



**HAL**  
open science

# Unravelling System Optimums by trajectory data analysis and machine learning

Ruiwei Chen, Ludovic Leclercq, Mostafa Ameli

► **To cite this version:**

Ruiwei Chen, Ludovic Leclercq, Mostafa Ameli. Unravelling System Optimums by trajectory data analysis and machine learning. *Transportation research. Part C, Emerging technologies*, 2021, 130, pp1-23. 10.1016/j.trc.2021.103318 . hal-03462087

**HAL Id: hal-03462087**

**<https://hal.science/hal-03462087v1>**

Submitted on 1 Dec 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.

# Unravelling System Optimums by trajectory data analysis and machine learning

Ruiwei Chen<sup>a</sup>, Ludovic Leclercq<sup>a,\*</sup>, Mostafa Ameli<sup>b</sup>

<sup>a</sup>Univ. Gustave Eiffel, Univ. Lyon, ENTPE, LICIT, F-69518, Lyon, France

<sup>b</sup>Univ. Gustave Eiffel, COSYS, GRETTIA, Paris, France

---

## Abstract

This work investigates network-related trajectory features to unravel trips that contribute most to system under-performance. When such trips are identified, feature analysis also permits determining the best alternatives in terms of routes to bring the system to its optimum. First, we define important trajectory features that unravel the trips contributing the most to the network under-performance. Second, based on supervised learning methods, we propose a two-step data-driven methodological framework to reroute a part of the users and make the system close to its optimum. The learning models are trained with trajectory features to identify which users should be selected, and which alternative routes should be assigned, thanks to the data and features obtained from two reference dynamic traffic assignment (DTA) simulations, under User-Equilibrium (UE) and System-Optimum (SO). We only focus on trajectory features that are **accessible in real time, such as network features and regular travel time information, so that the methods proposed can be implemented without requiring cumbersome network monitoring and prediction**. Finally, we evaluate the efficiency of the methods proposed through microscopic DTA simulations. The results show that by targeting 20 % of the users according to our selection model and moving them onto paths predicted as optimal alternative paths based on our rerouting model, the total travel time (TTT) of the system is reduced by 5.9 % in comparison to a UE DTA simulation. This represents 62.5 % of the potential TTT reduction from UE to SO, when all the users choose their path under the SO condition.

*Keywords:* Trajectory data analysis, Network-related trajectory features, System optimum (SO), Supervised learning, Polynomial regression, Naive Bayes classifier

---

## 1. Introduction

In urban areas, the conflict between the demand for increasing mobility and limited infrastructure capacities degrades the level of service of road networks. The consequences include: (i) economic loss resulting from wasted time and fuel in traffic jams, and (ii) environmental impacts due to air pollutants and noise emissions. Expanding infrastructures to improve network performance may not always be efficient in the long term (Braess et al., 2005). In this work, we focus on re-routing strategies that bring the system closer to the global optimum by moving some users out of their personal optimum. This solution occurs when all the users aim at minimizing their own travel cost, and it usually refers to user equilibrium (UE) (Wardrop, 1952; Smith, 1979). It is also known that the system optimum (SO) can be achieved if users cooperate to match a collective optimized function (i.e., minimizing total travel time) (Beckmann et al., 1956; Mahmassani and Peeta, 1993). Some studies show that SO can save up to 33 % of total travel costs in comparison to UE (Roughgarden and Tardos, 2002; Youn et al., 2008; van Essen et al., 2016). However, SO is achieved in return for higher individual travel costs experienced by some users. Therefore, instead of changing the paths of all users to achieve the perfect SO, it would be more efficient to focus on those who contribute the highest marginal gain in total travel cost to the whole system in return for a reasonable increase of their own travel costs.

Fortunately, researchers have found that only a small share of network users contribute most to system under-performance (Wang et al., 2012; Ameli et al., 2020b). The main challenges are how to identify (i) these users and

---

\*Corresponding author. Tel. : +33 (0) 4 72 04 77 16.

Email address: ludovic.leclercq@univ-eiffel.fr (Ludovic Leclercq)

17 (ii) their alternative paths. To tackle these problems, we rely on trajectory data analysis and supervised learning  
18 models. In the past few decades, various trajectory data have become available. These data help engineers, decision  
19 makers and researchers to propose corresponding strategies for improving urban mobility (Gonzalez et al., 2008; Ma  
20 et al., 2015; Saeedmanesh and Geroliminis, 2016; Lopez et al., 2017). For example, with detailed Global Positioning  
21 System (GPS) data from mobile phones, Wang et al. (2012) showed that the congestion of a given network is mostly  
22 due to very few network users who travel on the most congested road segments. However, this conclusion was  
23 obtained by removing part of the traffic demand from certain origin-destination pairs (O-D pairs) without giving  
24 alternative routing solutions. Çolak et al. (2016) showed that if 10 % of drivers adjust their routing behavior under SO  
25 conditions, the whole system benefits from 40 % of the potential travel time saving that could be achieved if all users  
26 behaved unselfishly. Nevertheless, their conclusion was based on a traffic assignment model where the travel time on  
27 links depended only on its volume-over-capacity (VoC) ratio (BUREAU, 1964), i.e., the links between travel times  
28 are time-independent. Travel time delay is modeled by vertical queues, without considering spillbacks in congested  
29 situations. This is not representative enough of how congestion spreads in dense urban areas. In addition, the SO path  
30 distribution cannot be easily known in real life (Peeta and Mahmassani, 1995; Yildirimoglu and Kahraman, 2018) so  
31 that we are not able to ensure whether these users are traveling under SO conditions or not.

32 In this work, we present methods to both target the most contributive users, i.e., those with the highest marginal  
33 total travel time gain, and re-route them to alternative optimal paths. These methods are designed considering a  
34 dynamic framework. The main objective is to define a small, simple, and relevant set of trajectory features that  
35 determine the trips to be modified. Briefly speaking, trajectory features are the characteristics of a user’s travel  
36 pattern, such as its length, travel time, average travel speed of the path, etc. In addition, the characteristic of the nodes  
37 and links on the pattern can also be taken into consideration, such as their topological features (degree, betweenness  
38 centrality, link lengths, etc), and traffic-related features (node/link capacity, traffic light cycles, etc). Since we aim to  
39 find the targeted users in practice based only on their trajectory features, these features should be kept easy to access  
40 in real-time. This is why we will consider only network features or regular travel time information without requiring  
41 expensive network monitoring and/or a prediction system. The second objective is to define alternative routes for  
42 the selected users by using the same features. A previous work presented in Chen and Leclercq (2019b) shows that  
43 that by rerouting a small part of users according to a single type of trajectory feature can improve the total system  
44 performance. However, this work will focus on selecting and rerouting users based on a combination of different  
45 trajectory features.

46 In this paper, we rely on results from dynamic UE and SO dynamic traffic assignment (DTA) simulations with the  
47 help of a microscopic simulator to investigate the differences in trajectory features. Note that with the same traffic  
48 demand on the same network, the assigned paths in UE and SO simulations can be compared pair-by-pair (Leclercq  
49 et al., 2016). This fact gives us access to the full picture of how UE / SO patterns differ. However, our purpose is to  
50 define a generic methodology that can rely on UE patterns based only on simple network-related features to perform  
51 user selection and re-routing. In addition, using a microscopic simulation framework allows us to easily assess network  
52 performance after selection and re-routing and provide a clear benchmark of our method. The contribution of this work  
53 is threefold:

- 54 • using comparative feature analysis of trajectories obtained in UE and SO simulations, we define the network-  
55 related trajectory features that unravel the trips contributing most to network under-performance;
- 56 • using the network-related trajectory features defined, we propose re-routing strategies for the selected users in  
57 order to improve total network performance (e.g., the total travel times of all vehicles);
- 58 • by applying machine learning techniques to vehicle trajectory data, we improve the total system performance  
59 of the network by changing the paths of a small share of network users without cumbersome DTA simulations.

60 One of the key contributions is that once the trajectory features are identified and the learning models are built, we  
61 are able to carry out user selection and re-routing strategies without knowing the actual network or trajectory features  
62 at SO state. The purposes of running a reference SO simulation are: (i) to provide the data set for training our machine  
63 learning models, and (ii) to serve as a criterion of the *optimal* network performance when we evaluate the efficiency  
64 of our method. However, during the method evaluation, it is still assumed that all the network users travel under  
65 UE perspectives and only their trajectory features at UE state are accessible. Under these assumptions, we can still

66 improve the whole network performance by targeting users and moving them to predicted alternative paths, based only  
67 on their trajectory features and/or regular travel time information under UE condition, thanks to our proposed learning  
68 models. This makes it possible to solve real-world congestion problems without cumbersome network monitoring  
69 and prediction.

70 The remainder of this paper is organized as follows. Firstly, we present the case study description, the overview  
71 of the method and the definitions of trajectory features in Section 2. Then Section 3 presents the case study results  
72 and discussions. Finally, Section 4 presents the main conclusions and future research perspectives of this work.

## 73 2. Methodology

74 This section presents the methodological framework, the definitions of trajectory features, and mathematical for-  
75 mulations of the supervised learning models used in this work. Table 1 shows the main notations.

### 76 2.1. Method overview

77 This subsection mainly focuses on the general methods without entering into detailed computation. Figure 1  
78 presents the overall framework of our methodology. We carry out two reference DTA simulations under UE and SO  
79 conditions, with the same dynamic OD matrix and road network as inputs. These two scenarios are denoted as UE-ref  
80 and SO-ref, respectively. The main idea is to select  $\alpha$  % of the users, give them rerouting guidance, and evaluate  
81 whether the system’s performance is improved. Therefore, there are two important phases in the methodology: (i) the  
82 selection of targeted users and (ii) the determination of rerouting strategies. The total travel time (TTT) of all users  
83 in the whole network is considered as the criterion for evaluating the system’s performance. **The trajectories obtained  
84 in the SO simulation, named as SO paths/trajectories in the paper, are assumed to be the optimal paths for the system  
85 optimum. In fact, they are the paths that bring the least marginal TTT to the whole system when additional users  
86 travel in the network through these paths.** The features of SO trajectories are used only for the construction of the data  
87 set. The main objective is to achieve a significant TTT reduction of the whole system by moving a small portion of the  
88 users onto their *predicted* SO trajectories, based on the application of data mining and supervised learning methods to  
89 trajectory feature data.

90 For each step, we first need to prepare the data set of trajectory features obtained from UE-ref and SO-ref, as  
91 shown in the blocks in the data preparation part of Figure 1. Based on the literature review, there are some key  
92 network features that contribute to system underperformance, such as the betweenness centrality (BC) of nodes and  
93 links (Freeman, 1977; Wang et al., 2008, 2012; Chen and Leclercq, 2019a,b; Bellocchi et al., 2020), corridor capacity  
94 (Laval and Castrillón, 2015), the distribution and cycle of traffic lights (Laval and Castrillón, 2015), etc. Section 2.3  
95 will present the detailed definitions of the network-based trajectory features analyzed in this work. However, we first  
96 introduce the definition and computational methods for obtaining the path marginal cost (PMC).

97 In particular, we consider the differential of path marginal costs (diffPMC) between the UE and SO states of the  
98 network as the indicator to evaluate the marginal total travel gain that can be achieved by moving a user and then  
99 assess its potential contribution to improving the system’s performance. Indeed, the marginal cost of a path represents  
100 the *increase* of travel cost to the whole system due to an additional vehicle on that path. In the SO state, users are  
101 on routes with equal and minimum PMC so that no users can shift to any other paths that have a lower marginal  
102 travel cost (Sheffi, 1985). Therefore, the re-assignment of the users with the largest PMC difference between UE and  
103 SO can shift the system close to the system optimum. However, this is not an option in practice as PMC can hardly  
104 be estimated in the reference situation (UE), and the differences in PMC are unavailable because the SO solution is  
105 unknown. This is the reason why we rely on analyzing simulated trajectory data. Here, we focus on network-related  
106 trajectory features that are easy to obtain in practice. Furthermore, to evaluate the quality of user selection, we are  
107 going to run new simulations considering that the selected users now have new predefined routes while the others keep  
108 traveling under the UE discipline. This permits us to account for the influence of the route guidance of the selected  
109 users on the other users in the system. In this work, we use a simulation-based method to compute PMC. Detailed  
110 definitions and computation methods are presented in Section 2.2.

