION**

26 The authors confirm contribution to the paper as follows: study conception and design: Yue Su, Wenbo Fan,  
27 Minyu Shen; data collection: Yue Su; analysis and interpretation of results: Yue Su, Wenbo Fan, Jakob  
28 Puchinger, Minyu Shen; draft manuscript preparation: Yue Su, Wenbo Fan, Jakob Puchinger, Minyu Shen.  
29 All authors reviewed the results and approved the final version of the manuscript.  
30  
31

## REFERENCES

1. Wardrop, J.G. Some Theoretical Aspects of Road Traffic Research, in: Inst Civil Engineers Proc London, UK, 1952, pp. 72-73.
2. Beckmann, M. J., McGuire, C. B., Winsten, C. B., and Koopmans, T. C. Studies in the Economics of Transportation, in: Yale University Press, New Haven, Connecticut, 1956
3. Simth, M. The Existence, Uniqueness and Stability of Traffic Equilibria, *Transportation Research Part B*, 1979. 13(4): 295-304.
4. Frank, M., Wolfe, P. Algorithm for Quadratic Programming. *Naval Research Logistics*, 1956. 3: 95-110.
5. Nie, Y., Zhang, H. M., and Lee, D. Models and Algorithms for the Traffic Assignment Problem with Link Capacity Constraints. *Transportation Research Part B*, 2004. 38: 285-312.
6. Xu, S., Jiang, W., Deng, X., Shou, Y. A Modified Physarum-inspired Model for the User Equilibrium Traffic Assignment Problem. *Applied Mathematical Modelling*, 2018. 55: 340-353.
7. Martin, B.V., Manheim, M.L. A Research Program for Comparison of Traffic Assignment Techniques. *Highway Research Record*, 1965. 88: 69-84.
8. Ruiter, E.R. The Prediction of Network Equilibrium-The State of Art. Presented International Conference on Transportation Research, Bruges, Belgium, 1973.
9. Wilkie, D.F., Stefanek, R.G. Precise Determination of Equilibrium in Travel Forecasting Problems Using Numerical Optimization Techniques. *Highway Research Record*, 1971. 369: 239-252.
10. Bae, B., Kim, H., Lim, H., Liu, Y., and Han, L.D.. Missing Data Imputation for Traffic Flow Speed Using Spatial-temporal Cokriging. *Transportation Research Part C*, 2018. 88:124-139.
11. Kolmogorov, A. On the Representation of Continuous Functions of Several Variables as Superpositions of Smaller Number of Variables. *Soviet. Math. Dokl*, 1956. 108: 179-182.
12. Polson, N.G., Sokolov V.O. Deep Learning for Short-term Traffic Flow Prediction. *Transportation Research Part C*, 2017. 79: 1-17.
13. Rahman, R., Hasan, S. Learning Traffic Assignment of a Network: Some Experimental Findings from a Deep Learning Approach. *IEEE Intelligent Transportation System Conference*, 2019
14. Hinton, G.E. What Kind of Graphical Model is the Brain? Presented at 19<sup>th</sup> International Joint Conference on Artificial Intelligence, 2005.
15. Bengio, Y., Lamblin, P., Popovici, D., Larochelle, H. Greedy Layer-wise Training of Deep Networks. *Advances in Neural Information Processing Systems*, 2006. 19: 153-160.
16. Yang, H., Yagar, S. Traffic Assignment and Signal Control in Saturated Road Networks. *Transportation Research Part A*, 1995. 29(2): 125-139.

