Origin-Destination Flow and Traffic Parameter Estimation Based on Cellular Network Data and Vehicle Movement Historical Records

Mardešić, Nikola

UNIVERSITY OF ZAGREB
FACULTY OF TRANSPORT AND TRAFFIC SCIENCES

Nikola Mardešić

ORIGIN-DESTINATION FLOW AND TRAFFIC PARAMETER ESTIMATION BASED ON CELLULAR NETWORK DATA AND VEHICLE MOVEMENT HISTORICAL RECORDS

MASTER THESIS num 6284

Zagreb, 2021.

University of Zagreb
FACULTY OF TRANSPORT
AND TRAFFIC SCIENCES
Vukelićeva 4, HR-10000 Zagreb
GRADUATE STUDY

Graduate study: Intelligent Transportation Systems
Chair: Department of Intelligent Transportation Systems
Course: Data mining

GRADUATE THESIS ASSIGNMENT

Applicant: Nikola Mardešić
Matriculation number: 0135243785
Study programme: Intelligent Transportation Systems and Logistics

Assignment title: Origin-Destination Flow and Traffic Parameter Estimation Based on Cellular Network Data and Vehicle Movement Historical Records

Assignment title in Croatian: Određivanje Izvorišno-Odredišnih Protoka Vozila te Prometnih Parametara Zasnovano na Podacima Celularne Mreže i Povijesnih Zapisa Kretanja Vozila

Assignment description:

This thesis aims to determine the origin-destination flow of vehicles in the wider area of the city of Rijeka using two large data sets. The first data set is a collection of GPS logs of trips collected during four years on the observed area. The second data set consist of travel logs generated via the communication between mobile phones and cellular network base stations on the observed area. The flow of vehicles can be indirectly estimated by deviating the calculated speed from the free-flow speed. To achieve this, it is required to reconstruct the user trajectories on the observed road network using the two large data sets.

Supervising teacher: Chairperson of graduate thesis committee:

Administrator:

Sveučilište u Zagrebu
Fakultet prometnih znanosti

MASTER THESIS num 6284

ORIGIN-DESTINATION FLOW AND TRAFFIC PARAMETER ESTIMATION BASED ON CELLULAR NETWORK DATA AND VEHICLE MOVEMENT HISTORICAL RECORDS

ODREĐIVANJE IZVORIŠNO-ODREDIŠNIH PROTOKA VOZILA TE PROMETNIH PARAMETARA ZASNOVANO NA PODATCIMA CELULARNE MREŽE I POVIJESNIH ZAPISA KRETANJA VOZILA

Mentor: prof. dr. sc. Tonči Carić
Komentor: Leo Tišljarić, mag. ing. traff.

Student: Nikola Mardešić
JMBAG: 0135243785

Zagreb, September 2021.

Origin-Destination flow and traffic parameter estimation based on cellular network data and vehicle movement historical records

Abstract:

With suitable algorithms, it is possible to ascertain a certain spatiotemporal logic for a given area by combining cellular and vehicular data. This thesis proposes a set of methods that aim to estimate trajectories of sparse cellular data by reconstructing a road network using the Floating Car Data (FCD). Thenceforth, generating a pool of alternative paths of the reconstructed network with a shortest-path algorithm and conclusively computing an algorithm that allocates the user an alternative route with the highest spatial and temporal similarity to the observed ground-truth origin-destination pair (user) trajectory. With the data on the user trajectories used through a transportation network, this thesis aims to develop a model which calculates the networks traffic flow estimates.

Keywords: NoSQL, C#, large data sets, traffic data analysis, trajectory estimation, traffic flow estimation

Određivanje izvorišno-odredišnih protoka vozila te prometnih parametara zasnovano na podatcima celularne mreže i povijesnih zapisa kretanja vozila

Sažetak:

Pomoću prikladnih algoritama, moguće je utvrditi određenu prostorno-vremensku zakonitost prometnog toka obuhvaćenog područja kombiniranjem podataka o celularnoj mreži i povijesnih zapisa kretanja vozila. U ovom se radu predlaže skup metoda kojima je cilj procijeniti putanje kretanja korisnika celularne mreže rekonstrukcijom cestovne mreže pomoću plutajućih podataka o vozilu (FCD). Nadalje, nad rekonstruiranom cestovnom mrežom se izvršava algoritam najkraćeg puta te se generira n alternativnih trajektorija. Konačni korak procjene trajektorije predstavlja izvršavanje algoritma koji korisniku dodjeljuje alternativni put s najvećom prostornom i vremenskom sličnošću s promatranim parom ishodišno-odredišne putanje korisnika. Pomoću podataka o korisničkim putanjama korištenim kroz prometnu mrežu, ovaj rad teži razviti model koji izračunava procjene protoka prometa na mreži.

Ključne riječi: NoSQL, C#, veliki skupovi podataka, analiza prometnih podataka, procjena trajektorija kretanja korisnika, procjena prometnog toka

CONTENTS

1. Introduction

In this thesis, cellular and vehicular data are analysed to estimate Origin-Destination (OD) trajectories with the purpose of developing a traffic flow model. Accordingly, this thesis proposes a set of methods that aim to estimate the trajectories used through a transportation network from a sparse cellular network data set and a vehicle historical movement data set.

Although traditional traffic detection systems such as traffic counters produce vast quantities of real-time and historical data, their coverage is limited to the location of the monitoring infrastructure. Hence, they are not necessarily suitable for higher resolution analysis. An alternative to the costly and spatially restricted traditional detection methods can be found in cellular networks. By analysing Call Detail Records (CDRs), events that occur when users initiate or receive phone activities, mobile phone operators collect large quantities of data containing information about the time and location of the antennas users connected to. The main potential of the signalling events is the opportunity to use them without any additional measurement infrastructure. The spatial and temporal resolution of such data is, however, typically low [1]. Nevertheless, the data contains information on the activities of a large pool of users in space and time and can be used as a lucrative source for the traffic analysis of the entire road network.

There is a growing body of research utilizing this data for traffic flow and parameter analysis 2. With suitable algorithms, it is possible to ascertain a certain spatio-temporal logic for a given area based on these data. Alongside this, one can employ traditional traffic science data sources, such as historical records, to correlate estimates and "ground truth". Developing a model which calculates flow estimates sufficiently concurrent with historical records is one of the guiding interests behind this research.

1.1. Thesis objectives

The increasing volume of large, open traffic data sets enables researchers to gain new insights into the traffic network. Although characterized as unstructured, open data sets contain a wealth of information. The first step of the Thesis is to process vehicle movement historical data records to determine the speed profile for road segments in the city of Rijeka and its neighbouring regions. During the pre-processing, it is necessary to conduct a geo-allocation of the

generated road segments into appropriate pre-defined cellular *OD* network cells. Furthermore, this Thesis aims to perform a data-driven network trajectory estimation by fusing the processed cellular *OD* and vehicular *FCD* data. With the data on the user trajectories used through a transportation network, this thesis aims to develop a model which calculates the networks traffic flow estimates.

1.2. Thesis structure

Following the introduction, Chapter 2 provides a brief review of the literature related to trajectory estimation based on cellular data. Chapter 3 describes the fundamentals of graph theory with an emphasis on the shortest path method used in this study, Dijkstra's algorithm. Moreover, a brief overview of the traffic flow and its equations is presented. Proceeding, Chapter 4 describes the principal data sets utilized in this Thesis and their pre-processing steps required to calculate trajectory and flow estimates. With the introduction of the data sets, Chapter 4 describes the methods developed to calculate trajectory and flow estimates. Chapter 5 presents and discusses the outcomes generated by the developed methods. Finally, Chapter 6 summarizes the methodology of the Thesis and provides a review of the concluded activities and their results.

2. State of the art

Per Hoteit et al. [2], mobile data-based research facilitated a great leap forward in different perspectives of human characteristic research fields, such as virus spreading [3], urban and transport planning [4], network design [5], population estimating [6], community detection [7], carbon footprint [8], etc. Chen et al. [9] inferred that mobile phone data uncovered and enhanced the understanding of major patterns in human movements. Such as home-work commuting [10], which aid the policymakers in optimising the commuting routes and has given industries such as the retail industry a big insight into potential shop areas.

Moreover, further applications of mobile subscription data are observed in mobility behaviour estimation in ITS applications. To be able to estimate trajectories, Schlaich et al. [11] used sequences of Land Area Update (LAUs) to derive trajectories for mobile users. Thus, comparing a series of LAUs to a set of pre-generated routes between an inferred start and end position of the observed user. The route showing the highest similarity concerning the sequence of LAU events was chosen as the trajectory describing the corresponding user's mobility. To infer a most-likely travelled path of an observed cell-user, Leontiadis et al. [12] proposes to calculate the shortest path using lowered link costs for the links inside cells that the user connected to during the trip to make the route more likely to pass through these cells. Another similar approach presented by Wu et al. [13] is to fetch a predefined set of alternative routes for each ground-truth OD-pair and select the route that has the highest spatial similarity with the cells that a user connected to during the trip. To infer the trajectory for the road network traffic, Fillekes [1] utilises map-matching techniques custom to GPS data.