Table 1: List of notations

Notations	Explanations
$T$	Travel time
$\mathcal{H}$	Planning horizon (the duration of $\mathcal{H}$ is 3600 minutes)
$TTT$	The total travel time of the network
$\Delta TTT_{\text{relative}}$	Relative TTT reduction with respect to UE-ref simulation
$\Delta TTT_{\text{relative w.r.t ref reduction}}$	Relative TTT reduction with respect to the <i>optimal</i> TTT reduction: $TTT_{\text{UE-ref}} - TTT_{\text{SO-ref}}$
LTT	Link travel time
PTT	Path travel time
LMC	Link marginal cost
PMC, $\widehat{\text{PMC}}$	Path marginal cost and dimensionless path marginal cost (normalized by free-flow travel time)
diffPMC	The difference between PMC of UE pattern and SO pattern
BC	Abbreviation of Betweenness Centrality
$i$	Index of user
$j$	Index of link
$p, P$	Index of trajectory feature, and the total number of analyzed trajectory features
$\alpha, \alpha^*$	Fraction and the <i>optimal</i> fraction of users that are selected to change trajectories
$x, \mathbf{x}, \mathbf{X}$	Scalar, vector, and matrix of attributes composed by trajectory feature(s)
$y, \mathbf{y}$	Target scalar and target vector to be predicted and they are obtained from the simulation data set
$y', \mathbf{y}'$	Scalar and vector <i>predicted</i> by supervised-learning model, with feature vectors as attributes
$w, \mathbf{w}$	Scalar and vector of regression coefficient
$f(\cdot)$	The learning model for targeted user selection, with trajectory features as input and diffPMC as output
$g(\cdot)$	The classifier for labeling whether a path is an SO path, with trajectory features as input and the path's class as output
$C$	Class of labels of a path
$\Delta\tau$	The snapshot time step of the microscopic simulator to collect simulation results (60s in this work)
$n(\tau)$	Spatially-averaged number of vehicles on a link during the time step $[\tau, \tau + \Delta\tau]$
$\phi(\cdot)$	Link performance function: the input is $n(\tau)$ and the output is link travel time
$\Delta t$	The time step for computing path travel time and path marginal cost (60s in this work)
$\mathcal{D}$	The total number of OD pairs in the network
$s_{i,d}$	The number of available alternative paths of the user $i$ to the destination $d$ , $d \in \mathbb{R}^D$
$\mathcal{G}$	The directed graph representing the network
$\mathcal{A}$	The set of links on the graph
$\mathcal{N}$	The set of nodes on the graph
$\mathcal{L}_i$	The trajectory of user $i$
$L_i$	Length of trajectory $\mathcal{L}_i$
$J_i, K_i$	Number of links and nodes on trajectory $\mathcal{L}_i$
$\mu_l$	Mean distance between two consecutive traffic lights on a trajectory
$\sigma_l$	Standard deviation of distances between two consecutive traffic lights on a trajectory
$\bar{\mu}_l$	Dimensionless mean distance of two consecutive traffic lights on a trajectory
$G_i^{\text{BC}}$	Gini coefficient of node betweenness centrality on trajectory $\mathcal{L}_i$
$G_i^{\text{dynamicBC}}$	Gini coefficient of dynamic link betweenness centrality on trajectory $\mathcal{L}_i$
$Q$	One-lane link saturation capacity during green time
$t_g, t_r$	Green time and red time of traffic light cycle
$\mu_g, \mu_r, \sigma_g, \sigma_r$	Mean green / red time, and the corresponding standard deviations of traffic lights on a trajectory
$\delta$	The coefficient of variance: $\frac{\sigma}{\mu}$
$\text{Cap}_i$	Mean MFD capacity of trajectory $\mathcal{L}_i$
$\eta_i^{\text{critical}}$	Normalized length of critical links on trajectory $\mathcal{L}_i$
$E_{i,j}$	Dynamic link betweenness centrality of link $j$ on trajectory $\mathcal{L}_i$
$\bar{E}_i, \widehat{E}_i$	Mean and value of dynamic betweenness centralities of all the links on trajectory $\mathcal{L}_i$
$\text{BC}_i, \widehat{\text{BC}}_i$	Mean and median of betweenness centralities of all nodes on a trajectory $\mathcal{L}_i$

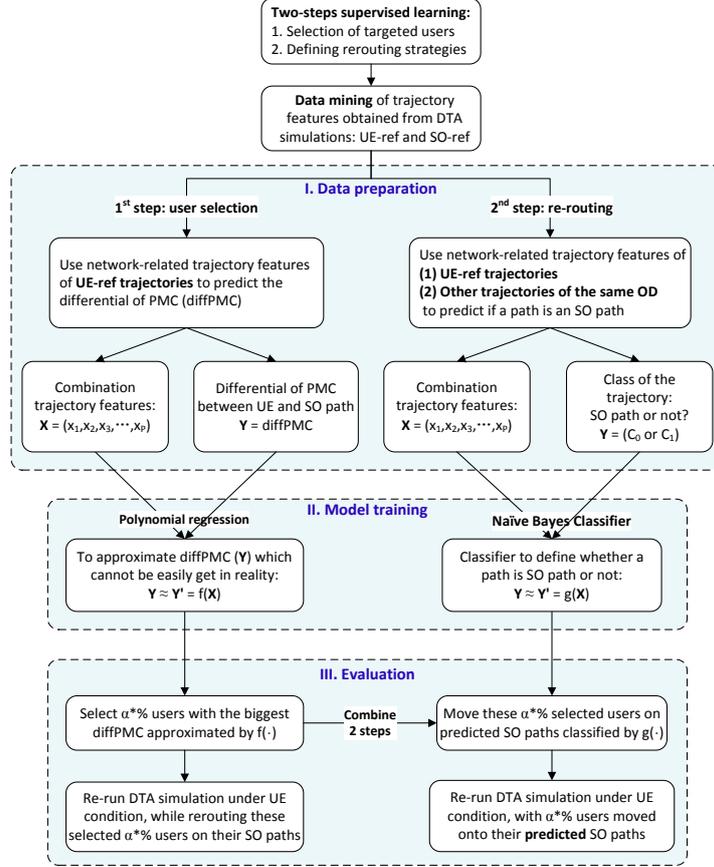


Figure 1: Methodology flow chart of this work

## 2.2. Link marginal cost (LMC) and path marginal cost (PMC)

The path marginal costs are computed based on time-dependent link marginal costs (LMC). LMC is defined as the change of link travel time caused by one additional unit of flow on the link. In the DTA simulation, LMC is time-dependent and can be computed by Equation (1) (Peeta and Mahmassani, 1995):

$$LMC_j(t, n) = T_j(t, n) + n(t) \times \frac{\partial T_j(t, n(t))}{\partial n(t)}, \quad (1)$$

where  $T_j(t)$  denotes the link travel time at  $t$  on link  $a_j$ ,  $n(t)$  denotes the spatially-averaged number of vehicles on link  $a_j$ . The challenge is to calculate  $\frac{\partial T_j(t, n(t))}{\partial n(t)}$ . We assume that (i) the vehicle type on the network is single (i.e., mono class user), and (ii) the traffic light plan of the whole network is given. Therefore, the link marginal cost can be considered as an intrinsic feature of the link, and it depends only on  $n$ . It can be estimated via a simulation-based method. In this work, we use Symuvia<sup>1</sup> platform for dynamic traffic simulations. Symuvia is a dynamic microscopic trip-based simulator based on a Lagrangian discretization of the LWR model (Leclercq et al., 2007). Here, we present how we use Symuvia to compute simulation-based LMC. For the details of the traffic simulation, readers can refer to Ameli et al. (2020c).

The main idea of simulation-based methods is to approximate a link performance function, i.e., link travel time (LTT) vs. accumulation, for every link of the network. Then the LMC can be easily calculated by the deviation. To get

<sup>1</sup>Note that Symuvia is an open-source simulator which is available on GitHub: <https://github.com/Ifsttar/Open-SymuVia>

link performance functions, the main steps for computing LMC are as follows. First, we run a reference simulation with different traffic demand profiles. During each time step  $[\tau, \tau + \Delta\tau]$ , we compute the average travel time ( $T_{\tau,j}$ ) and the spatially-averaged number of vehicles ( $n_{\tau,j}$ ) on link  $a_j$ , and then save them as sample points  $\{(n_{\tau,j}, T_{\tau,j})\}$ . Second, for each link, we compute its maximum  $n$  by multiplying its length with the maximum density, and then split the values of  $n$  into  $\mathcal{K}$  equal-width classes. Then for each class  $k$ , we gather all the points  $(n_{\tau,j}, T_{\tau,j})$  of which  $n_{\tau,j}$  belongs to the class  $k$ , and compute the corresponding average values of  $n$  and  $T$  during the whole simulation period, in order to get  $(n_{k,j}, T_{k,j})$  for link  $a_j$ . At last, we fit a regression model representing a time-dependent link performance function  $\phi_j : n \rightarrow T$ , based sample points of  $(n_{k,j}, T_{k,j})$ . In this work, we choose linear and/or quadratic regression models for fitting link performance functions. Therefore, during the time step  $[\tau, \tau + \Delta\tau]$ , we can compute  $n(\tau)$  and  $\frac{dT}{dn(\tau)}$  for a given time  $\tau$ . By replacing  $T_j$  in Equation (1) by  $\phi_j$ , the LMC of  $a_j$  can be computed by Equation (2). It is worth mentioning that the estimated performance function  $\phi_j$  is only used to compute LMC and then path marginal costs. The path travel times of users are still obtained from the results of microscopic simulator Symuvia, by computing the difference between the arrival time and enter time of users. These path travel times are used to compute User Equilibrium at each iteration.

$$LMC_j(\tau, n) = \phi_j(n(\tau)) + n(\tau) \times \frac{d\phi_j(n)}{dn} \Big|_{n=n(\tau)}, \quad (2)$$

where  $n(\tau)$  is estimated as  $n(\tau) = \frac{TTT(\tau)}{\Delta\tau}$ .  $TTT(\tau)$  denotes the total travel time of all users travelling on link  $a_j$  during that time step  $[\tau, \tau + \Delta\tau]$ .

Therefore, the PMC on a trajectory  $\mathcal{L}_i$  can then be obtained by summing up all the time-dependent LMC on the trajectory (Peeta and Mahmassani, 1995). The time step for computing PMC is  $\Delta t = 60$  s in our case study. The total number of time steps is denoted by  $|\mathcal{H}|$ .  $PMC_i$  is then computed as follows:

$$PMC_i = \sum_t^{|\mathcal{H}|} \sum_j LMC_{j,t} \delta_{j,t}, \quad (3)$$

where  $\delta_{j,t}$  is the incidence indicator. If the entering time of user  $i$  on the link  $a_j$  belongs to the time interval  $[t, t + \Delta t]$ , then  $\delta_{j,t} = 1$ . Otherwise,  $\delta_{j,t} = 0$ . The computation of LMC and PMC is time-dependent. In SO-based DTA based on microscopic simulator, the PMCs are computed after network loading at each iteration. The links passed through by a user  $i$  are already known, and the average number of vehicles (denoted as  $n(t)$ ) of all links at all time intervals  $[t, t + \Delta t]$  are already computed, too. To compute the PMC of the user  $i$ , we check its entering time on each link that is on the current loaded path of the user  $i$ , and see the entering time belongs to which time interval. Then we use the corresponding  $n(t)$  to compute LMC(t) based on Equation(2), and sum up all LMCs of all links on the path to get the PMC.

When we analyze trajectory features in order to determine which users should be re-routed and which alternative paths should be chosen, it is assumed that we don't have real-time arrival information of users on each link of their paths. Therefore, the computations of LMC, PMC and diffPMC are *departure-time centered*. In other words, we compute PMC and diffPMC of a user  $i$  by using Equation (3), where  $t$  corresponds to its entering time onto the network. In addition, it is worth noticing that the diffPMC is computed based only on features obtained in the UE simulation. Indeed, user  $i$  has a pair of paths resulting from the UE-ref and SO-ref simulations. As we assume that moving the user from the UE path to the SO path should bring the system close to its SO, users with a high diffPMC value should be selected first, as mentioned in Section 2.1. However, when we try to re-route a user who is traveling on the network under the UE condition, we only have access to traffic-state characteristic (LTTs and LMCs) in the UE state. Therefore, even though we can move it to its SO path, the diffPMC from UE to SO path should be computed based on the LMCs obtained in UE-ref. To be more precise, we denote  $LMC_{j,t}^{UE}$  as the LMC of link  $a_j$  computed from the UE-ref simulation during the time step  $[t, t + \Delta t]$ . We denote  $PMC_i^{UE}$  and  $PMC_i^{SO}$  as the PMCs of the UE and the SO path for user  $i$ , respectively. These variables are computed by Equation (4):

$$PMC_i^{UE} = \sum_t^{|\mathcal{H}|} \sum_j LMC_{j,t}^{UE} \delta_{j,t}^{UE}, \quad PMC_i^{SO} = \sum_t^{|\mathcal{H}|} \sum_j LMC_{j,t}^{UE} \delta_{j,t}^{SO}, \quad (4)$$

165 where  $\delta_{j,t}^{\text{UE}}$  and  $\delta_{j,t}^{\text{SO}}$  denote the incidence indicators defined in Equation (5) and Equation (6), respectively.

$$\delta_{j,t}^{\text{UE}} = \begin{cases} 1 & \text{if the user } i \text{ is on the link } a_j \text{ during } [t, t + \Delta t] \text{ in UE-ref simulation;} \\ 0 & \text{Otherwise.} \end{cases} \quad (5)$$

$$\delta_{j,t}^{\text{SO}} = \begin{cases} 1 & \text{if the user } i \text{ is on the link } a_j \text{ during } [t, t + \Delta t] \text{ in SO-ref simulation;} \\ 0 & \text{Otherwise.} \end{cases} \quad (6)$$

166 Therefore, the diffPMC of the user  $i$  in this work is calculated by the Equation (7).

$$\text{diffPMC}_i = \text{PMC}_i^{\text{UE}} - \text{PMC}_i^{\text{SO}} \quad (7)$$

167 It is worth mentioning that users may have the same paths under both UE and SO conditions. Therefore, they are  
 168 already on their optimal paths under both UE and SO perspectives, and will not be chosen as users to be re-routed in  
 169 this work. The definitions of PMC and diffPMC in Equation (4) to Equation (7) confirm that. In fact, if a user has  
 170 the same path in both UE and SO simulation, its diffPMC is zero. Since we aim to re-route the users who have the  
 171 largest value of diffPMC, these users with zero diffPMC would not be chosen. The value of diffPMC is the target  
 172 value  $y$  that we aim to train in Section 2.5. Before presenting the learning process, we introduce detailed definitions  
 173 and formulations of the network-related trajectory features that are used in this work in the following Section 2.3.