17. Nguyen, S., James, L. Traffic-An Equilibrium Assignment Program. 1976, In: Centre de recherche sur les transports, Université de Montréal.
18. LeBlanc, L. J., Morlok, E. K., and Pierskalla, W. P. An Efficient Approach to Solving the Road Network Equilibrium Traffic Assignment Problem. *Transportation Research*, 1975. 9(5): 309-318.
19. Yagar, S. Dynamic Traffic Assignment by Individual Path Minimization and Queuing. *Transportation Research*, 1971. 5: 179-196.
20. Merchant, D. K. and Nemhauser, G. L. A Model and an Algorithm for the Dynamic Traffic Assignment Problems. *Transportation Science*, 1978. 12(3): 183-199.
21. Merchant, D. K. and Nemhauser, G. L. Optimal Conditions for a Dynamic Traffic Assignment Model. *Transportation Science*, 1978. 12(3): 200-207.
22. Ho, J. K. A Successive Linear Optimization Approach to the Dynamic Traffic Assignment Problem. *Transportation Science*, 1980. 14: 295-305.
23. Nie. Y. A Cell-based Merchant-Nemhauser Model for System Optimum Dynamic Traffic Assignment Problem. *Transportation Research Part B*, 2011. 45(2): 329-342.
24. Long, J., Szeto, W.Y., Shi, Q., Gao, Z, Huang, H.J. A Non-linear Equation System Approach to the Dynamic Stochastic User Equilibrium Simultaneous Route and Departure Time Choice Problem. *Transportmetrica A*, 2015b. 11(5):388-419.
25. Patriksson, M. The Traffic Assignment Problem: Models and Methods, 1994, (VSP Publishers, Utrecht, Netherlands).
26. Li, Y., Li, Z., Li, L. Missing Traffic Data: Comparison of Imputation Method. *Intelligent Transport Systems*, 2014. 8: 51-57
27. Han, L., Chin, S.M., Hwang, H.L. A Spatial-temporal Approach for High Resolution Traffic Flow Imputation. Presented at Transportation Research Board 96<sup>th</sup> Annual Meeting. Washington, D.C., 2016
28. Strub, I. S, Bayen, A. M. Weak Formulation of Boundary Conditions for Scalar Conservation Laws: an Application to Highway Traffic Modelling. *International Journal of Robust and Nonlinear Control*, 2006. 16(16): 733-748.
29. Zou, H., Yue, Y., Li, Q., Yeh, A.G.O. An Improved Distance Metric for the Interpolation of Link-based Traffic Data Using Kriging: a Case Study of a Large-scale Urban Road Network. *Information Science*, 2012. 26: 667-689.
30. Shamo, B., Asa, E., Membah, J. Linear Spatial Interpolation and Analysis of Annual Average Daily Traffic Data. *Journal of Computing in Civil Engineering*, 2015. 29(1): [https://doi.org/10.1061/\(ASCE\)CP.1943-5487.0000281](https://doi.org/10.1061/(ASCE)CP.1943-5487.0000281)
31. Zhong, M., Sharma, S., Lingras, P. Genetically Designed Models for Accurate Imputation of Missing Traffic Counts. *Transportation Research Record: Journal of Transportation Research Board*, 2004. 1879: 71-79.

32. Li, L., Su, X., Zhang, Y., Lin, Y., Li, Z. Trend Modeling for Traffic Time Series Analysis: an Integrated Study. *IEEE Transactions on Intelligent Transportation Systems*, 2015. 16: 3430–3439.
33. Gan, M., Chen, C.P., Li, H.X., Chen, L. Gradient Radial Basis Function Based Varying-coefficient Autoregressive Model for Nonlinear and Nonstationary Time Series. *IEEE Signal Process Letter*, 2015. 22: 809–812.
34. Smith, B., Scherer, W., Conklin, J. Exploring Imputation Techniques For Missing Data in Transportation Management Systems. *Transportation Research Record: Journal of Transportation Research Board*, 2003. 1836: 132–142. <sup>[1]</sup><sub>[SEP]</sub>
35. Ni, D., Leonard, J. D. Markov Chain Monte Carlo Multiple Imputation Using Bayesian Networks for Incomplete Intelligent Transportation Systems Data. *Transportation Research Record: Journal of Transportation Research Board*, 2005. 1935: 57–67. <sup>[1]</sup><sub>[SEP]</sub>
36. Robi, P.S., Dixit, U.S. Application of Neural Networks in Generating Processing Map for Hot Working. *Journal of Materials Processing Technology*, 2003. 142(1): 289-294
37. Svozil, D., Kvasnicka, V., Pospichal, J. Introduction to Multi-layer Feed-forward Neural Network, *Chemometrics and Intelligent Laboratory Systems*, 1997. 39: 43-62.
38. Zhou, X., Taylor, J. DTALite: A queue-based mesoscopic traffic simulator for fast model evaluation and calibration, *Cogent Engineering*, 2014, 1(1): 1-19.
39. LeCun, Y., Bengio, Y., Hinton, G. Deep Learning. *Nature*, 2015. 521: 436-444.