3. Trajectory and traffic flow estimation theory

3.1. Trajectory estimation

The following section describes the fundamentals of graph theory and Dijkstra's algorithm, which serves as the primary instrument for generating a pool of alternative trajectories in this Thesis.

3.1.1. Graph theory

Per Wilson [14], graph *G* consists of a non-empty finite set *V(G)* of elements called vertices (or nodes), and a finite set *E(G)* of distinct unordered pairs of (not necessarily distinct) elements of *V(G)* called edges. *V(G)* is called the vertex set and *E(G)* the edge set of *G*. An edge *{v,w}* is said to join the vertices *v* and *w*, and is usually abbreviated to *vw*. Graphs can be undirected and directed (*digraphs*). In undirected graphs, there is no difference in the orientation of the connection that connects some two nodes of the graph, that is, the volume of the connection can be used in both directions, while the directional connection defines the direction of the connection between two nodes of the graph. The graph is defined as an ordered triple: [15]

$$G \ = \ (V, E, W) \tag{3.1}$$

where *V* is a set of all nodes of the graph:

$$V = \{v_0, ..., v_n\} \tag{3.2}$$

E is a set of all connections (edges) in the graph:

$$E = \{(v_i, v_j) | (v_i, v_j) \in V^2, i \neq j\} \tag{3.3}$$

and *W* is a set of weights defined over a set of edges *E*:

$$W = \{w_{i,j} | (v_i, v_j) \in E\} \tag{3.4}$$

where $w_{i,j} = f_w((v_i, v_j))$. Function $f_w : E(v_i, v_j)) \rightarrow W(i, j)$, assigns a weight $w_{i,j}$ to the connection (v_i, v_j). In undirected graphs, connections *E* are a set of disordered pair nodes, while in directed graphs, node pairs are arranged. A graph in which different connections between the same nodes are allowed is called a *multigraph* or *pseudograph*. The row of a graph *r(G)* is defined as the number of nodes in the graph *G*: [15]

$$r(G) = |V|, V \in G \tag{3.5}$$

Size of graph $l(G)$ is defined as a number of connections in G:

$$l(G) = |E|, E \in G \tag{3.6}$$

Figure 3.1 illustrates an example of a directed graph. The graph is defined as:

$$V = \{A, B, C, D\} \tag{3.7}$$

$$E = \{(A, B), (B, C), (C, D), (D, A), (B, A), (A, D), (D, C), (C, B)\} \tag{3.8}$$

$$W = \{w_{A,B} = 5, w_{B,C} = 15, w_{C,D} = 5, w_{D,A} = 12, w_{B,A} = 15, w_{A,D} = 10, w_{D,C} = 5, w_{C,B} = 5\} \tag{3.9}$$

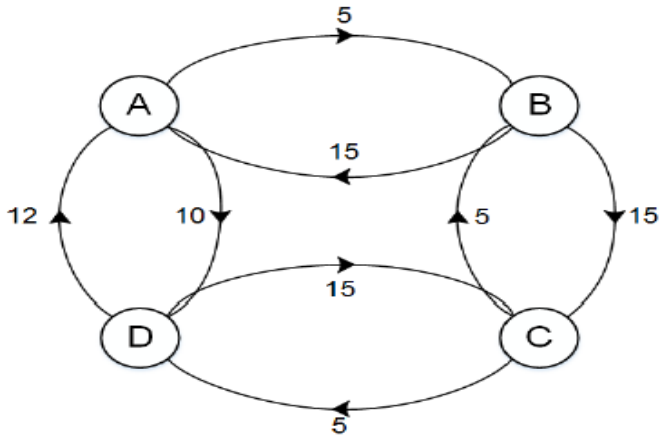


Figure 3.1: Directed graph example [15]

3.1.2. Path in the graph

In graph theory, the path in a graph is a finite or infinite sequence of links that connect a set of nodes. The path can be simple if the nodes in it are not visited more than once or complex if the nodes are re-visited. In a directed graph, a directed path is a set of links that connect the nodes of the graph with the restriction that all links are directed [16]. The path $p(v_1; v_n)$ represents a sequence of nodes and edges in the graph ranging from v_1 to v_n. If there are multiple connections between two adjacent nodes of the graph, the path is defined by a series of connections: [15]

$$p(v_1, v_n) = ((v_1, v_2), (v_2, v_3), ..., (v_{n-1}, v_n)) \in E \times E \times ... \times E \tag{3.10}$$

in such a manner that node v_i is adjacent to node v_{i+1} for $1 \leq i < n$.

The distance Y between two nodes in the graph is defined as the length of the shortest path between them, and if such a path does not exist then the distance is infinite [15].

3.1.3. Shortest path problem

In graph theory, the shortest path problem can be defined as the difficulty of finding the shortest connected route between two points or vertices, $i.e.$ the minimum sum of the edges on all possible connected routes between the two vertices. In terms of a traffic network, the nodes of the graph (network) represent intersections while edges represent the roads in which each edge has a weight denoting the length of the road. The weight of the path in graph G, travelling from node v_1 to node v_n is defined as: [15]

$$\sum_{i=1}^{n-1} f_w((v_i, v_{i+1})) \tag{3.11}$$

3.1.4. Dijkstra's algorithm

Dijkstra's algorithm determines the shortest path tree from a single source node, building a set of nodes that have a minimum distance from the source. The complexity of the algorithm is $(O(|V|^2)$, where $|V|$ describes the number of nodes in the graph. Dijkstra's algorithm is used to find the shortest distances path or the least cost path, depending on what the weights of the edges of the graph are set. Dijkstra's algorithm can be used to find the shortest path between cities. The nodes of the graph represent the cities, the connection weights of the graph represent the distance between pairs of cities connected by a road. As a result, the shortest path algorithm is very often used in network systems for determining paths for vehicles. A description of the algorithm is given below, and the $Pseudocode$ is shown in the Algorithm 1: [15]

– The initial node is marked with v_i. Distance Y is the distance from the start node v_i to a certain end node v_e.

1) Each node in the graph is assigned a specific initial distance; the initial node v_i distance is equal to 0, all other nodes in the graph have an infinite distance.

2) All nodes of the graph are marked as unvisited. The initial node is set as the current one. A list of unvisited nodes is made, which initially consists of all of the nodes in the graph.

3) For the current node, the distances to all neighbouring unvisited nodes are calculated, that is, to each node with which it has a valid connection. Furthermore, the calculated distance is compared to the currently assigned distance and the smaller value is assigned to the path. $E.G.$; if the current node A has a distance of 6 and the edge that connects it to the neighbouring node B has a distance of 2, then the distance to B (through A) is equal to 8 (6+2=8). If B had a previous distance that is greater than 8, the value gets overwritten to 8.

4) When the algorithm iterates through all the neighbours of the current node, the current node is marked as visited and deleted from the list of unvisited nodes. A visited node will never be visited again.

5) If the minimum distance between nodes in the unvisited list is infinite (occurs when there is no valid path from the initial node to the remaining unvisited nodes) or if there are no more nodes in the unvisited list, the algorithm stops. On the opposite, the algorithm chooses the next valid unvisited node and returns to step 3.