### 174 2.3. Network-related trajectory features

175 Network-related trajectory features are obtained by aggregating features of links and/or nodes. Here we use a  
 176 modeled network to define different network-related trajectory features. A road network is modeled as a directed  
 177 graph  $\mathcal{G} = (\mathcal{N}, \mathcal{A})$ , composed with  $K$  nodes and  $J$  links.  $\mathcal{N} = \{n_1, n_2, \dots, n_i, \dots, n_K\}$  is the set of nodes and  $\mathcal{A} =$   
 178  $\{a_1, \dots, a_j, \dots, a_J\}$  is the set of links. A trajectory  $\mathcal{L}_i$  with length  $L_i$  is composed by a set of links ( $a_j$ ) and nodes ( $n_k$ ):

$$\mathcal{L}_i = \{\{a_{i,1}, \dots, a_{i,j}, \dots, a_{i,J_i}\}, \{n_{i,1}, \dots, n_{i,k}, \dots, n_{i,K_i}\}\}, \quad (8)$$

179 where  $J_i$  and  $K_i$  denote the number of links and the number of nodes on  $\mathcal{L}_i$ ,  $a_{i,j} \in \mathcal{A}$ ,  $n_{i,k} \in \mathcal{N}$ . Therefore, network-  
 180 related trajectory features of  $\mathcal{L}_i$  result from aggregating network features of ( $a_{i,j}$ ) and/or ( $n_{i,k}$ ). There are two categories  
 181 of network features to be aggregated: (i) graph-theory based and (ii) traffic-state based. **We select the features based**  
 182 **on a literature review, and keep those which can be calculate in real-time with limited information (such as monitoring,**  
 183 **network traffic light system, and current travel time).**

#### 184 2.3.1. Graph-theory based features: aggregation of centrality metrics

185 The aggregate metrics of *critical* nodes on a trajectory can be considered as one of its network-related trajectory  
 186 features. In graph theory, there are several metrics to define whether a node is critical to the whole network. For  
 187 example, the *betweenness centrality* ( $BC$ ) of a node  $n$  corresponds to the ratio of shortest paths crossing  $n$  over all  
 188 possible shortest paths for all origin-destination pairs of the network (Freeman, 1977; Girvan and Newman, 2002).  
 189 The  $BC$  of node  $n$  is calculated by Equation (9).

$$BC(n) = \sum_{i \neq j} \frac{\sigma_{ij}(n)}{\sigma_{ij}}, \quad (9)$$

190 where  $\sigma_{ij}(n)$  denotes the number of shortest paths from any node  $n_i$  to any node  $n_j$  crossing node  $n$ , and  $\sigma_{ij}$  denotes  
 191 the total number of shortest paths from  $n_i$  to  $n_j$ . In the case study in this work, the *shortest paths* for calculating  $BC$

192 are measured by distance based on the topological parameters of the network, without taking into account the dynamic  
 193 traffic condition. Therefore, for trajectory  $\mathcal{L}_i$ , the mean node BC of  $\mathcal{L}_i$  is defined as follows:

$$\overline{BC}_i = \frac{1}{K_i} \sum_{n_{i,j}} BC(n_{i,j}). \quad (10)$$

194 where  $K_i$  denotes the number of nodes on  $\mathcal{L}_i$ . In addition to the mean value, we can also compute other statistical  
 195 metrics of node BCs on  $\mathcal{L}_i$ , such as the median value of all node BCs, denoted as  $\widehat{BC}_i$ .

196 There may be some critical nodes on a trajectory that are the main contributions to the congestion. Therefore,  
 197 we also consider the Gini coefficient of node BCs as a network-related trajectory feature to be analyzed. The Gini  
 198 coefficient ( $G$ ) is a measure of inequality, defined as the mean of absolute differences between all pairs of variables  
 199 for a particular measure. The minimum value is 0 when all the measurements are equal. The theoretical maximum is  
 200 1 for an infinitely large set of observations where all the measurements but one have a value of 0, which is ultimate  
 201 inequality.  $G$  is calculated by equation (11).

$$G = \frac{\sum_{i=1}^n \sum_{j=1}^n |x_i - x_j|}{2n^2 \bar{x}}, \quad (11)$$

202 where  $x$  denotes an observed value,  $n$  denotes the number of values observed, and  $\bar{x}$  denotes the mean value of all  $x$ .  
 203 In this work, for a given trajectory, the Gini coefficient of node BC can be considered as a measure of the existence of  
 204 nodes with a large BC value on a trajectory. The Gini coefficient of node BC on a trajectory  $\mathcal{L}_i$  can be calculated by:

$$G_i^{\text{BC}} = \frac{\sum_{j=1}^{K_i} \sum_{h=1}^{K_i} |BC_{i,j} - BC_{i,h}|}{2K_i^2 \overline{BC}_i}, \quad (12)$$

205 where  $K_i$  denotes the number of nodes (intersections) on  $\mathcal{L}_i$ .  $BC_{i,j}$  and  $BC_{i,h}$  denote the BC of the  $j^{\text{th}}$  and  $h^{\text{th}}$  node on  
 206  $\mathcal{L}_i$ .  $\overline{BC}_i$  denotes the mean value of all node BCs on a trajectory  $\mathcal{L}_i$ .

### 207 2.3.2. Graph-theory based features: mean distance between traffic lights

208 In urban areas, the distribution of traffic lights on the network can affect the route choices of users, and contribute  
 209 to the system's congestion. Here we present a dimensionless feature that can interpret the distribution of traffic lights  
 210 on a path. For each trajectory  $\mathcal{L}_i$ , we compute the mean distance between two consecutive intersections with traffic  
 211 lights (denoted as  $\mu_l$ ). Then we normalize it by dividing the length of  $\mathcal{L}_i$ :

$$\bar{\mu}_l = \frac{\mu_l}{L_i}. \quad (13)$$

212 In the two subsections above (Section 2.3.1 and Section 2.3.2), we present several important network-related  
 213 trajectory features that depend only on the topological characteristics of the network. In the next subsection, we  
 214 introduce several features that take the dynamic traffic states into account.

### 215 2.3.3. Traffic-state features: mean MFD trajectory capacity

216 An indicator of trajectory capacity is calculated by aggregating simple traffic-state features (e.g., fundamental  
 217 diagram) and traffic light data of links on the trajectory. Let  $Q$  denote saturation capacity during the green time. For  
 218 links with traffic lights, the outflow capacity of a link is calculated as  $\frac{t_g}{t_g+t_r} Q$ , where  $t_g$  and  $t_r$  represent the green time  
 219 and red time of the traffic light cycle. With known link lengths and a given traffic light plan, we can compute the mean  
 220 MFD capacity for trajectory  $\mathcal{L}$ , based on the conclusion of [Laval and Castrillón \(2015\)](#). A trajectory is considered  
 221 as a one-lane corridor composed of several road segments separated by traffic lights. Let  $\mu_l$  and  $\sigma_l$  denote the mean  
 222 distance between two consecutive traffic lights and the corresponding standard deviation.  $\mu_g, \mu_r, \sigma_g$  and  $\sigma_r$  denote the  
 223 mean green time, mean red time, and the corresponding standard deviation of traffic lights on  $\mathcal{L}$ . Let  $\delta = \frac{\sigma}{\mu}$  denotes the

224 coefficient of variance. The mean MFD capacity is then determined by three dimensionless values: (i) mean red time  
 225 over mean green time ratio:  $\rho = \frac{\mu_r}{\mu_g}$ , (ii) mean block length to mean green time ratio:  $\lambda = \frac{\mu_l}{\mu_g}$ , and (iii) the coefficient  
 226 of variance of green light time, red-light time, and block length. It is assumed that the coefficient of variance is the  
 227 same for these three variables, denoted as  $\delta$ . With these parameters computed for a trajectory, an approximation for  
 228 the mean corridor capacity is given in Laval and Castrillón (2015) and the formulation is presented in Equation (14).  
 229 To make the features dimensionless for the analysis, the traffic flow is represented in the unit of  $Q$ , and the density is  
 230 presented in jam density units.

$$\text{Cap} = \min\left\{\frac{1}{1 + \rho(0.58\delta\lambda + 1.64\lambda^2 - 5.3\lambda + 4.99)}; \frac{\mu_g}{\mu_g + \mu_r}\right\}. \quad (14)$$

231 The mean MFD capacity of a trajectory is considered as a network-related feature. All the parameters of Equation (14)  
 232 can be obtained by aggregating a sequence of link lengths between each pair of constructive traffic lights, and the traffic  
 233 cycle data of links on the trajectory.

234 Besides the MFD capacity, the percentage of critical links on a path can also indicate whether this trajectory  
 235 contributes to the performance of the whole network. In this work, a link is considered as *critical* if it has a relatively  
 236 small saturation capacity  $Q$ . In this work, we consider  $a_{i,j}$  as *critical* link if  $Q_j \leq 110\% \text{Cap}_i Q$ , where  $Q_j = \frac{t_g^j}{t_g^i + t_r^i} Q$ .  
 237 Therefore, the normalized length of critical links on  $\mathcal{L}_i$  reads:

$$\eta_i^{\text{critical}} = \frac{\sum_{\text{critical link } l} l}{L_i}. \quad (15)$$

238  $\eta_i^{\text{critical}}$  is considered as metrics to quantify trajectory features related to critical links.

#### 239 2.3.4. Traffic-state features: dynamic betweenness centrality of links

240 Bellocchi et al. (2020) define the concept of *reachability* of links by taking into account the dynamic traffic-state  
 241 features of the network, i.e., the dynamic link betweenness centrality. When the average speed of each link  $a_{i,j}$  is  
 242 known, we can consider the following measure calculated as  $E(a_{i,j}(t)) = \frac{e_i(t) + e_j(t)}{2}$ , where

$$e_i(t) = \frac{1}{K-1} \sum_{h \in \mathcal{N} \setminus i} \frac{d_{kh}^{ff}}{\tau_{kh}(t)}, \quad (16)$$

243 where  $d_{kh}^{ff}$  denotes the travel time of the shortest time path in free-flow condition between nodes  $k$  and  $h$  (for all  
 244  $h \in \mathcal{N} \setminus k$ ).  $\tau_{kh}(t)$  denotes the actual path travel time (PTT) of the shortest-time path at time  $t$ , between the same pair  
 245 of Origin-Destination nodes. Note that when we compute  $e_i(t)$  here,  $t$  corresponds to the *departure time* of user  $i$  on  
 246 the network. The PTT of user  $i$  is in fact computed by summing up all the *departure-time-centered* LTTs of links that  
 247 are on the trajectory  $\mathcal{L}_i$ . The LTTs can be obtained by a simple monitoring system without any dynamic prediction.  
 248  $K$  denotes the total number of nodes in the network.  $E(a(t))$  denotes a measure of the accessibility of link  $a = (i, j)$   
 249 and it is evaluated with a number between 0 and 1. The closer the value of  $E(a(t))$  is to 1.0, the less congested link  $a$   
 250 is. Then, for trajectory  $\mathcal{L}_i$ , we can compute the mean and median values of the dynamic link BCs on it:  $\bar{E}_i$  and  $\widehat{E}_i$ .

251 Similarly, for a given trajectory  $\mathcal{L}_i$ , we can also compute the Gini coefficient of dynamic BCs of all the links on  
 252  $\mathcal{L}_i$  by the following equation:

$$G_i^{\text{dynamicBC}} = \frac{\sum_{j=1}^{J_i} \sum_{h=1}^{J_i} |E_{i,j} - E_{i,h}|}{2J_i^2 \bar{E}_i}, \quad (17)$$

253 where  $J_i$  denotes the number of links on  $\mathcal{L}_i$ .  $E_{i,j}$  denotes the dynamic link BC of link  $a_j$  on  $\mathcal{L}_i$ .  $\bar{E}_i$  denotes the mean  
 254 value of all the dynamic link BCs on the same trajectory  $\mathcal{L}_i$  and it is computed by  $\bar{E}_i = \frac{1}{J_i} \sum_{j=1}^{J_i} E_{i,j}$ .

255 The two trajectory features presented in these two subsections, the mean MFD capacity and the dynamic BC of  
 256 links, take into account the dynamics of the traffic such as the traffic light plan, the monitored LTTs, etc. However,  
 257 they still remain simple to access and easy to calculate in practice.