Table 3.1 Dijkstra's algorithm variable description

Variable	Description
G	System graph
v_i	Node of graph G
Q	List of unvisited nodes of graph G
u	Graph G node with the lowest weight
alt	Total path difficulty

Algoritam 1 Dijkstra's algorithm [15]

Input: v_i, G

Output: Q

1: $dist[v_i] \leftarrow 0$ # *Distance from v_i to v_i*
2: **for** $node\ v$ in G **do** # *Initialisation*
3: **if** $v \neq v_i$ **then**
4: $dist[v] \leftarrow infinity$ # *Unknown distance from v_i to v*
5: $previous[v] \leftarrow undefined$ # *Prior node in the path from v_i*
6: **end if**
7: $add\ v\ to\ Q$ # *All nodes initially in Q*
8: **end for**
9: **while** Q is not $empty$ **do** # *Main loop*
10: $u \leftarrow node\ in\ Q\ with\ min\ dist[u]$ # *v_i in the first step*
11: $remove\ u\ from\ Q$
12: **for** $neighbour\ v$ of u **do** # *With v still not removed from Q*
13: $alt \leftarrow dist[u] + length(u, v)$
14: **if** $alt < dist[v]$ **then** # *A shorter path towards node v is found*
15: $dist[v] \leftarrow alt$
16: $previous[v] \leftarrow u$
17: **end if**
18: **end for**
19: **end while**

3.2. Traffic flow and parameters

Traffic flow is the simultaneous movement of several vehicles on the road in a certain order. To be able to describe a traffic flow, it is necessary to define indicators. These indicators, in traffic flow theory, are called the basic parameters of the traffic flow or the basic magnitudes of the traffic flow. The main indicators for describing traffic flows are: [17]

- traffic flow, q [veh/s]
- traffic flow density, g [veh/km]
- traffic flow speed, v [m/s]
- vehicle travel time in the flow, t [s]

3.2.1. Traffic flow

The term traffic flow indicates the number of vehicles that pass through the cross-section of the observed road in a unit of time in one direction for one-way roads or in both directions for two-way roads. Depending on the mode of observation, there are two types of flows: [17]

1) Describes the flow of the vehicle concerning the cross-section of the road in a unit of time and is shown by the expression 3.12. (Figure 3.2)

$$q = gV \tag{3.12}$$

where:

- q [veh/h] - vehicle flow
- g [veh/km] - vehicle density
- V [km/h] - vehicle speed

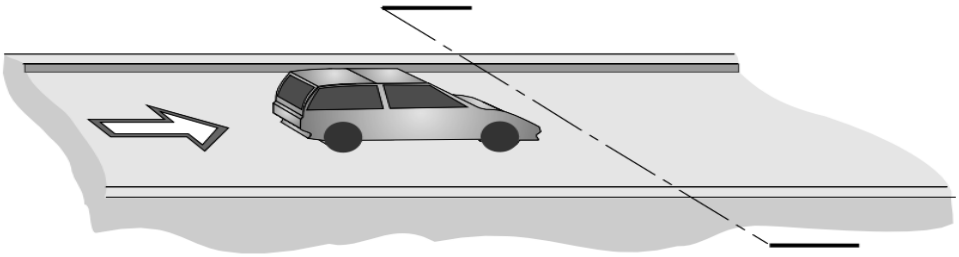


Figure 3.2: Cross-section traffic flow [17]

2) Traffic flow on an observed area represents the arithmetic mean of the flow on an n-section observed area, where $n \to \infty$. (Figure 3.3)

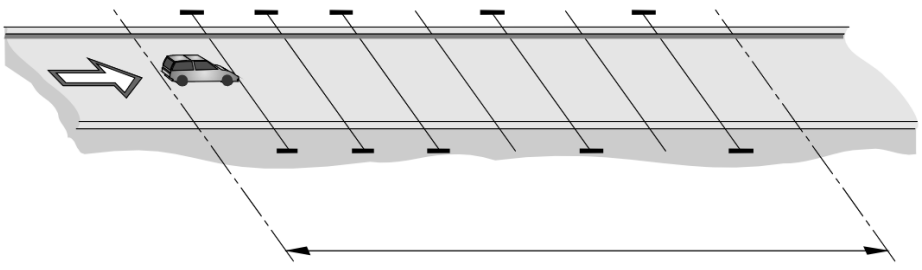


Figure 3.3: Road section traffic flow [17]

3.2.2. Traffic flow density

Traffic flow density is expressed as the number of vehicles located on the observed part of the road. It is especially expressed for one direction, per traffic lane, *i.e.* for both directions for two-way roads. Given the time period of observation, the density can be expressed as: [17]

1) The number of vehicles in the road at the time of observation which is calculated according to the expression 3.13.

$$g = \frac{N}{s} \tag{3.13}$$

   where:

   - g [veh/km] - vehicle density
   - N [veh] - number of vehicles on the observed area at time of measure
   - s [km] - length of the observed area

2) The arithmetic mean of a sequence of density measures on the observed area.

3.2.3. Traffic flow speed

The traffic flow speed indicates the average speed of all vehicles moving in the observed area of the road infrastructure. Depending on the method of measurement, the following differ: [17]

1) The mean time traffic velocity
   - The mean time velocity is spatially related to the measurement area (road cross-section), and temporally to the observation period. It represents the arithmetic mean of all measured speeds in the traffic flow of vehicles that pass through the observed section on the road in a defined period. The mean time velocity is defined by the expression 3.14:

$$\overline{v_t} = \frac{1}{N} \sum_{i=1}^{N} v_i \tag{3.14}$$

2) Mean spatial traffic flow velocity
   - The mean spatial velocity is spatially related to a section of the road and temporally to a moment in time. It represents the arithmetic mean of the speeds of all vehicles on a certain section of the road at a certain point in time. The mean spatial velocity is defined by the expression 3.15:

$$\overline{v_s} = \frac{\sum_{i=1}^{n} v_i}{n} \tag{3.15}$$

Depending on the conditions on the road, speeds v_t and v_s assume the following names according to traffic flow theory: [17]

- $Free\ flow\ speed - FFS$ - describes the speed of the vehicle in ideal conditions without interaction with other vehicles. Most often it represents the maximum theoretical speed of movement on the road.
- $Normal\ flow\ speed$ - is divided into stable, semi-stable and unstable traffic flow speed. Depends on the current road conditions.
- $Saturated\ traffic\ flow$ - the speed that vehicles achieve when the traffic flow is at the maximum capacity of the road. Describes traffic in conditions when vehicles interactively affect the speed of the flow.
- $Forced\ traffic\ flow$ - describes the speed of the vehicle during a stop-start drive. The speed of all vehicles is approximately equal and oscillates in the interval $< 0, \overline{v_{zt}}]$.

4. Research methodology

4.1. Data and pre-processing

4.1.1. Overview

The following section describes the road network data, the cellular OD data and the pre-processing steps applied to each data source.

The initial *FCD* data of the road network and the cellular *OD* CDR (Call Detail Record) data are described in sections 4.1.2 and 4.1.3. The pre-processing activities conducted upon the *FCD* data have the aim of calculating the average speed of each link (road segment) for each predefined time interval. *FCD* pre-processing activities are described in section 4.1.4. Enriching the *FCD* data set by assigning road segments into appropriate cells is described in section 4.1.5.

4.1.2. FCD data

Vehicle historical movement data used in this thesis are GPS (Global Positioning System) traces processed in the scope of a Faculty of Transport and Traffic Sciences project, $SORDITO$[1]. The raw GPS logs were collected by the company MIREO Inc.. During a period spanning from August 2009 to October 2014, MIREO Inc. tracked the movement of approximately 4200 vehicles. The tracked vehicle fleet is versatile and consists mostly of delivery vehicles (vans, caddies, small trucks) and taxi cars. Within sixteen months, over seven billion GPS logs across Croatia and its neighbouring regions were obtained using navigation devices installed in the vehicles. Figure 4.4a shows the speed profile of the road network processed in the scope of SORDITO. The color ramp indicates the speed on the road spanning from red (speed $\leq$ 5 km/h) to green (speed $\geq$ 100 km/h). While figure 4.1a illustrates the process of logging FCD data of a route [18], [19].

[1]Project SORDITO - System for Route Optimization in Dynamic Transport Environment RC.2.2.08-0022, funded by the European Union from the European Regional Development Fund

(a) Example of a tracked route **(b)** Visualisation of speed profiles of the network

Figure 4.1: Collected data across Croatian and its neighbouring regions, [18]

Accumulated GPS logs are structured in a CSV (Comma Separated Values) format. Each line of the CSV describes a road segment in the wider area of the city of Rijeka used in this thesis. One link is a part of the road bounded by two intersections. Each link is defined with a unique identification key (ID), link polarity, length in meters, static speed, speed limit and two geographical coordinates (start and endpoint). The format of the record is as follows:

$$LinkID;Way;Length;MSpeed;Limit;x_1;y_1;x_2;y_2$$

Table 4.1 describes each attribute of a link.

Table 4.1 Link attribute description

Attribute	Value	Description
LinkID	214695	Value of the unique road segment ID in the digital map
Way	2	Indicates a one-way/two-way link and its direction: 0 - two-way link; 1 - one-way link whose direction of movement is from point $P_1(x_1, y_1)$ to $P_2(x_2, y_2)$; 2 - one-way link whose direction of movement is from point $P_2(x_2, y_2)$ to $P_1(x_1, y_1)$; 3 - closed road
Len	340	Link length [m]
MSpeed	38	Static speed assigned by MIREO [km/h]
Limit	60	Link speed limit [km/h]
x_1	15.9521055221558	Longitude of the point $P_1(x_1, y_1)$ in decimal degrees
y_1	45.7848684748881	Latitude of the point $P_1(x_1, y_1)$ in decimal degrees
x_2	15.95250248909	Longitude of the point $P_2(x_2, y_2)$ in decimal degrees
y_2	45.781823298653	Latitude of the point $P_2(x_2, y_2)$ in decimal degrees

For each road segment, an additional file exists and contains a set of calculated speeds generated when the vehicle traversed the road segment. In addition to the derived speeds, the file contains the $Coordinate\ Universal\ Time$ (UTC), which represents the time when the log was generated. The format of the record is as follows:

$$UTC;v_1;v_2;v_3;v_4$$

Figure 4.2 visualizes the format of the link speed data, while Table 4.2 describes the format of the speed log.

```
1  UTC;v1;v2;v3;v4
2  1406117262;39.3;35.5;35.5;35.5
3  1402476019;33.3;4.1;4.1;4.1
4  1402914779;28.0;3.9;3.9;3.9
5  1385801758;39.5;39.9;39.9;39.9
6  1367582102;33.5;34.3;34.3;34.3
7  1368191304;19.3;6.4;6.4;6.4
8  1369646157;33.5;36.3;36.3;36.3
9  1375447951;36.5;3.5;3.5;3.5
10 1376303234;24.0;28.2;28.2;28.2
11 1391072039;32.0;3.9;3.9;3.9
12 1370336328;30.5;31.4;31.4;31.4
13 1399892728;17.0;15.0;15.0;15.0
14 1380009296;28.5;32.0;32.0;32.0
15 1396951663;17.5;25.1;25.1;25.1
16 1375434254;28.0;21.7;21.7;21.7
17 1324628910;35.5;33.2;33.2;33.2
18 1370604006;32.0;3.8;3.8;3.8
19 1359455801;27.0;14.1;14.1;14.1
20 1354002031;0.0;8.3;8.3;8.3
21 1404395714;40.0;3.8;3.8;3.8
22 1405306203;63.0;62.9;62.9;62.9
23 1405464093;41.0;6.3;6.3;6.3
24 1405796636;14.5;1.6;1.6;1.6
25 1405799211;35.0;36.7;36.7;36.7
```

Figure 4.2: CSV format linka

Table 4.2 Link speed description

Attribute	Data type	Description
UTC	int (4 B)	UTC GPS log time (seconds) - the exact date can be obtained from UTC
v_1	double (8 B)	Average GPS speed of all records of one recorded vehicle traversing the link
v_2	double (8 B	Average uniform speed per segment calculated based on the distance travelled between adjacent GPS records of the movement of one vehicle on the link
v_3	double (8 B)	Uniform speed calculated on the basis of the distance travelled between the first and last movement record of a vehicle and the elapsed time
v_4	double (8 B)	Uniform speed calculated on the basis of the air distance travelled between the first and last GPS record of the movement of one vehicle and the elapsed time

SORDITO project results inferred that it is best to use speed v_2 in further calculations, as it was observed that in most cases speeds v_3 and v_4 had the same value as v_2 [18]. For this reason, the speed v_2 will be used in this thesis.

4.1.3. Cellular OD data

The cellular network data used in this thesis is an Origin-Destination (OD) matrix provided by the company Ericsson Nikola Tesla to the Faculty of Transport and Traffic Sciences. The CDR data was collected using cellular base stations, where a CDR is saved for every event (phone call or text message) that a user makes [20]. It is important to note that in this Thesis, the cellular data consists of pre-processed *CDR* data in form of a *OD* matrix. The data set consists of 505,992 logs divided into 42 cells across the city of Rijeka and its surrounding regions and resemble the user activity of a typical working day. It is important to note that in contrast to the FCD data set users, the nature of the tracked users in the cellular data set is unknown. Moreover, the user ground truth trajectories consist only of the start cell and end cell of the user's path, whereas the typical format of a user trajectory would consist of all the cells that the user used while traversing from point A to point B. Figure 4.3 visualizes the cell distribution and coverage area of the observed data in city Rijeka and its surrounding regions.

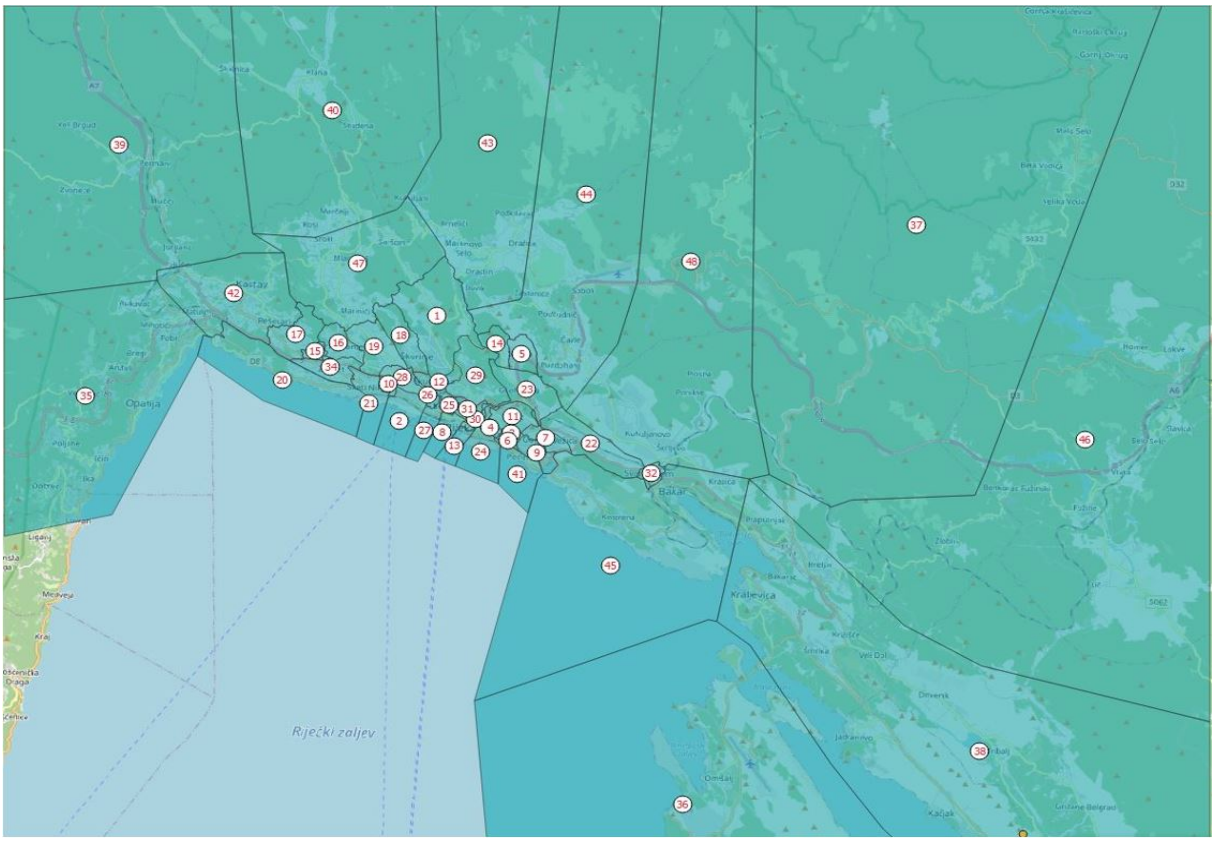


Figure 4.3: Cellular data cell distribution and coverage (turquoise polygons)in Rijeka and adjacent region

Accumulated cellular logs are structured in a CSV format. Each line of the CSV describes an individual user log. Each cellular OD log contains 9 numerical attributes described in Table 4.3.

Table 4.3 Cellular OD attributes and description

Attribute	SI unit	Description
Interval		Log start time interval
Duration	s	Log duration
Air Distance	m	Log traversed air distance
Air Speed	m/s	Log average air speed
Start ID		Log start sector (cell) ID
End ID		Log end sector (cell) ID
Road Distance	m	Log traversed road distance
Road Speed	m/s	Log average road speed
Mode		Log transport mode

The observed data consists of five time intervals in a day. Due to the length of the intervals, each log belongs to only one interval. Table 4.4 describes each time interval in the data set.