#### 258 2.4. Feature scaling and normalization

259 Since a trajectory has multiple features spanning varying degrees of magnitude, range, and units, we should carry  
 260 out feature scaling before the application of machine learning techniques. According to their definitions, some of the  
 261 trajectory features defined in Section 2.2 and Section 2.3 are already dimensionless: the median of node BC on  $\mathcal{L}_i$   
 262 ( $\widehat{BC}_i$ ), the Gini coefficient  $G_i^{\text{BC}}$  and  $G_i^{\text{dynamicBC}}$ , the MFD capacity ( $\text{Cap}_i$ ), the normalized length of critical links on a  
 263 trajectory ( $\eta_i^{\text{critical}}$ ), and the normalized average distance between two consecutive intersections with traffic lights ( $\bar{\mu}_i$ ).

264 Regarding the path marginal costs, we normalize the PMC by trajectory free-flow travel time (fftt), as shown in  
 265 Equation (18). **The trajectory free-flow travel time is computed by summing link fftt of the links that are passed**  
 266 **through by the path.** The link fftt can be approximated by dividing the distance of  $a_j$  by the maximum authorized  
 267 speed on  $a_j$ , which depend only on the intrinsic features of a given network.

$$\widehat{\text{PMC}}_i = \frac{\text{PMC}_i}{\text{fftt}_i}. \quad (18)$$

268 Now we have clear definitions and formulations of all the trajectory features used in this work. In the next two  
 269 subsections, Section 2.5 and Section 2.6, we present detailed formulations of the machine learning methods used in  
 270 this work, following the two-step methodological framework introduced in Section 2.1 and Figure 1. **In fact, apart**  
 271 **from the features presented in Section 2.3, there are various kinds of other trajectory features such as path travel time,**  
 272 **path travel distance, degree of nodes, degree of links, etc. According to some previous descriptive analysis and results**  
 273 **presented in Chen and Leclercq (2019b), the betweenness centralities and mean MFD capacities are two potential**  
 274 **features that can define SO paths. That is the reason why we mainly focus on the trajectory features presented in**  
 275 **Section 2.3 in this work.**

#### 276 2.5. First phase: selection of targeted users

277 In this subsection, we present the methods for selecting targeted users based on the users' trajectory features, i.e.,  
 278 the left branch of 1<sup>st</sup>- step shown in Figure 1. The main framework is consistent with the three parts introduced in  
 279 Figure 1: data set preparation, model training, and evaluation.

280 First, we prepare the data set. For all the  $M$  users on the network, we compute their network-related features in  
 281 the reference UE simulation and save them in vector  $\mathbf{x} \in \mathbb{R}^P$ , where  $P$  denotes the number of trajectory features to be  
 282 analyzed. We also compute diffPMC of each user, according to Equation (4) to Equation (7), and we save it in  $y \in \mathbb{R}$ .

283 Then, we carry out supervised learning with these  $M$  training points, in order to obtain  $f : \mathbf{x} \rightarrow y$ , where  $y$  is  
 284 the target scalar to be approximated based on  $\mathbf{x}$ . Therefore, for any user  $i$  with  $P$  trajectory features under the UE  
 285 condition, we can approximate its potential PMC decrease, i.e., diffPMC, by  $y \approx y' = f(\mathbf{x}_i)$ . The value of  $y'$  defines  
 286 whether a user should be targeted to change trajectory. In this work, we mainly rely on linear regression (LR) and  
 287 polynomial regression (PR) to train  $f$  (Seal, 1967; Cameron and Trivedi, 2005). The aim of the LR is to find a column  
 288 vector  $\mathbf{w} = (w_1, w_2, \dots, w_p)^T$  so that  $y = \varepsilon + \mathbf{x}^T \cdot \mathbf{w} = \varepsilon + \sum_{j=0}^P w_j \times x_j$ .  $w_i$  denotes the weight associated to each  
 289 factor in  $\mathbf{x}$ .  $\varepsilon \sim \mathcal{N}(0, \sigma^2)$  is the Gaussian error term. Based on  $M$  samples, we obtain the matrix expression of the  
 290 LR:  $\mathbf{y} = \mathbf{X} \cdot \mathbf{w} + \epsilon$ , with  $\epsilon \in \mathbb{R}^M$  as the vector of estimation error. Then we rely on the  $M$  sample set to estimate  $\mathbf{w}$  by  
 291 minimizing  $\epsilon = \|\mathbf{y} - \mathbf{X} \cdot \mathbf{w}\|^2$  based on the least-squares estimation (Equation (19) and (20)).

$$\mathbf{w} = \arg \min_{\mathbf{w}} \sum_{i=1}^M \|y_i - \mathbf{x}^i \cdot \mathbf{w}\|^2 = \arg \min_{\mathbf{w}} \sum_{i=1}^M \|y_i - \sum_{j=0}^P w_j \times x_j^i\|^2 \quad (19)$$

$$\mathbf{w} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}. \quad (20)$$

292 Moreover, there are often nonlinear relationships between the feature vector and the target. The polynomial regression  
 293 (PR) model is also frequently used to address nonlinear relationships. The PR model is a polynomial transformation  
 294 based on the LR model. For a model with only one variable, the  $k^{\text{th}}$  order polynomial model is given by  $y = w_0 +$   
 295  $w_1 x + w_2 x^2 + \dots + w_k x^k + \varepsilon$ . Here, the dimension of the feature vector is more than one and we should also take into

296 account the interaction terms between different features. For example, a polynomial regression model of the 2<sup>nd</sup> order  
 297 with two variables reads:  $y = w_0 + w_1x_1 + w_2x_2 + w_{11}x_1^2 + w_{22}x_2^2 + w_{12}x_1x_2 + \varepsilon$ . In this work, the dimension of the  
 298 feature vector is  $P = 7$ . We save all the trajectory features of  $M$  users in the matrix of attributes  $X = [\mathbf{x}^1, \dots, \mathbf{x}^M]$ , and  
 299 we save all the diffPMC of these  $M$  users in  $\mathbf{y} \in \mathbb{R}^M$ . Then, we train the LR and PR models based on the data set  
 300 composed with  $X$  and  $\mathbf{y}$ . The detailed implementation and the numerical results are presented in Section 3.2.

301 Lastly, we evaluate the efficiency of our learning method with the steps shown in *part-III* of Figure 1. We re-  
 302 run the DTA simulation by selecting a certain percentage of users according to their  $y'$ , i.e., *predicted* diffPMC, and  
 303 we move them to their *SO paths* simulated by the SO-ref scenario. Therefore, we evaluate whether rerouting these  
 304 selected users can lead to significant network performance improvement or not, by comparing (i) the TTT obtained in  
 305 this evaluation DTA scenario and (ii) the TTTs obtained in UE-ref and SO-ref simulations.

## 306 2.6. Second phase: re-routing strategy based on combining network-related features

307 Up to now, to evaluate the efficiency of the selection step, we have assumed that the *optimal* route guidance  
 308 strategy for all the selected users is to assign them to their SO paths. Here, we relax this assumption. For a selected  
 309 user, among all the paths in the feasible space, we aim to find the *best* path for bringing the system's performance  
 310 closer to the system optimum, without knowing the SO solutions computed via cumbersome DTA simulations. To this  
 311 end, we rely on supervised classification based on the network-related features of trajectories under the UE condition.  
 312 By assuming that SO paths are the *optimal* paths for reaching the system optimum, we aim to classify whether a path  
 313 is the SO solution or not based on network-related trajectory features. This step corresponds to the right branch of the  
 314 2<sup>nd</sup> – step shown in Figure 1. We still need the results from the SO-ref simulation in order to prepare the data set and  
 315 train our classification model. However, once the model is built and validated, we can label whether a path is an SO  
 316 path or not, based only on the trajectory features.

### 317 2.6.1. Data preparation

318 We first prepare the data set by computing the features of trajectories obtained in the UE-ref and SO-ref simula-  
 319 tions. More specifically, we assume that a user  $i$  entering the network has  $s_{i,d}$  available paths, computed from both  
 320 the UE-ref and SO-ref simulations, between its OD pair with  $d$  as the destination,  $d \in \mathbb{R}^D$  where  $D$  denotes the total  
 321 number of OD pairs in the network. Each of these paths has its own vector of features under:  $\mathbf{x}_{s_{i,d}}$ , which are composed  
 322 of values of network-related features presented in Section 2.3 under the UE condition. For all these paths, we label  
 323 them by two trajectory categories: (i)  $C_1$  if the trajectory is the SO solution and (ii)  $C_0$  otherwise. We save the labels  
 324 of all the paths in the target vector  $\mathbf{y}$ . In addition, we compute the trajectory features for all the  $s_{i,d}$  paths, and we save  
 325 the features of all the paths in the matrix of attributes  $X = \{\mathbf{x}_{s_{i,d}}\}_{i \in \mathbb{R}^M, d \in \mathbb{R}^D}$ .

### 326 2.6.2. Model training

327 Using the prepared data set, we now train a model  $g : X \rightarrow y = (C_0 \text{ or } C_1)$ , in order to predict whether a path is an  
 328 SO path or not, based on the feature table  $X$ .

329 In this work, we rely on the Bayes Theorem (Wu et al., 2008; Pearl, 2014; Taheri et al., 2014) to solve the  
 330 classification problem. Classification learning is the process of predicting a discrete class label  $C = \{C_1, \dots, C_m\}$  for a  
 331 test attribute  $X = \{X_1, \dots, X_n\}$  (Wu et al., 2008; Pearl, 2014; Taheri et al., 2014). Here, learning algorithms are given  
 332 the task of training a classifier that assigns a class label to a sample based on the attribute. In this work, we assume  
 333 that there are only two classes: (i)  $C_0$  contains the path that is not an SO path and (ii)  $C_1$  contains the path is labeled  
 334 as an SO path. Then we assign a trajectory  $\mathcal{L}$  to  $C_0$  or  $C_1$ , based on its feature vector  $\mathbf{x} = (x_1, \dots, x_P)^T$ .

335 One effective classifier is the Bayes Classifier. It is constructed based on Bayes' theorem and the probability that  
 336 a given sample belongs to a particular class. Let  $H$  be a hypothesis, such that the trajectory  $\mathcal{L}$  belongs to class  $C$ .  
 337 Our objective is to determine the probability that the hypothesis  $H = \text{Prob}(H|\mathbf{x})$  holds, given that we already know  $\mathbf{x}$ .  
 338 According to Bayes' theorem,  $\text{Prob}(H|\mathbf{x})$  can be expressed as:

$$339 \text{Prob}(H|\mathbf{x}) = \frac{\text{Prob}(\mathbf{x}|H)\text{Prob}(H)}{\text{Prob}(\mathbf{x})}, \quad (21)$$

339 where  $Prob(\mathbf{x}|H)$  denotes the posterior probability of  $\mathbf{x}$  conditioned on  $H$ .  $Prob(H)$  and  $Prob(\mathbf{x})$  denote the prior  
 340 probability of  $H$  and  $\mathbf{x}$ , respectively. When there are only two classes, a sample with attribute  $\mathbf{x}$  is labeled as the class  
 341  $C = C_0$  if and only if the following condition is met.

$$g_B(\mathbf{x}) = \frac{Prob(C = C_0|\mathbf{x})}{Prob(C = C_1|\mathbf{x})} \geq 1, \quad (22)$$

342 where  $g_B(\mathbf{x})$  is called the Bayesian classifier.

343 The probabilities in Equation (21) can be estimated by the data set used for the learning process. However, the  
 344 estimation of  $Prob(\mathbf{x}|C)$  is non-trivial and learning an optimal Bayesian network is an NP-hard problem (Chickering,  
 345 1996; Chickering et al., 2004; Taheri et al., 2014). To simplify the problem, we assume that all the attributes are  
 346 independent given the hypothesis, which leads to

$$Prob(\mathbf{x}|H) = Prob(x_1, \dots, x_P|H) \approx \prod_{j=1}^P Prob(x_j|H), \quad j \in \{1, 2, \dots, P\}, \quad (23)$$

347 where  $P$  denotes the dimension of  $\mathbf{x}$ , i.e., the number of features to be analyzed for the trajectories in this work. Then  
 348 the classifier in Equation (22) reads:

$$g^{NB}(\mathbf{x}) \approx \frac{Prob(C = C_0)}{Prob(C = C_1)} \prod_{j=1}^P \frac{Prob(x_j|C = C_0)}{Prob(x_j|C = C_1)}. \quad (24)$$

349  $g^{NB}(\mathbf{x})$  is called *Naive Bayes (NB)* classifier. The assumption that all the attributes are independent given that  $H$  is  
 350 called *conditional independence*. The NB classifier is one of the most efficient and effective learning algorithms for  
 351 discrete classification problems (Kotsiantis et al., 2007; Wu et al., 2008).