Table 4.4 Cellular OD time intervals

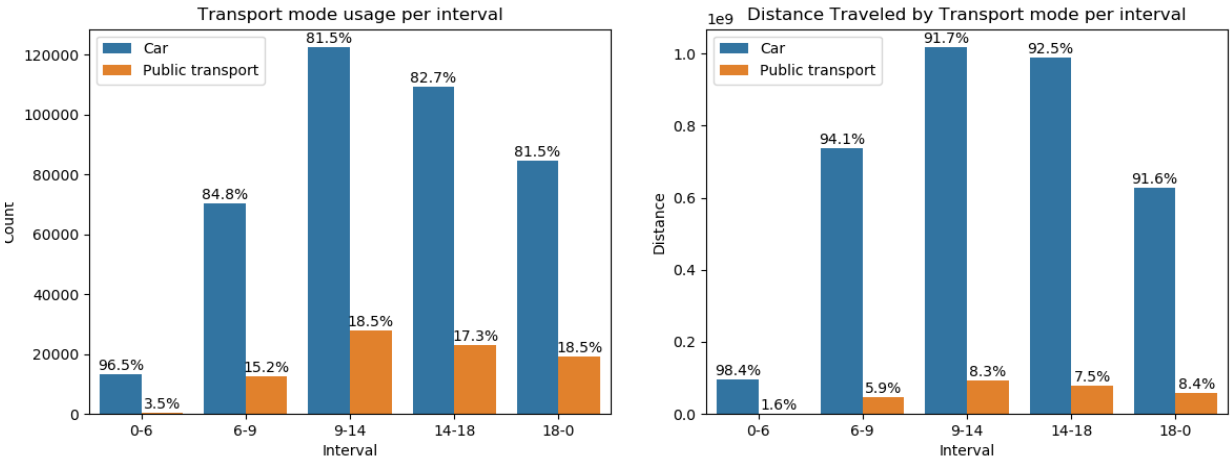
Value	Description
0	Time interval spanning from 00 h to 06 h
1	Time interval spanning from 06 h to 09 h
2	Time interval spanning from 09 h to 14 h
3	Time interval spanning from 14 h to 18 h
4	Time interval spanning from 18 h to 00 h

The user logs in the data set were classified into four transport modes describes in Table 4.5.

Table 4.5 Cellular OD transport modes

Value	Description
0	Car
1	Public transport
2	Walking
3	Cycling

Considering the FCD data set contains only vehicle movement historical records, in this thesis only transport modes "Car" and "Public transport" were observed. Figure 4.4 visualizes the distribution of the observed transport modes usages per interval in the data set.



(a) Total transport mode usage per interval **(b)** Total distance traveled by transport mode per interval

Figure 4.4: Transport mode usage per interval

In addition to the user logs, the cellular OD data set contains a *.shp* file spatially describing the coverage area of the cells. The attributes of the shapefile are listed in Table 4.6 (It has to be noted that not all cells contain a value for the attribute "Population").

Table 4.6 Cellular shapefile description

Attribute	Description
Name	Name of the cell given by the providing company
Web address	Official web-address of the cells *flagship* city or landmark
Population	Number of residents in the cell area
ID	ID of the cell

4.1.4. FCD pre-processing

To be able to estimate the user trajectory using a shortest-path algorithm, the original FCD data needs to be processed in such a manner that the output resembles the complete road network with each road segment containing the average speed per interval described in Table 4.4. The processing was performed using the object-oriented programming language *C#*[2].

[2]C# - https://docs.microsoft.com/en-us/dotnet/csharp/

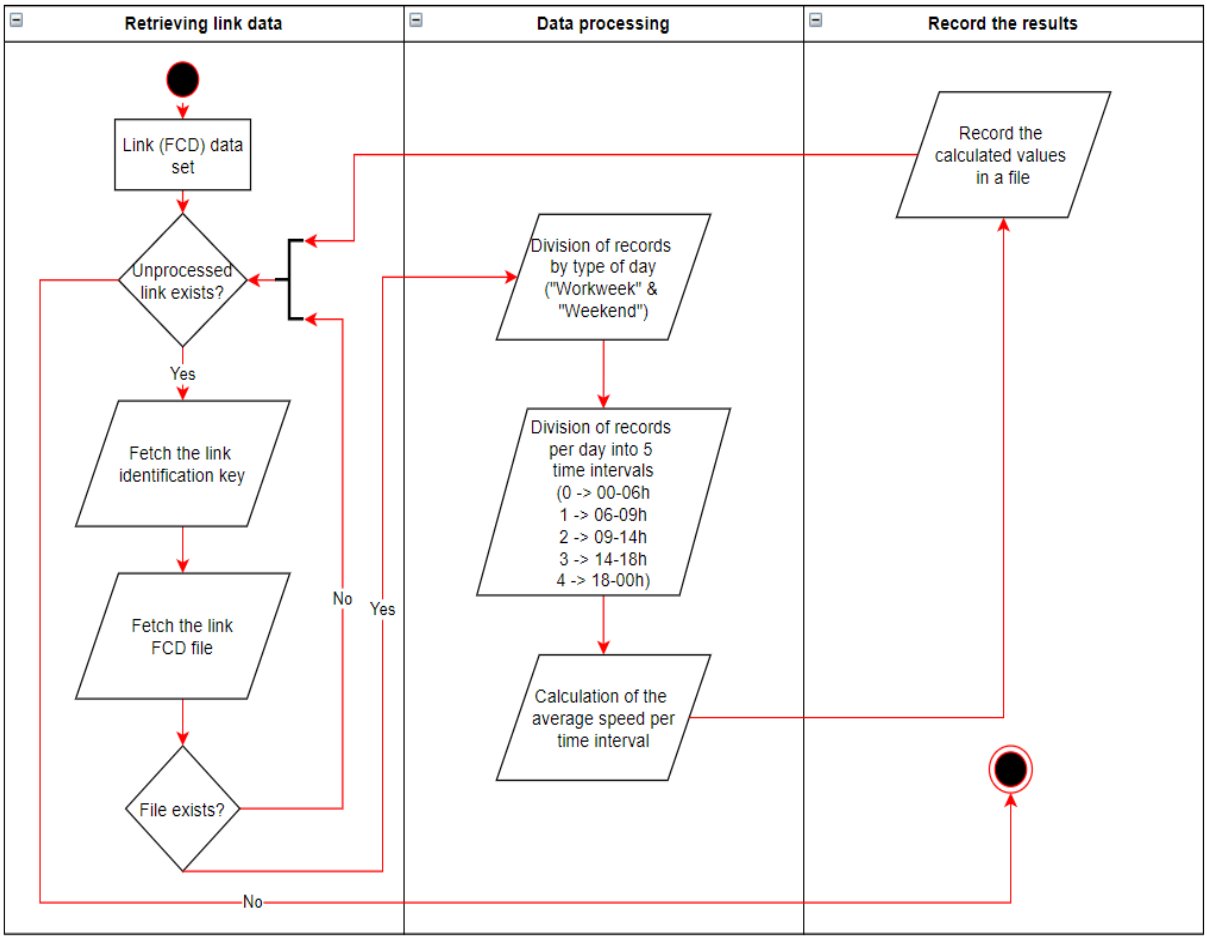


Figure 4.5: FCD data pre-processing process flow diagram

The complete process flow diagram of the FCD pre-process activities is visualized in Figure 4.5.

Process flow activities description:

– ***Retrieving link data***

- The "Retrieving link data" layer of the algorithm is a set of commands that, based on the obtained identification key, find the file in which all the recorded GPS records of the observed link are located. This layer serves as a condition for executing all other layers of the algorithm. The identification key of the link represents the element in the first column in Table 4.1

– ***Data processing***

- The first step of the layer "Data processing" is the process of converting epoch UTC to be able to sort the data into two-day types. The UTC variable represents the element in the first column in Figure 4.2
- The second step refers to the process of sorting the records of each day type into appropriate time intervals described in Table 4.4

- The final step of the layer is the process of calculating the average speed of the observed link following the time interval and day type as shown in Formula 4.1

$$\overline{V} = \frac{\sum_{i=1}^{N} V_i}{N} \tag{4.1}$$

Table 4.7 Formula 4.1 parameter description

Parameter	Description
$\overline{V}$	Average speed of the link in the observed interval and day type
V_i	Record speed in the observed interval and day type
N	Number of records in the observed interval and day type

– ***Record the results***

- The final layer of the algorithm writes the calculated average speeds from the previous layer into the appropriate file

The output of the process is an *.txt* file. Each line of the file contains a unique link, its metadata and calculated average speeds per interval for two types of days. Example of a link output (Table 4.9 describes the attributes of the following example):

$$Link\ metadata;Workweek;T_0\ speed;T_1\ speed;T_2\ speed;T_3\ speed;T_4\ speed$$
$$;Weekend;T_0\ speed;...;T_4\ speed$$

Table 4.8 Link output description

Attribute	Description
Link metadata	Static information of the link described in Table 4.1 In comparison to the original attributes in Table 4.1, the link output metadata stores the Link Road Category attribute instead of the Link Way attribute
Workweek	Days ranging from Monday to Friday
$T_x\ speed$	Workweek average speed of the link for the time interval x
Weekend	Days ranging from Saturday to Sunday
$T_x\ speed$	Weekend average speed of the link for the time interval x

4.1.5. Link-Cell assignment

For each user, only the start cell ID and the end cell ID is defined. Thus, to be able to estimate the user's trajectory from a predefined set of alternative routes, each link has to be assigned to an appropriate cell ID. To geo-assign road segments into cells, an open-source application, QGIS[3], was used. A shapefile (*.shp*) format that spatially describes the coverage area of cells using polygons and the FCD data with geo-coordinates of the links serve as input to the QGIS program. As output an enriched FCD data (Table 4.9) *.txt* file is generated.

To assign a link into an appropriate cell a QGIS built-in method *Vector selection*[4] was used. The first step was loading the *.shp* file of the cells and the *.txt* file of the road network into QGIS. After loading the network and the cell coverage areas, the *Extract by location* method was used upon the created layers. If links starting coordinate was within a cell polygon, the cell ID was appended to that link.

Figures 4.6 and 4.7 visualize the distribution of links per cell. As the FCD data set was collected by tracking the movement of primarily logistical vehicles and taxis, the link usage of the same mainly revolved around the usage of main roads. Furthermore, while some cells contain a substantial amount of links within them, some cells do not contain any or do contain a sum of fewer than 10 links. The locations and distributions of links serve as a crucial factor when estimating the user trajectories. Thus, when analysing the results, one must bear in mind the spatial distribution of the used data.

[3]QGIS - https://www.qgis.org/en/site/

[4]QGIS Documentation - https://docs.qgis.org/3.16/en/docs/user_manual/processing_algs/qgis/vectorselection.html

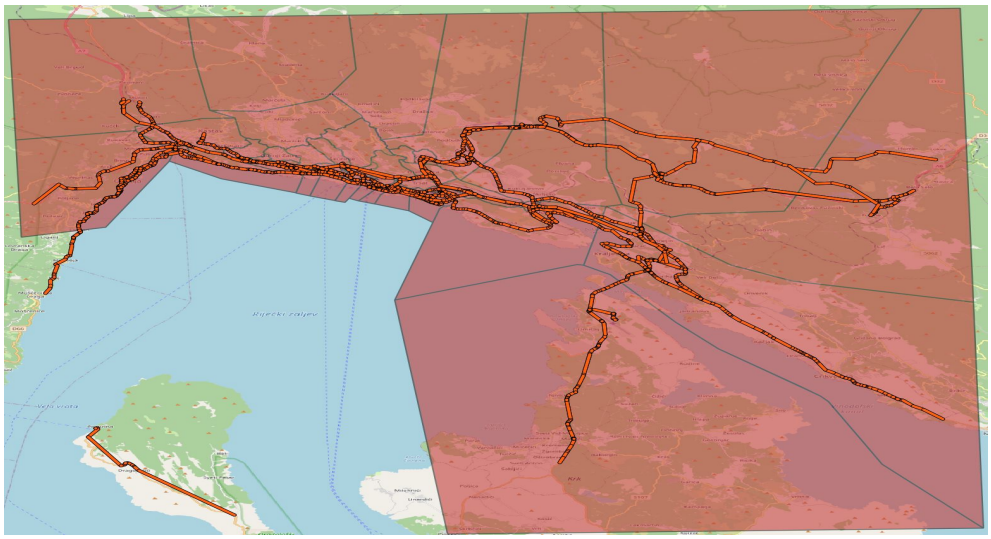


Figure 4.6: Cell shapefile coverage area and FCD links

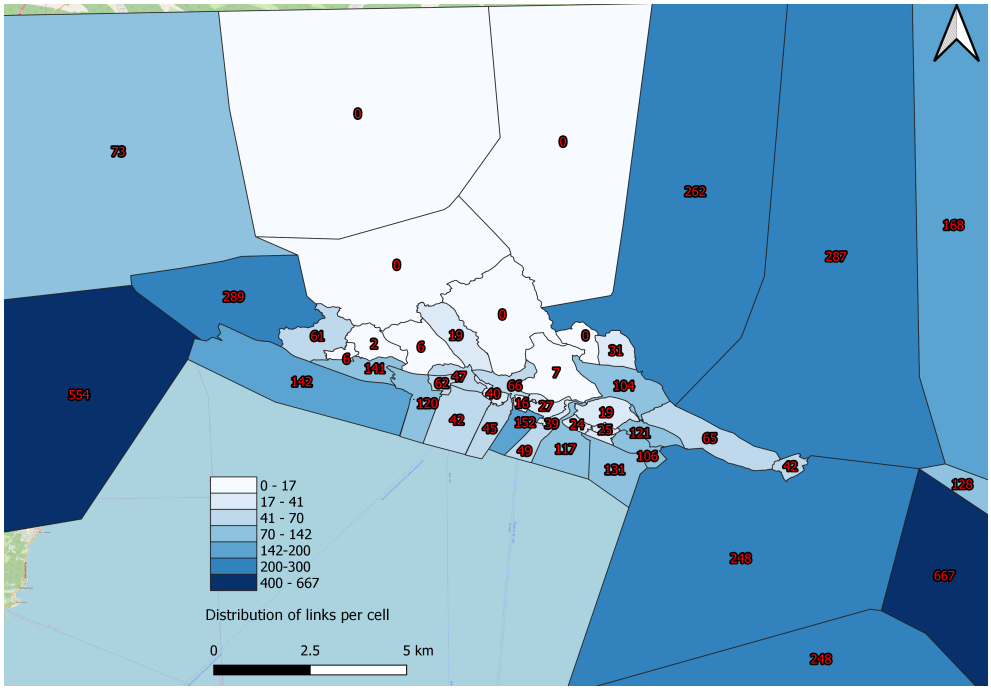


Figure 4.7: Link distribution per cell

4.2. Trajectory estimation

4.2.1. Overview

The following section describes the process of estimating user trajectories from the pre-processed data sets.

The general approach of estimating the user trajectories is illustrated in Figure 4.8. The aim is to fetch a predefined set of alternative routes for each OD-pair and select the route that has the highest spatio-temporal similarity with the ground truth cellular OD-pair.

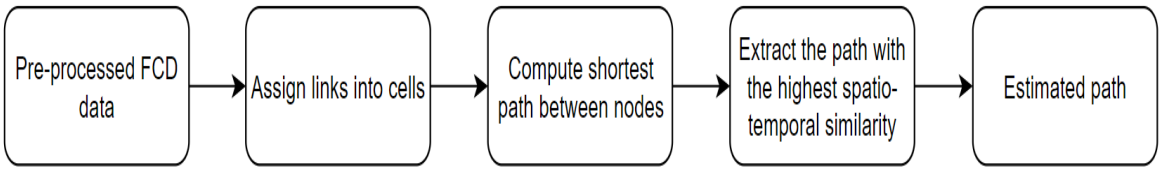


Figure 4.8: General trajectory estimation workflow

The first approach based solely on the length of edges in the vortex is described in section 4.2.2. To be able to generate more accurate results, a modified version of the first approach is described in section 4.2.3. This approach compares the spatial and temporal aspects of the predefined set of routes to the ground truth paths. The third approach tested in this thesis, described in section 4.2.4, is based on the road categories and the time needed to traverse the edges in the graph.

4.2.2. Length based edge criteria

It is assumed in this approach that individuals tend to follow the shortest path when travelling between locations. Therefore, when computing the shortest path algorithm, the weight of edges is equal to the length of the road segment.

For each node in the graph, the algorithm iterates through the whole set of candidate edges and retrieves the shortest paths to all other nodes that have possible connections to the start node. This step is computed for each time interval described in Table 4.4. Input data of the algorithm represents the pre-processed FCD data set containing the metadata of the observed road network. The output of the algorithm is a set of nodes of the network and their corresponding weights describing the total required difficulty (length) for traversing from the start node to the observed node. The shortest-path algorithm computed for each node is described in 3.1 and shown in Algorithm 1.