352 We use the data from UE and SO reference simulations to train the NB classifier in this work. All the paths used  
 353 are identified in both simulations for each OD pair. These paths are added to the set of alternative paths. Then, we  
 354 compute the feature vectors  $\mathbf{x}$  for all paths. For a trajectory  $\mathcal{L}$  between a given OD pair, we define the class variable  $y$   
 355 as:

$$y = \begin{cases} 0 & \text{if } C_{\mathcal{L}} = C_0; \\ 1 & \text{if } C_{\mathcal{L}} = C_1. \end{cases} \quad (25)$$

356 Therefore, the NB classifier to be trained is defined as follows:

$$\hat{y} = g^{NB}(\mathbf{x}) = \arg \max_y Prob(y) \prod_{j=1}^P P(x_j|y). \quad (26)$$

357  $Prob(y)$  can be estimated by the relative frequency of  $y = 1$  and  $y = 0$  in  $\mathbf{Y}$ . For  $P(x_j|y)$ , we apply the Gaussian Naive  
 358 Bayes algorithm to solve the classification problem, where the posterior probability of features  $Prob(x_j|C)$  is assumed  
 359 to be Gaussian:

$$P(x_j|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_j - \mu_y)^2}{2\sigma_y^2}\right), \quad (27)$$

360 where  $\sigma_y$  and  $\mu_y$  are estimated by the data set. By substituting Equation (27) in Equation (26), the Gaussian NB  
 361 classifier reads:

$$\hat{y} = g^{\text{NB}}(\mathbf{x}) = \arg \max_y \text{Prob}(y) \prod_{j=1}^P \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_j - \mu_y)^2}{2\sigma_y^2}\right). \quad (28)$$

362 The detailed implementation of training an NB classifier based on the data set in this work and the case study  
 363 results are presented in Section 3.4.2.

### 364 2.6.3. Method evaluation

365 Lastly, we evaluate the performance of the combination of the two-step methods: the user selection and re-routing  
 366 method. We first select  $\alpha^*$  % from all the  $M$  users as targets, according to their *predicted* diffPMC with the help of  
 367 the  $f$  trained in the Section 2.5. We assume that the OD pairs and departure time of these users are known, and then  
 368 we compute the trajectory features under the UE condition for all the alternative paths of the targeted users in order  
 369 to obtain the table of feature attributes:  $\mathbf{X}_{\text{target}}$ . Therefore, we can predict the labels of all the alternative paths for  
 370 the targeted users, via  $Y' \approx g(\mathbf{X}_{\text{target}})$ . Finally, we re-run a UE simulation with the same demand profile and network,  
 371 after pre-defining the paths of selected users as the paths labeled by  $Y' = C_1$ . By comparing the TTT of (i) the whole  
 372 system in this test scenario with (ii) the TTT obtained in UE-ref and SO-ref simulations, we evaluate the efficiency of  
 373 the two-step methods proposed.

## 374 3. Numerical experiment and results

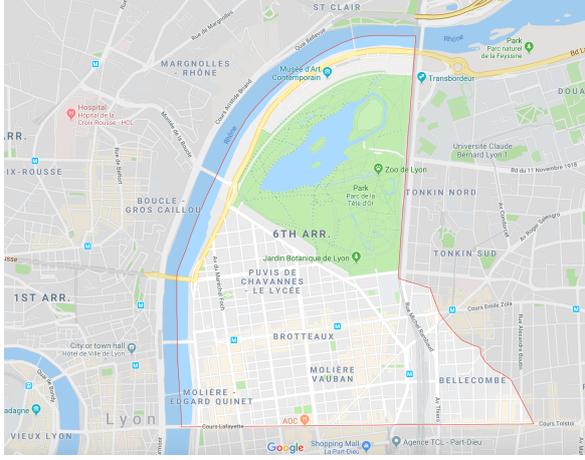
375 In the previous section, the general methodological framework is presented as well as the detailed formulations  
 376 of the LR, PR and NB models. Note that, in the methods proposed, we fix the fraction of selected users in priority  
 377 to assure a fair comparison of TTT reduction and evaluate the efficiency of the method. In practice, we can also  
 378 select users and define alternative paths according to a certain threshold of the computed diffPMC instead of fixing  
 379 a certain number of targeted users. In addition, the proposed methodology can be easily implemented, and we are  
 380 free to choose DTA models and/or simulators. As long as we are able to obtain full information on network-related  
 381 trajectory features, we can apply the methods proposed to any other DTA simulators/platforms. In this work, we apply  
 382 this framework to a numerical case study with the microscopic simulator Symuvia. In this section, we present the case  
 383 study scenarios and results.

384 In this study, we run two reference simulations via Symuvia based on the UE and SO conditions on the same  
 385 road network and with a given demand profile. The calculation process for the UE and SO solutions are detailed in  
 386 (Ameli et al., 2020a,b). We can compute their PMCs and network-related trajectory features listed in Section 2.2 and  
 387 Section 2.3. The data set of the trajectory features will then be used in the supervised learning process. Recall that  
 388 we use only link features under the UE condition when computing the traffic-state related features of both the UE  
 389 trajectories and SO trajectories. More precisely, we use link travel time, link marginal cost, and link average speed  
 390 under the UE condition to compute PTT, PMC, diffPMC, and dynamic BC of links. Therefore, the trajectory feature  
 391 differences for a pair of trajectories result only from whether the trajectory is chosen under a UE or under an SO  
 392 perspective. This helps us to unravel the characteristics of the SO structure under UE-based traffic conditions.

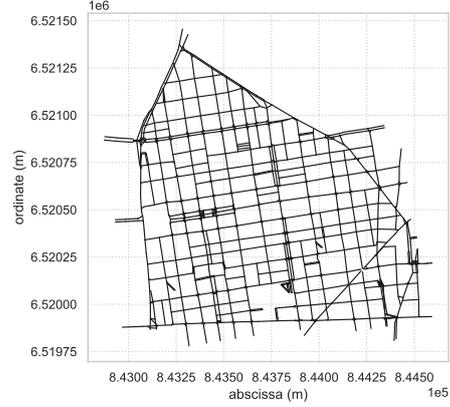
### 393 3.1. Case study description

394 In this study, we consider the road network of the 6<sup>th</sup> district of Lyon (*Lyon6*), France. The area of Lyon6 is  
 395 presented in Figure 2(a). Figure 2(b) shows the link-level representation of the main road network. There are a total  
 396 of 786 links, 457 nodes, and 710 OD pairs in the network. Figure 2(c) presents the time-dependent traffic demand  
 397 in the network. There are a total 13475 users ( $M = 13475$ ) traveling on the network during a simulation period of 4  
 398 hours (from 7:00 to 11:00 a.m.). Figure 2(c) shows that the peak hour is from 8:00 to 9:30 am.

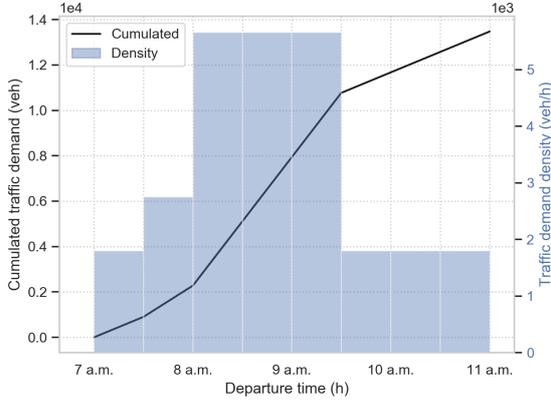
399 Figure 2(d) presents the macroscopic fundamental diagram (MFD) of the whole Lyon6 network in the UE and SO  
 400 simulations. It represents the spatial average of network traffic conditions (total travel distance vs. total experienced  
 401 travel time) through time. The MFD confirms that the traffic conditions in SO are better than in UE, as the total travel  
 402 time is shorter in SO than in UE for a given total travel distance.



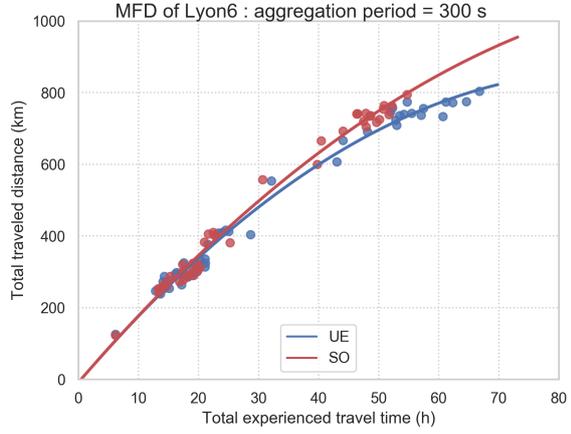
(a) Area of the 6<sup>th</sup> district of Lyon, France, © Google Maps 2021



(b) The modeled network of Lyon6 (excluding the park).



(c) Traffic demand



(d) MFDs in UE and SO reference simulations

Figure 2: Lyon6 network: test case characteristics in UE-ref and SO-ref simulations with Symuvia

### 3.2. Results from reference scenarios

Figure 3(a) presents the distribution of PTT on the network of both the reference UE simulation (UE-ref) and the SO simulation (SO-ref). We can see that the PTT decreases from UE-ref to SO-ref, and the network is less congested in the SO than in the UE-ref simulation. Figure 3(b) shows the distribution of PMC. We also find the PMCs of the SO trajectories are smaller than the UE trajectories. This observation is consistent with the assumption of the SO condition where users choose paths with the minimum PMC between its OD pair.

We assume that the TTT of the whole system reaches its minimum under the SO condition. Therefore, the TTT reduction from UE-ref to SO-ref denotes the improvement of the *optimal* performance that can be achieved when all users choose their paths under the SO perspective. The reference simulations show that from the UE to SO, the TTT of all users falls by  $6.56 \cdot 10^5$  seconds, representing 9.44 % of the TTT in the UE-ref simulation, as presented in Table 2. This reduction of TTT from UE-ref to SO-ref is then considered as the reference TTT reduction when evaluating the efficiency of targeting and re-routing strategies. More precisely, we use two criteria to evaluate the improvement of network performance: (i) the relative TTT reduction with respect to the UE-ref simulation (Equation (29)) and (ii) the relative TTT reduction with respect to the *optimal* TTT reduction from the reference scenarios (Equation (30)).

$$\Delta TTT_{\text{relative}} = 100 \% \times \frac{(TTT_{\text{UE-ref}} - TTT_{\text{test scenario}})}{TTT_{\text{UE-ref}}}. \quad (29)$$

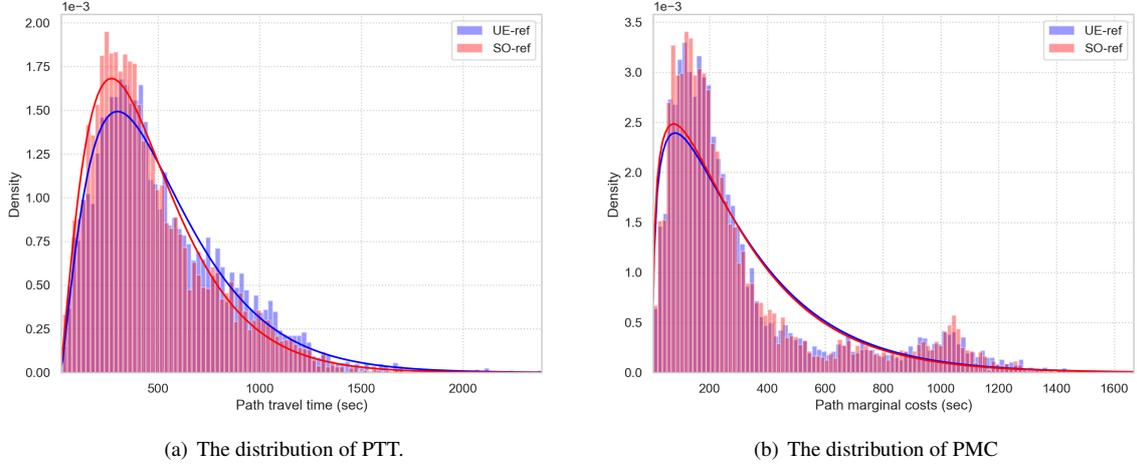


Figure 3: The distributions of path travel time (PTT) and path marginal cost (PMC)

$$\Delta TTT_{\text{relative w.r.t ref reduction}} = 100\% \times \frac{(TTT_{\text{UE-ref}} - TTT_{\text{test scenario}})}{TTT_{\text{UE-ref}} - TTT_{\text{SO-ref}}}. \quad (30)$$

417 Before the application of the supervised learning methods mentioned in Section 2 to our case study, we first deter-  
 418 mine the *optimal* fraction of users to be targeted, i.e.,  $\alpha^*$  %, based on a sensitivity analysis presented in Section 3.2.  
 419 Thereafter, when evaluating the proposed learning methods, we take  $\alpha^*$  % directly as the fraction of user selection.  
 420 The sensitivity analysis and results are presented in the following Section 3.2. Moreover, the reference test case pre-  
 421 sented in Section 3.2 can serve as the criterion for the method evaluation. The results from this reference case will  
 422 confirm the fact that we can achieve a significant TTT reduction by selecting a small share of the users who have  
 423 the largest diffPMC values and moving them to their SO paths. Therefore, when we evaluate the proposed learning  
 424 models, we compare the TTT reduction of the test cases with the TTT reduction obtained in the reference cases.

425 Then, after presenting the reference test cases, we apply the two-step learning framework to our case study in  
 426 Section 3.2 and Section 3.4.2. To prepare the data set, we compute trajectory features for all the users and prepare the  
 427 data set, including: (i) the training feature vectors composed of network-related trajectory features, (ii) the diffPMC  
 428 for all users as the target vector, and (iii) the labels of simulated paths as the target vector. To select the users, we use  
 429 the data set to train the LR and PR regression models. For re-routing, we train the NB classifier to predict whether a  
 430 given path is an SO path or not. Lastly, for the evaluation phase of both learning processes, we run DTA simulations  
 431 with the selected users moving onto their simulation/predicted SO paths in order to evaluate the TTT reduction of the  
 432 whole network.

433

### 434 3.3. The reference case: selection of targeted users according to actual path marginal cost

435 In this subsection, we aim to determine the *optimal* percentage  $\alpha^*$  of users to be selected. With the  $\alpha^*$  determined,  
 436 we then re-run a test scenario by selecting  $\alpha^*$  % users with the largest values of diffPMC, and move them to the SO  
 437 paths obtained in the SO-ref simulation. Then the TTT reduction of this test scenario in comparison to the TTT in the  
 438 UE-ref simulation is the criterion of the potential TTT reduction that can be achieved by rerouting  $\alpha^*$  % users.

439 The main steps for the sensitivity analysis to determine  $\alpha^*$  are the following. First, we compute the diffPMC for all  
 440 the  $M$  users based on the simulation results obtained in UE-ref and SO-ref, according to Equation (4) to Equation (7).  
 441 Then we rank all users according to their values of diffPMC, and select  $\alpha$  % of them with the largest values of  
 442 diffPMC as targeted users. Afterwards, we give these selected users predefined paths obtained in SO-ref and re-run a  
 443 DTA simulation where other users still travel under the UE condition. Finally, we evaluate the TTT reduction of the  
 444 system in the test scenario in comparison to UE-ref simulation.

Table 2: Simulation results of the UE-reference and SO- reference scenarios of the Lyon6 network. The bold numbers represent the results by re-routing an *optimal* fraction of users to SO paths.

Different scenarios	UE	SO	Results after rerouting different fraction of selected users ( $\alpha$ %) with large diffPMC value									
	Ref	Ref	1 %	2 %	3 %	4 %	5 %	10 %	15 %	<b>20 %</b>	25 %	30 to 70 %
TTT ( $10^6$ s)	6.94	6.29	6.93	6.73	6.72	6.62	6.68	6.66	6.47	<b>6.42</b>	6.52	6.39
$\Delta TTT$ ( $10^5$ s)	–	6.56	0.18	2.17	2.21	3.30	2.73	2.87	4.81	<b>5.28</b>	4.31	5.58
$\Delta TTT$ per user (s)	–	<b>48.7</b>	<b>1.3</b>	<b>16.1</b>	<b>16.4</b>	<b>24.5</b>	<b>20.3</b>	<b>21.3</b>	<b>35.7</b>	<b>39.2</b>	<b>32.0</b>	<b>41.4</b>
$\Delta TTT_{rel}$ (%)	–	9.44	0.26	3.13	3.18	4.74	3.93	4.14	6.92	<b>7.60</b>	6.20	8.02
$\Delta TTT_{rel, to ref}$ (%)	–	100	2.76	33.14	33.69	50.24	41.62	43.81	73.33	<b>80.56</b>	65.71	85.01

(TTT is the total travel time.  $\Delta TTT$  is the difference of TTT between the UE-ref simulation and the SO-ref simulation.  $\Delta TTT$ /user denotes the TTT reduction per user. The relative TTT reduction ( $\Delta TTT_{rel}$ ) is computed by Equation (29). The relative TTT with respect to the reference ( $\Delta TTT_{rel, to ref}$ ) is computed by Equation (30).)

445 To analyze the sensitivity of system performance improvement to the fraction of selected users, we select  $\alpha \in$   
446  $\{1, 2, 3, 4, 10, 15, 20, 25, 30, 35, 40, 45, 50, 60, 70\}$  and run the corresponding test scenarios mentioned in the above  
447 paragraph. Then we compute the relative TTT reductions according to the criteria mentioned in Section 3.2, in order  
448 to find the *optimal* fraction  $\alpha^*$  %. The results of these simulation scenarios are presented in Table 2. The results show  
449 that we can achieve a significant improvement in network performance by moving only a small share of the users to  
450 the paths considered to be *optimal* for the whole system, i.e., the paths assigned under the SO principle. Table 2 shows  
451 that re-routing 20 % the users to their SO-trajectories can lead to more than 80 % reduction in TTT concerning the  
452 reference case. **In average, the travel time saving is 39 s per user.** In other words, if we can efficiently target the 1/5  
453 of the users that have the highest values of diffPMC, we can achieve more than 3/4 of the TTT reduction of the reference  
454 case, where all the users travel on SO trajectories instead of traveling under the UE principle.

455 We also carry out benchmark (BM) scenarios by randomly selecting  $\alpha$  % of the users, with  $\alpha \in$   
456  $\{1, 2, 3, 4, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 60, 70\}$ . For each fraction  $\alpha$  %, we take 8 random samples of the users  
457 independently, and give them predefined paths as SO trajectories. Then we run DTA simulations under the UE con-  
458 dition to evaluate the TTT reduction in comparison to the UE-ref simulation. The results in Table 2 and the results  
459 of the BM scenarios are plotted in Figure 4. The results of the BM scenarios show that randomly moving a fraction  
460 of the users to their SO paths cannot ensure a reduction of system TTT until we choose a sufficient number of users  
461 (30 %). In addition, the random selection is not efficient at all. In fact, by randomly selecting 70 % of the users and  
462 re-routing them to SO paths, we can achieve on average less than 5 % of relative TTT reduction with respect to the  
463 TTT in the UE-ref simulation. However, by selecting only 20 % of the users who have the largest potential decrease  
464 of PMC, i.e., diffPMC, we can already obtain more than 7.5 % of TTT reduction from the UE-ref simulation.

465 Figure 4 shows that the diffPMC is a good criterion for selecting targeted users. By selecting users according to  
466 their PMCs reduction value from UE-ref to SO-ref and moving them onto SO paths, the system TTT reduction is  
467 significantly higher than in the cases where the users are selected randomly.

468 Based on these subsection results, we consider that  $\alpha^* = 20$  is the reference fraction for selecting targeted users.  
469 There are a total 13475 users in the network. The reference number of users that we select is then denoted as  $M_{target} =$   
470 2696. In addition, the TTT reduction obtained after moving 20 % of the users to their SO path, 7.60 %, is considered  
471 as a criterion of potential TTT reduction when evaluating the efficiency of our learning methods for user selection  
472 (Section 3.2) and user re-routing (Section 3.4.2).

#### 473 3.4. First step results: selection of targeted users by predicting diffPMC based on other trajectory features

475 The results in the reference case in Section 3.2 show that we can obtain more than 7.6 % of TTT reduction w.r.t  
476 UE case, by only selecting 20 % of users who have the largest values of diffPMC. Their diffPMC are computed based  
477 actual UE and SO paths obtained in the reference scenarios. However, we aim to re-route users without knowing  
478 their SO paths ahead, and the actual diffPMCs are not accessible. TIn this subsection, we aim to approximate users'  
479 diffPMC based on network-related trajectory features via a supervised learning process, and then show that we can  
480 also reduce the system TTT significantly, by re-routing 20 % of the users who have the largest values of *learned*  
481 diffPMC. More precisely, we aim to select  $M_{target}$  users according to the diffPMC *learned* by a regression model, and

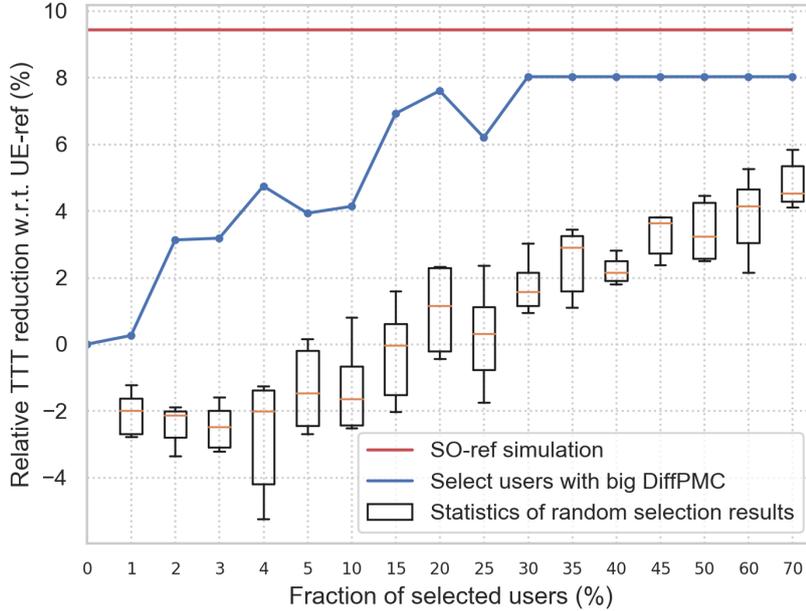


Figure 4: Relative TTT reduction with respect to TTT in UE-ref after selecting different fractions of users and re-routing them to SO paths. The reference relative TTT reduction from the UE-ref to the SO-ref simulation is 9.44 %. The black-line boxplots represent the statistics of TTT reduction after randomly moving users to their SO trajectories: (i) the orange lines represent median values, (ii) the boxes represent the interquartile ranges, and (iii) the upper and lower whiskers represent the 95<sup>th</sup> and 5<sup>th</sup> percentile.

482 move them on paths obtained from the SO-ref simulation in order to achieve a reduction of system TTT as close to that  
 483 obtained in Section 3.2 as possible. This step corresponds to the 1<sup>st</sup> step of our methodological framework, presented  
 484 as the left branch of the method flow chart (Figure 1) in Section 2.

#### 485 3.4.1. Data preparation

486 First, we prepare the data set for training the regression model. Note that we still use the results from the UE-ref  
 487 and SO-ref simulations as the data set. Among the 13,475 users on the network, we select  $M = 13127$  users' trajectory  
 488 feature data as the sample set. The eliminated users have not finished their trips in our simulation horizon. Each user  
 489  $i$  has a pair of trajectories,  $\mathcal{L}_i^{\text{UE}}$  and  $\mathcal{L}_i^{\text{SO}}$  obtained from the two simulations, respectively. We compute the normalized  
 490 PMC reduction of  $i$  by  $y_i = \text{diffPMC}_i = \widehat{\text{PMC}}_i^{\text{UE}} - \widehat{\text{PMC}}_i^{\text{SO}}$  (ref. Equation (4) to Equation (7)). Then we save the  
 491 computed diffPMCs in the target vectors to be approximated denoted by  $\mathbf{y} \in \mathbb{R}^M$ . For the same users, we also compute  
 492 other dimensionless trajectory features based on the results of the UE-ref simulation:  $\widehat{BC}_i$ ,  $G_i^{\text{BC}}$ ,  $\widehat{E}_i$ ,  $G_i^{\text{dynamicBC}}$ ,  $Cap_i$ ,  
 493  $\eta_i$ ,  $\mu_i$ . The values of these features define the feature vectors for user  $i$ ,  $\mathbf{x}^i = (x_1, \dots, x_P)^T$  where  $P = 7$ . Then we  
 494 define an  $M \times P$  data matrix  $\mathbf{X} = [\mathbf{x}^1 \dots \mathbf{x}^M]^T$ , where  $M = 13127$ , which is the number of the sample size. Therefore,  
 495 the objective is to use the known data set  $\mathbf{X}$  and  $\mathbf{y}$  in order to train a model:

$$f : \mathbb{R}^P \rightarrow \mathbb{R}$$

$$\mathbf{x} \rightarrow y = f(\mathbf{x}).$$

#### 496 3.4.2. Training and evaluation of the regression model

497 Here, we train two regression models (LR and PR),  $f_{\text{lin}}$  and  $f_{\text{poly}}$ , based on the simulated data set  $\mathbf{y}$  and  $\mathbf{X}$ . Then  
 498 we compute the approximate  $\mathbf{y}'_{\text{lin}} = f_{\text{lin}}(\mathbf{X})$  and  $\mathbf{y}'_{\text{poly}} = f_{\text{poly}}(\mathbf{X})$ . Therefore, we can sort the users by  $y'$  and select 20 %  
 499 of the users with the biggest predicted differential of PMC. Afterward, we give these users SO-ref trajectories and  
 500 re-run the DTA simulation, in order to evaluate the TTT reduction of the system.

Table 3: Scores of the LR and PR models for predicting the PMC reduction from UE to SO, based on all the seven network-related features. The correlation and RMSE are computed between (i)  $y$  (the actual PMC reduction obtained in UE-ref and SO-ref scenarios) and (ii)  $y'$  (the learned PMC reduction with trajectory features as attributes).