The final output of the process is a set of alternative routes covering the observed road network. Each route contains a set of road segments ranging from the first to the last link in the

route and their respective cell ID's, the total traversed length of the route, average speed and the time needed to traverse the route. With the obtained route metadata and the metadata of each cellular user (Table 4.3), similarity computation is conducted to select the route that has the highest spatial similarity with the observed ground-truth cellular OD-pair. The Pseudocode is shown in Algorithm 2 while the used variables are described in Table 4.10. Input to the algorithm is a list of cellular users and the set of alternative routes of the network. As an output, the algorithm generates a *.txt* file containing cellular users and their respective "best-fit" routes. Description of the algorithm:

- The first step of the algorithm is to load a *.txt* file which contains all the cellular OD user logs (each line contains the metadata of one user) and to load the *.txt* file containing the road network alternative routes generated in the prior process.
- Thereafter, for each user's log, select only links that are either geo-located in the user log's start cell ID or end cell ID. By doing so, the algorithm narrows down the possible paths. If such a path exists, it is added to a list of valid routes of the observed user. Appended values format:

  Start Link ID;End Link ID;Route Length;Route Time;Route Average Speed
- If the valid routes list is not empty, the algorithm iterates through the list and calculates the similarity of the observed path's length to the ground-truth observed users travelled distance. As shown in Formula 4.2, the similarity is calculated in such a manner that the ground-truth length traversed by the user resembles a 100% accuracy (similarity) value to which all candidate values will be compared.

$$S = \frac{L_P}{L_U} * 100 \tag{4.2}$$

Table 4.9 Formula 4.2 parameter description

Parameter	Description	Unit
S	Measure of similarity	%
L_P	Observed route length	m
L_U	Observed user logged length	m

- After measuring the accuracy of each value, the overall accuracy of the selected path is further calculated by comparing each attribute (length, time, speed) of the path to the ground-truth path. Lastly, the algorithm stores the path with the highest travelled length similarity.

Algoritam 2 Spatial similarity computation algorithm

Input: U, L

Output: Q

for $user\ u$ in U **do** *# Main loop*

if $logMode \neq 0$ or $logMode \neq 1$ **then** *# Only transport modes Car and*

$continue$ *# Public transport are valid*

end if

$validConn \leftarrow ()$ *# Initialise a empty valid connection list*

for $sLink$ in L **do** *# Each link is viewed as a route start point*

if $sLink$ is not in $u.startCellID$ **then** *# Link has to be in the users start cell ID*

$continue$

end if

for $eLink$ in L **do** *# Inner loop, each link is viewed as a route end point*

if $eLink$ is not in $u.endCellID$ **then** *# Link has to be in the users end cell ID*

$continue$

end if

$tempLen \leftarrow L[sLink][eLink][u.interval].length$

$tempTime \leftarrow L[sLink][eLink][u.interval].time$

$tempSpeed \leftarrow L[sLink][eLink][u.interval].speed$

$validConn.Add(sLink.ID, eLink.ID, tempLen, tempTime, tempSpeed)$

end for

end for

if $validConn$ is not $empty$ **then**

$bestMatch.sim \leftarrow 0$ *# Initialise a variable for storing the best match path*

for $tempPath$ in $validConn$ **do**

$tempPath.sim \leftarrow \frac{tempPath.length}{u.length} \times 100$ *# Additional steps ensure the value is never above 100*

if $tempPath.sim > bestMatch.sim$ **then** *# A closer similarity is found*

$bestMatch \leftarrow tempPath$

end if

end for

end if

$bestMatch.overallAccuracy \leftarrow \frac{(\frac{bestMatch.length}{u.length} + \frac{bestMatch.time}{u.time} + \frac{bestMatch.speed}{u.speed}) \times 100}{300}$

$Q.Add(u, bestMatch)$ *# Append the output with the user metadata*

 # and the best match path metadata

end for

Table 4.10 Algorithm 2 variable description

Variable	Description
U	List of all users and their respective logs
L	List of pre-processed road segments on the observed network
Q	Output list containing users and their "best-match" paths
u	Current observed user
logMode	Transportation mode classified in the user's log
validConn	List containing valid paths for the observed user
sLink	Link that is within the log's start ID cell
eLink	Link that is within the log's end ID cell
tempLen	Traversed length of the observed path in the log's time interval
tempTime	Total time required to traverse the observed path in the log's time interval
tempSpeed	Average road speed of the observed path in the log's time interval
bestMatch	Variable which stores the current highest similarity path
tempPath	Current observed path in the list of valid connections

4.2.3. Modified length based edge criteria

The observed road network has a total of 4924 links, thus resulting in approximately 4924! combinations of paths. Out of the pool of all valid paths, a great number of them will likely have a perfect traversed length match with the observed ground-truth cellular OD-pair, yet their temporal and velocity values will be nowhere near their ground-truth counterparts. Thus the similarity computation algorithm described in section 4.2.2 will have a high probability of selecting a route which shares no characteristic to the ground-truth path, except spatially. To be able to bypass that issue, this method aims to enrich the process of similarity computation by considering the spatial and temporal attributes of each observed route.

The similarity computation follows the first two steps of loading the graph and processed data and the creation of a list containing the valid links, as described in section 4.2.2. Thereafter, the algorithm validates each route from the list and selects paths that have a traversed length similarity value greater or equal to 90% of the observed ground-truth path. Such an approach aims to create a pool of paths that have a high similarity level to the ground-truth traversed length. The selected paths are subject to further validation in terms of calculating their spatial and temporal similarities to the observed ground-truth path. The process of validating the selected paths and choosing the "best-match" path is shown in Algorithm 3 while the presented variables are described in Table 4.11. Description of the algorithm:

- If the valid connections list is not empty, for each path calculate the traversed length similarity. If the calculated value is greater or equal to 90%, store the path to a new list.
- If the list containing valid paths with the traversed length similarity greater or equal to 90% is not empty, iterate through each path and calculate the sum of the following

combinations:

- 1. Length and Time accuracy sum
- 2. Length and Speed accuracy sum
- 3. Length, Time and Speed accuracy sum

– It is assumed that a path is valid if at least two out of three attributes describing it has a high level of similarity to the ground-truth path. Thus, for each calculated combination type, the algorithm checks if it is greater than the previous highest sum of the same combination type.

– To be able to validate all three combinations, a process of normalisation is required. Since the first two combinations calculate the sum of two attributes, the highest reachable value is equal to 200. The third combination is a sum of three attributes, each attribute has a maximum value of 100, thus the sum of the combination is equal to 300. To be able to determine which combination has the highest similarity value, all combinations are divided by their maximum sum and normalised to a range of 0 to 1.

– Finally, the algorithm assigns the path with the highest similarity value to the observed ground-truth path.

Table 4.11 Algorithm 3 variable description

Variable	Description
validConn	List containing valid paths for the observed user
tempPath	Current observed path in the list of valid connections
above90	List containing valid paths with the length similarity value greater or equal to 90%
globalLenTime	Path with the greatest Length and Time accuracy sum
globalLenSpeed	Path with the greatest Length and Speed accuracy sum
globalLenTimeSpeed	Path with the greatest Length, Time and Speed accuracy sum
$tempLenTime_{accuracy}$	Sum of the Length and Time accuracy of the observed path in list "above90"
$tempLenSpeed_{accuracy}$	Sum of the Length and Speed accuracy of the observed path in list "above90"
$tempLenTimeSpeed_{accuracy}$	Sum of the Length, Time and Speed accuracy of the observed path in list "above90"
bestMatch	Path with the highest similarity to the observed ground-truth path

Algoritam 3 Spatio-temporal similarity computation algorithm

if $validConn$ is not $empty$ **then**
  for $tempPath$ in $validConn$ **do**
    $tempPath.sim \leftarrow \frac{tempPath.length}{u.length} \times 100$
    if $tempPath.sim >= 90$ **then** *# A value greater that 90% is found*
      $above90.Add(tempPath)$
    end if
  end for
  if $above90.length > 0$ **then**
    $globalLenTime$ *# Initialisation of variables for storing the best-match path, values equal to 0*
    $globalLenSpeed$
    $globalLenTimeSpeed$
    for $tempPath$ in $above90$ **do**
      $tempLenTime_{accuracy} \leftarrow tempPath.length_{accuracy} + tempPath.time_{accuracy}$
      $tempLenSpeed_{accuracy} \leftarrow tempPath.length_{accuracy} + tempPath.speedaccuracy$
      $tempLenTimeSpeed_{accuracy} \leftarrow tempPath.length_{accuracy} + tempPath.time_{accuracy} + tempPath.speed_{accuracy}$
      if $tempLenTime_{accuracy} > globalLenTime.accuracy$ **then**
        $globalLenTime \leftarrow tempPath$
      end if
      if $tempLenSpeed_{accuracy} > globalLenSpeed.accuracy$ **then**
        $globalLenSpeed \leftarrow tempPath$
      end if
      if $tempLenTimeSpeed_{accuracy} > globalLenTimeSpeed.accuracy$ **then**
        $globalLenTimeSpeed \leftarrow tempPath$
      end if
    end for
  end if
  $globalLenTime.accuracy \leftarrow \frac{globalLenTime}{200}$ *# Each considered value can have a max sum of 100*
  $globalLenSpeed.accuracy \leftarrow \frac{globalLenSpeed}{200}$
  $globalLenTimeSpeed.accuracy \leftarrow \frac{globalLenTimeSpeed}{300}$
  $bestMatch \leftarrow Math.Max(globalLenTime.accuracy, Math.Max(globalLenSpeed.accuracy, globalLenTimeSpeed.accuracy))$
  # The path with the highest value is appended to the ground-truth path

4.2.4. Time based edge criteria

The time-based edge criteria algorithm works similarly to the previously described algorithms 4.2.2 and 4.2.3, except that the considered attribute is time. It is assumed in this approach that individuals tend to follow the time-wise shortest path when travelling between locations. Therefore, when computing the shortest path algorithm, the weight of edges is equal to the time required to traverse the road segment. Additionally, when calculating the weight of the link, the time value is adjusted according to the road category. The fastest path, thus time-wise shortest, is presumed to be the path that uses road categories such as motorways, highways, etc. Table 4.12 describes the value modifications based on the link road category. Each link's required time to traverse its length was modified in such a manner that with the increase of the theoretical speed of the road category, a decrease in the calculated time needed to traverse it was conducted. *Exempli gratia*; if a path contains a road segment that is categorised as a Highway, the weight of that segment is decreased by 10%. Such an approach aims to increase the probability of the algorithm selecting that link in its path. A similar approach was conducted per [1], where it was assumed that the probability of a user traversing a road segment increased with the importance of the category of the respective segment. The Pseudocode of the shortest-path algorithm applied to each link is shown in Algorithm 4.