	Linear	Polynomial				Ref selection 20 % bigDiffPMC
		$d = 3$	$d = 4$	$d = 5$	$d = 6$	
Correlation	0.1489	0.4022	0.4812	0.5516	<b>0.5701</b>	-
RMSE	0.1775	0.1522	0.1395	0.1263	<b>0.1226</b>	-
$\Delta TTT_{\text{rel}}$ w.r.t UE-ref (in %)	-8.6	4.13	2.38	5.55	<b>6.50</b>	7.60
$\Delta TTT_{\text{rel}}$ w.r.t SO-ref reduction (in %)	-91.40	43.79	25.24	58.84	<b>68.42</b>	80.56

(The relative TTT reduction is computed by Equation (29). The relative TTT with respect to reference is computed by Equation (30). The column titled by 20 % bigDiffPMC in the table presents the case where we select 20 % of the users with the largest value of diffPMC selected and move to their SO paths. )

501 Table 3 presents the results of this learning phase. The correlation and the root-mean-square error (RMSE) are  
502 computed between (i) the actual differential of PMC from UE-ref to SO-ref, and (ii) the predicted differential of  
503 PMC learned from  $\mathbf{X}$ , based on the LR model and the PR model with different degrees. The results show that the  
504 6<sup>th</sup> order PR model is able to efficiently predict the PMC reduction. By selecting 20 % of users with a large value of  
505 predicted diffPMC, we can achieve a 6.5 % reduction of TTT in comparison to the UE-ref simulation. According to  
506 Equation (30), this TTT reduction obtained in the test case reaches 68% of the reference reduction, which is the case  
507 where all the users travel under the SO condition.

508

### 509 3.5. Second step results: re-routing strategies by predicting SO paths based on trajectory features

510 The results from Section 3.2 are obtained after re-routing the selected users to their *exact* SO paths resulting from  
511 the SO-ref simulation. The purpose of this section is to predict whether a path is an SO path or not, based on the  
512 known network-related trajectory features. The problem can be easily transformed into a classification task: to predict  
513 whether a path belongs to the class of SO paths or not, for all the possible paths between a given OD pair. The  
514 implementation and results presented in this subsection correspond to the 2<sup>nd</sup> step of our methodological framework,  
515 presented as the right branch of the method flow chart (Figure 1) in Section 2.

#### 516 3.5.1. Data preparation

517 Here we prepare the target vectors  $\mathbf{y}$  and matrix of attributes  $\mathbf{X}$ , following the steps presented in Section 2.6.1.  
518 Let  $M_{\text{all}}$  denote the feasible paths for all users between all OD pairs. The matrix of the target vectors for training the  
519 NB model is composed of the actual classes of the paths:  $\mathbf{Y}_{\text{all}} = [y_1, \dots, y_i, \dots, y_{M_{\text{all}}}]$  where  $y_i = 0$  if  $C(\mathcal{L}_i) = C_0$   
520 and  $y_i = 1$  if  $C(\mathcal{L}_i) = C_1$ . We also compute the network-related features of all these paths and construct the attribute  
521 matrix  $\mathbf{X}_{\text{all}} = [\mathbf{x}_1, \dots, \mathbf{x}_i, \dots, \mathbf{x}_{M_{\text{all}}}]$ , where  $\mathbf{x}_i \in \mathbb{R}^5$  which means that we consider only five features as attributes, since  
522 we have 2 pairs of correlated feature. In fact, in order to satisfy the conditional independence assumption of NB  
523 classifier, we first carried out a correlation analysis with  $\mathbf{X}$ . The correlations in Figure 5 show that there are two pairs  
524 of trajectory features that are not independent: (i)  $\widehat{BC}$  and  $G^{\text{BC}}$  (Figure 5(a)), (ii)  $\widehat{E}_i$  and  $G^{\text{dynamicBC}}$  (Figure 5(b)). If  
525 the correlation between two features is estimated as high, then this couple of features are dependent on each other  
526 and cannot both be taken into account when constructing the matrix of attributes in the NB training process. For  
527 example, Figure 5(a) shows that  $\widehat{BC}$  and  $G^{\text{BC}}$  are highly correlated. Figure 5(b) shows that  $\widehat{E}_i$  and  $G^{\text{dynamicBC}}$  are  
528 highly correlated. Therefore, we do not take the two Gini coefficients into account when constructing the attribute  
529 vector for training  $g^{\text{NB}}$  in this work.

530 In our case study, there are a total of  $M_{\text{all}} = 107385$  possible paths for all the users of all OD pairs in the Lyon6  
531 network. To ensure that we have sufficient data to train  $g^{\text{NB}}(\mathbf{x})$ , we exclude the paths between OD pairs where fewer  
532 than 30 vehicles are traveling between them. The final number of remaining paths used in the learning process is  
533  $M_{\text{dataset}} = 45848$ . Then, we separate the data set into two parts: the training set and testing set. Each of them consists of  
534 50 % of the trajectories randomly selected among all the  $M_{\text{dataset}}$  paths. The corresponding class variables and feature  
535 vectors of these trajectories are denoted by  $\mathbf{Y}_{\text{train}} = [y_1, \dots, y_{M_{\text{train}}}]$ ,  $\mathbf{X}_{\text{train}} = [\mathbf{x}_1, \dots, \mathbf{x}_{M_{\text{train}}}]$ ,  $\mathbf{Y}_{\text{test}} = [y_1, \dots, y_{M_{\text{test}}}]$ , and  
536  $\mathbf{X}_{\text{test}} = [\mathbf{x}_1, \dots, \mathbf{x}_{M_{\text{test}}}]$ .

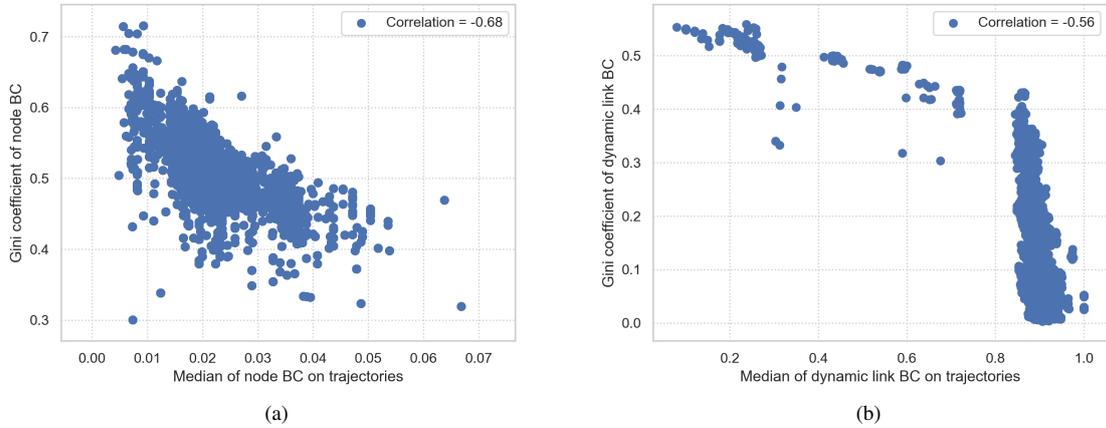


Figure 5: (a): The estimated correlation between the median node BC ( $\widehat{BC}$ ) and the Gini coefficient of BC ( $G^{BC}$ ); (b): The estimated correlation between median dynamic link BC ( $\widehat{E}_i$ ) and the Gini coefficient of dynamic link BC ( $G^{BC}$ ). (If the correlation between two features is estimated as high, then this couple of features are dependent on each other, and cannot both be taken into account when constructing the matrix of attributes in the NB training process.)

Table 4: Correction of class prediction by using the trained  $g^{NB}$ .

	$\widehat{\mathbf{Y}}_{\text{test}}$ v.s. $\mathbf{Y}_{\text{test}}$	$\widehat{\mathbf{Y}}_{\text{all}}$ v.s. $\mathbf{Y}_{\text{all}}$
Number of samples	22924	107385
% of correctly predicted classes	88.13 %	87.42 %

### 3.5.2. Training and evaluation of the Naive Bayes classifier

Here we use  $\mathbf{Y}_{\text{train}}$  estimate  $Prob(y)$ ,  $\sigma_y$  and  $\mu_y$  in Equation (28). With a known  $\mathbf{X}_{\text{train}}$ , we can train  $g^{NB}(\mathbf{x})$ . Note that we assume that the distribution of  $Y$  is the same for all the paths, so that  $Prob(y)$ ,  $\sigma_y$  and  $\mu_y$  remain the same for  $\mathbf{Y}_{\text{test}}$  and  $\mathbf{Y}_{\text{all}}$ . We can therefore estimate  $\widehat{\mathbf{Y}}_{\text{test}} \approx g^{NB}(\mathbf{X}_{\text{test}})$  and  $\widehat{\mathbf{Y}}_{\text{all}} \approx g^{NB}(\mathbf{X}_{\text{all}})$ . Then we compare the predicted class variables and the data set in order to evaluate the performance of the classifier. Table 4 presents the correction rate of the well-predicted class variables of  $\mathbf{Y}_{\text{test}}$  and  $\mathbf{Y}_{\text{all}}$ . If a simulated UE path is correctly labeled as  $C_0$  or a simulated SO path is correctly labeled as  $C_1$  by the trained NB classifier, then the conclusion about the prediction for this path is *correct*. The percentage of correctly predicted classes in Table 4 is computed by dividing the number of paths whose classes are correctly predicted by the total number of paths of the data set.

### 3.6. Results of combining two-step learning processes: selection and re-routing

Recall that in the reference case presented in Section 3.2, the TTT of the system is reduced 7.6% after re-routing 20% of the users with the largest value of diffPMC to their exact SO paths. In this subsection, we investigate whether system performance can still be significantly improved after selecting a small part of the users according to their *estimated* diffPMC, and rerouting them to their *predicted* SO paths. More precisely, we evaluate the efficiency of the two-step learning methodology in terms of TTT reduction in comparison to the UE-ref scenario, as shown in *part-III* of Figure 1 in Section 2.1. We select 20% of the users according to the diffPMC predicted by the PR model  $f_{\text{poly}}(\cdot)$  in Section 3.2. Instead of moving them into simulated SO paths, as presented in Section 3.2, we now reroute them on paths that are predicted to be SO paths according the NB classifier  $g^{NB}(\cdot)$  result in Section 3.5.2. Then we evaluate the TTT of this test scenario and compare it to the TTT in the UE-ref simulation.

First, for a selected user entering the network at time  $t$ , we identify their OD pair. Second, we search all the possible paths between this OD pair and compute the feature vector  $\mathbf{x}$  for each alternative path. Then, we rely on

Table 5: Rerouting strategies for targeted users

No.	Case	Action
1	None of the alternative paths is labeled as SO path	Do not move the user
2	One and only one path is predicted as SO path	Move the user onto this path
3	More than one alternative paths are labeled as SO paths	Randomly select <b>one path</b> from the predicted SO paths, and move the user on it

559  $g^{\text{NB}}(\mathbf{x})$  to predict whether a path is an SO-path or not. Since there may be more than one feasible path for a user, there  
 560 are three cases for the rerouting strategy, shown in Table 5.

561 Among all the 2696 selected users, 641 users have at least one predicted SO path among their alternative paths  
 562 of the same OD pair. These 641 users are finally targeted and moved to the predicted SO paths. There are some  
 563 users whose trajectories are not changed according to the NB classifier. This may be due to two reasons: (i) the  
 564 predicted SO paths are the same as the UE paths, and (ii) all the candidate paths of these users are predicted as  
 565 non-SO paths. Therefore, we pre-define the paths for these 641 users and run DTA simulations while the other users  
 566 continue traveling under the UE condition. Besides, in order to take into account the eventual estimation error caused  
 567 by the random selection of predicted SO paths in the case of the 3<sup>rd</sup> situation mentioned in Table 5, we carry out 12  
 568 independent DTA simulations. In each of these simulations, we randomly select only one predicted SO path for the  
 569 selected users who have at least two *predicted* SO paths.

570 The final simulation results are presented in Figure 6. The results show that by selecting 20% of the users with  
 571 the largest *predicted* diffPMC, and moving them onto the *predicted* SO paths, we can achieve on average 5.9% of  
 572 total travel time in comparison to the reference UE simulation. This performance is significantly better than the BM  
 573 scenarios, where we randomly select 20% of the users and move them on their SO paths obtained from SO-ref. We  
 574 also observe a small variation of TTT reduction in the 12 independent simulation scenarios. This infers that there  
 575 might be more than one alternative path that has close characteristics compared to the SO paths.

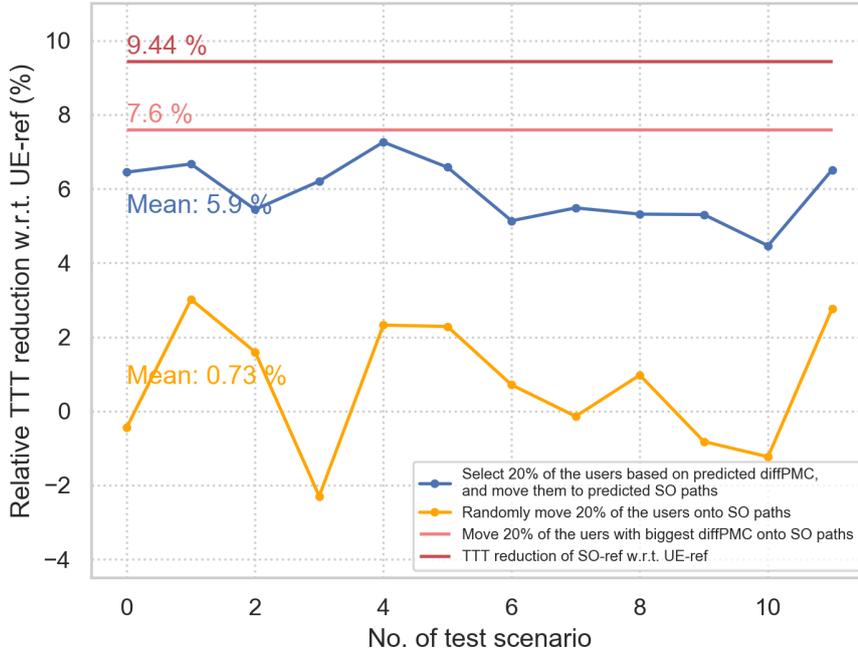


Figure 6: The relative TTT reduction with respect to UE-ref simulation for different test scenarios.

576 Furthermore, the final number of targeted users is 641, representing less than 5 % of the total users in the network.  
577 Nevertheless, by moving them onto the *predicted* SO paths, we achieve on average a TTT reduction of more than  
578 62 % in comparison to the reference case, where all the users choose their paths under the SO condition. In addition,  
579 among all the users, there are about 68 % of them have decreased their individual travel time in the re-routing scenario,  
580 with respect to the UE-ref scenario. The average travel time saving is about 146 s. For the rest 32 % of users who  
581 have increased their travel time after rerouting, the average increased travel time per user is about 200 s.<sup>9</sup>RC→LL:  
582 For the users that have been re-routed, there are 3.6 % of them who have increased their individual travel time after  
583 being re-routed, but the average increased travel time is 40 s. This is an acceptable increase of individual travel time  
584 and therefore there would be no difficulties for user compliance if we aim to apply the method into practical cases.  
585 Moreover, the results show that thanks to the trained PR and NB models, we are able to select and re-route a small  
586 share of network users to make a significant improvement to the performance of the whole network based only on  
587 their network-related trajectories.

#### 588 4. Conclusions and discussions

589 This work investigates network-related trajectory features to unravel the trips that contribute most to the system's  
590 under-performance. When such trips are identified, feature analysis also permits identifying the best alternatives in  
591 terms of routes to bring the system to its optimum.

592 Four features: node BC, dynamic link BC, the mean MFD capacity, and the mean distance between traffic lights,  
593 are identified as the key network-related features. The combination of these features defines the potential trips that  
594 contribute most to the system's under-performance. With these features as attributes, we build two learning models  
595 in order to (i) select the targeted trips that are critical for the system under-performance and (ii) re-route the users  
596 onto alternative paths in order to improve the whole system performance. We carry out a case study based on two  
597 reference DTA simulations for a middle-size network, from the perspectives of UE and SO. The microscopic traffic  
598 simulator (Symuvia) is used to obtain user trajectories and their corresponding network-related features. Note that the  
599 methodology proposed can be adapted to other DTA models and/or simulators because it uses the simulation results  
600 independently. In other words, as long as we are able to get full information on network-related trajectory features,  
601 we can apply the methods proposed to any other DTA simulators/platforms.

602 The application of our methodology to the real test case (Lyon6 network) shows that by selecting only 20 % of the  
603 trips and moving them onto *predicted optimal* paths; we can achieve an improvement of more than 62 % in network  
604 performance in comparison to the case in which all the users move from UE to SO paths. These results show that we  
605 can achieve a significant improvement in network performance by targeting a small share of users and moving them  
606 to *predicted* alternative paths, based only on their trajectory features and/or regular travel time information under the  
607 UE condition thanks to the learning models we propose.

608 This work opens various perspectives. The analyzed feature attributes can be obtained based on intrinsic features  
609 of the network (such as graphical characteristics, the spatial distribution of traffic lights), and/or a simple traffic  
610 monitoring system. Therefore, it would be possible to apply the proposed methodological framework to real-life  
611 practice if we have GPS trajectory data and network characteristics. Then we could re-route a small share of network  
612 users in the real-world via a centralized route planning system by providing users route guidance to reduce traffic  
613 congestion throughout the whole network. In particular, this real-world application would be promising if connected  
614 autonomous vehicles (CAVs) represent a large share of network users in the future since it would be easier and more  
615 efficient to re-route CAVs, which are already monitored by a centralized system.

616 In addition, we can consider more than one criterion to evaluate the improvement of system performance. In this  
617 work, we only consider minimizing the system's total travel time. In future works, we could also consider adding  
618 environmental criteria, such as minimizing the total air pollutant emissions of the whole system. Moreover, we can  
619 still improve the performance of model training, concerning the learning model of the user selection step, the 6<sup>th</sup> order  
620 of the polynomial regression model might be heavy if we deal with a data set with high-dimensional data. Other  
621 supervised learning models can also be investigated, such as Neural Network or Gaussian process regression with  
622 different kernels. For the classifier training, one direction can be to investigate better classification models to unravel  
623 SO path characteristics in order to avoid the cases (i) where there are than 1 predicted alternative paths or (ii) there is  
624 no predicted path at all, for a selected user. One possibility is to use an imbalance classification process because we  
625 have an imbalanced data set of feasible paths, where there are much more non-SO paths than SO paths.

## 626 5. Acknowledgements

627 This project has received funding from the European Research Council (ERC) in the framework of the European  
628 Union’s Horizon 2020 research and innovation program (Grant agreement No. 646592 – MAGnUM project).

## 629 References

- 630 Ameli, M., Lebacque, J.-P., Leclercq, L., 2020a. Cross-comparison of convergence algorithms to solve trip-based dynamic traffic assignment  
631 problems. *Computer-Aided Civil and Infrastructure Engineering* 35 (3), 219–240.
- 632 Ameli, M., Lebacque, J.-P., Leclercq, L., 2020b. Improving traffic network performance with road banning strategy: A simulation approach  
633 comparing user equilibrium and system optimum. *Simulation Modelling Practice and Theory* 99, 101995.
- 634 Ameli, M., Lebacque, J.-P., Leclercq, L., 2020c. Simulation-based dynamic traffic assignment: Meta-heuristic solution methods with parallel  
635 computing. *Computer-Aided Civil and Infrastructure Engineering* 35 (10), 1047–1062.
- 636 Beckmann, M., McGuire, C. B., Winsten, C. B., 1956. *Studies in the economics of transportation*. Tech. rep.
- 637 Bellocchi, L., Bellagha, M. H., Latora, V., Geroliminis, N., 2020. A measure of dynamic efficiency for multimodal interconnected urban systems.  
638 In: *Transportation Research Board 99th Annual Meeting (TRB2020)*. TRB.
- 639 Braess, D., Nagurny, A., Wakolbinger, T., 2005. On a paradox of traffic planning. *Transportation science* 39 (4), 446–450.
- 640 BUREAU, O. P. R., 1964. *Traffic assignment manual*. US Department of Commerce.
- 641 Cameron, A. C., Trivedi, P. K., 2005. *Linear Models*. Cambridge University Press, p. 65–115.
- 642 Chen, R., Leclercq, L., 2019a. Characterizing system optimum by trajectory data analysis. In: *22nd EURO Working Group on Transportation*  
643 *Meeting (EWGT 2019)*. EWGT.
- 644 Chen, R., Leclercq, L., Sep. 2019b. Unravelling System Optimum Structure by trajectory data analysis. In: *hEART 2019, 8th Symposium of the*  
645 *European Association for Research in Transportation*. Budapest, Hungary, p. 9p, hEART 2019, 8th Symposium of the European Association for  
646 *Research in Transportation*, Budapest, HONGRIE, 04-/09/2019 - 06/09/2019.  
647 URL <https://hal.archives-ouvertes.fr/hal-02436879>
- 648 Chickering, D. M., 1996. Learning bayesian networks is np-complete. In: *Learning from data*. Springer, pp. 121–130.
- 649 Chickering, D. M., Heckerman, D., Meek, C., 2004. Large-sample learning of bayesian networks is np-hard. *Journal of Machine Learning Research*  
650 5 (Oct), 1287–1330.
- 651 Çolak, S., Lima, A., González, M. C., 2016. Understanding congested travel in urban areas. *Nature communications* 7, 10793.
- 652 Freeman, L. C., 1977. A set of measures of centrality based on betweenness. *Sociometry*, 35–41.
- 653 Girvan, M., Newman, M. E., 2002. Community structure in social and biological networks. *Proceedings of the national academy of sciences*  
654 99 (12), 7821–7826.
- 655 Gonzalez, M. C., Hidalgo, C. A., Barabasi, A.-L., 2008. Understanding individual human mobility patterns. *nature* 453 (7196), 779.
- 656 Kotsiantis, S. B., Zaharakis, I., Pintelas, P., 2007. Supervised machine learning: A review of classification techniques. *Emerging artificial intelli-*  
657 *gence applications in computer engineering* 160 (1), 3–24.
- 658 Laval, J. A., Castrillón, F., 2015. Stochastic approximations for the macroscopic fundamental diagram of urban networks. *Transportation Research*  
659 *Procedia* 7, 615–630.
- 660 Leclercq, L., Laval, J. A., Chevallier, E., 2007. The lagrangian coordinates and what it means for first order traffic flow models. In: *Transportation*  
661 *and Traffic Theory 2007. Papers Selected for Presentation at ISTTT17* Engineering and Physical Sciences Research Council (Great Britain) Rees  
662 *Jeffreys Road Fund* Transport Research Foundation TMS Consultancy Ove Arup and Partners, Hong Kong Transportation Planning (International)  
663 PTV AG.
- 664 Leclercq, L., Verchier, A., Krug, J., Menendez, M., 2016. Investigating the performances of the method of successive averages for determin-  
665 ing dynamic user equilibrium and system optimum in manhattan networks. In: *DTA2016, 6th International Symposium on Dynamic Traffic*  
666 *Assignment*. pp. 1–p.
- 667 Lopez, C., Leclercq, L., Krishnakumari, P., Chiabaut, N., Lint, H., 2017. Revealing the day-to-day regularity of urban congestion patterns with 3d  
668 speed maps. *Scientific Reports* 7 (1), 14029.
- 669 Ma, X., Yu, H., Wang, Y., Wang, Y., 2015. Large-scale transportation network congestion evolution prediction using deep learning theory. *PloS one*  
670 10 (3), e0119044.
- 671 Mahmassani, H. S., Peeta, S., 1993. Network performance under system optimal and user equilibrium dynamic assignments: implications for ATIS.  
672 *Transportation Research Board*.
- 673 Pearl, J., 2014. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Elsevier.
- 674 Peeta, S., Mahmassani, H. S., 1995. System optimal and user equilibrium time-dependent traffic assignment in congested networks. *Annals of*  
675 *Operations Research* 60 (1), 81–113.
- 676 Roughgarden, T., Tardos, É., 2002. How bad is selfish routing? *Journal of the ACM (JACM)* 49 (2), 236–259.
- 677 Saeedmanesh, M., Geroliminis, N., 2016. Clustering of heterogeneous networks with directional flows based on “snake” similarities. *Trans-*  
678 *portation Research Part B: Methodological* 91, 250–269.
- 679 Seal, H. L., 1967. *Studies in the history of probability and statistics. xv the historical development of the gauss linear model*. *Biometrika* 54 (1-2),  
680 1–24.
- 681 Sheffi, Y., 1985. *Urban transportation networks*.
- 682 Smith, M. J., 1979. The existence, uniqueness and stability of traffic equilibria. *Transportation Research Part B: Methodological* 13 (4), 295–304.
- 683 Taheri, S., Yearwood, J., Mammadov, M., Seifollahi, S., 2014. Attribute weighted naive bayes classifier using a local optimization. *Neural Com-*  
684 *puting and Applications* 24 (5), 995–1002.
- 685 van Essen, M., Thomas, T., van Berkum, E., Chorus, C., 2016. From user equilibrium to system optimum: a literature review on the role of travel  
686 information, bounded rationality and non-selfish behaviour at the network and individual levels. *Transport reviews* 36 (4), 527–548.

- 687 Wang, H., Hernandez, J. M., Van Mieghem, P., 2008. Betweenness centrality in a weighted network. *Physical Review E* 77 (4), 046105.
- 688 Wang, P., Hunter, T., Bayen, A. M., Schechtner, K., González, M. C., 2012. Understanding road usage patterns in urban areas. *Scientific reports* 2,  
689 1001.
- 690 Wardrop, J. G., 1952. Road paper. some theoretical aspects of road traffic research. *Proceedings of the institution of civil engineers* 1 (3), 325–362.
- 691 Wu, X., Kumar, V., Quinlan, J. R., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G. J., Ng, A., Liu, B., Philip, S. Y., et al., 2008. Top 10 algorithms  
692 in data mining. *Knowledge and information systems* 14 (1), 1–37.
- 693 Yildirimoglu, M., Kahraman, O., 2018. Searching for empirical evidence on traffic equilibrium. *PloS one* 13 (5), e0196997.
- 694 Youn, H., Gastner, M. T., Jeong, H., 2008. Price of anarchy in transportation networks: efficiency and optimality control. *Physical review letters*  
695 101 (12), 128701.