Table 4.12 Road category weight adjustment

Road type	Weight adjustment
Highway	Weight = Weight - Weight * 0.1
Expressway	Weight = Weight - Weight * 0.09
City Avenue / Main Road	Weight = Weight - Weight * 0.08
State Road	Weight = Weight - Weight * 0.04
Main Street	No adjustment
Local Street	No adjustment

Algoritam 4 Road type time adjusted Dijkstra's algorithm

Input: v_i, G

Output: Q

$time[v_i] \leftarrow 0$ # *Time required to traverse from v_i to v_i*

for *node* v in G **do** # *Initialisation*

if $v \neq v_i$ **then**

$time[v] \leftarrow infinity$ # *Unknown time from v_i to v*

$previous[v] \leftarrow undefined$ # *Prior node in the path from v_i*

end if

$add\ v\ to\ Q$ # *All nodes initially in Q*

end for

while Q is not *empty* **do** # *Main loop*

$u \leftarrow node\ in\ Q\ with\ min\ time[u]$ # *v_i in the first step*

$remove\ u\ from\ Q$

for *neighbour* v of u **do** # *With v still not removed from Q*

$valueL \leftarrow length(u, v)$ # *Length of the observed segment*

$valueS \leftarrow speed(u, v)$ # *Speed of the observed segment*

$valueT \leftarrow \frac{\frac{valueL}{1000}}{valueS}$ # *Time required to traverse the observed segment*

$adjustedT \leftarrow valueT \times adjustmentFactor[roadType(u, v)]$

$alt \leftarrow time[u] + valueT$ # *Total time of the current path*

if $alt > time[v] + adjustedT$ **then** # *A faster path towards node v is found*

$time[v] \leftarrow alt$

$previous[v] \leftarrow u$

end if

end for

end while=0

After the successful creation of a weighted graph of the road network, similarity computation is conducted to select the route that has the highest Spatio-temporal similarity to the observed ground-truth cellular OD-pair. As described in sections 4.2.2 and 4.2.3, the first two steps of the algorithm are loading the processed data and creating a list of valid connections. Furthermore, out of the valid connections list, the algorithm creates a new list in which it stores only paths that have a travelled time similarity value greater or equal to 90% of the observed ground-truth path. Such a list aims to create a pool of paths that have a high similarity level to the ground-truth travelled time. If the newly created list is not empty, each path in the list is

subject to a process of calculating the sum of the following combinations:

$$Time - Length_{accuracy} = \frac{Time_{accuracy} + Length_{accuracy}}{200} \tag{4.3}$$

- The maximum accuracy in 4.3 is equal to the value 1 ((100 + 100) / 200)

$$Time - Speed_{accuracy} = \frac{Time_{accuracy} + Speed_{accuracy}}{200} \tag{4.4}$$

- The maximum accuracy in 4.4 is equal to the value 1 ((100 + 100) / 200)

$$Time - Length - Speed_{accuracy} = \frac{Time_{accuracy} + Length_{accuracy} + Speed_{accuracy}}{300} \tag{4.5}$$

- The maximum accuracy in 4.5 is equal to the value 1 ((100 + 100 + 100) / 300)

The algorithm aims to iterate through all paths and find the one whose combination (either one of the described combinations) has the highest Spatio-temporal value. After iterating through the whole list, the "best-match" path is appended to the observed ground-truth cellular OD-pair.

4.3. Traffic flow estimation

As described in section 3.2, and shown in expression 4.6, traffic flow indicates the number of vehicles that pass through a cross-section of an observed road in a unit of time.

$$q = gV \ [veh/h] \tag{4.6}$$

Thus, to be able to calculate the traffic flow, it is required to know the speed of the vehicles and the vehicle density of the observed area during the observation period. By estimating the user paths, this thesis has generated a set of links on which it is assumed that the observed vehicles (users) travelled. Each generated path contains the metadata of its links with an emphasis on the link length and the user's average velocity on the link. Thus, for each observed link, the mean velocity could be calculated by the following expression where $\overline{v}\ [km/h]$ is the mean

velocity, v_i $[km/h]$ is the velocity of the observed user and n is the count of all observed users:

$$\overline{v} = \frac{\sum_{i=1}^{n} v_i}{n} \tag{4.7}$$

and the vehicle density of the observed link by the following expression where g $[veh/km]$ represents the density, N is the number of observed users and s $[km]$ is the length of the observed link:

$$g = \frac{N}{s} \tag{4.8}$$

However, the described expressions 4.7 and 4.8 must take into account a observation time-window. Since the user time intervals in this thesis range from three (3) to six (6) hours, the process of allocating a user to a specific link in a specific time frame is subject to a high degree of uncertainty. Without data on when the users started their route, modelling the user's location at a certain time-stamp may result in an occurrence of either none, one or multiple users traversing the observed link at the time-stamp. *Exempli Gratia;* multiple users have the same starting location (link) of their respective route but do not commence their trips at the same time, nevertheless, the chosen model may allocate each user to the link at the same time interval despite their time of commute. Consequently, not knowing how many users are on a link in an observed time interval results in calculating the link density and velocity with a high degree of uncertainty.

Regarding the inability to calculate the *standard* free-flow speed of the network, this thesis will indirectly estimate the state of the traffic flow by calculating the deviation of the mean vehicle velocity of all users from the free-flow speed of the observed link. It is assumed in this approach that the *FFS* (Free Flow Speed) of a link is equal to the speed limit of the road segment described in section 4.1.2. Furthermore, while computing the trajectory estimation, the velocity of a link resembled an aggregated value of the entire interval and was hence the same for each user. Therefore, in this approach, the mean speed of all users crossing a link is equal to the pre-allocated velocity of the road segment.

A description of the traffic flow estimating algorithm is given below:

1) The first step of the algorithm is to load a *.txt* file which contains all estimated user paths of the observed method and the pre-processed *FCD* data set *.txt* file containing the complete road network with an emphasis on links, their speed limits and calculated mean speeds in intervals.

2) Thereafter, for each link in the generated OD cellular user's trajectory estimate the deviation of its speed from the FFS value; velocity[t] represents the mean velocity value on the link in the observed time interval t while FFS describes the free-flow speed that is equal

to the speed limit of the observed link:

$$Deviation = \frac{velocity[t]}{FFS}[\%] \tag{4.9}$$

3) Finally, after calculating the deviation, store it to the appropriate time interval and link.

The outcome of the algorithm is a list of links and their deviations from the FFS in the observed time interval. As the deviation is denoted as a percentage value of the FFS, the final verdict of the links traffic flow at the observed time interval is classified into two groups. Links with values lower than 90% of the FFS are considered to be congested at the time of observation and links with values greater or equal to 90% are considered as road segments with a high count of vehicles per hour, *i.e.* the average velocity of the link is greater or equal to the free-flow speed.

5. Results

The following section analyses and interprets the results generated by the trajectory estimation algorithms described in section 4.2 and the indirect traffic flow estimation algorithm described in section 4.3.

5.1. Trajectory estimation algorithms

The results of the trajectory estimation algorithms are grouped into three categories of transport mode data:

1) Transport mode: Car
   - Similarity statistic of users that traversed their route using a Car as a transport mode. Tables 5.1, 5.2 and 5.3 give an overview of the distribution of the similarity results for the different trajectory estimation methods. Each table describes the similarity distribution per interval and an overall distribution that combines data from all observed intervals. In addition, Figure 5.1 shows a bow plot distribution of the similarity value for the different methods.

2) Transport mode: Bus
   - Similarity statistic of users that traversed their route using a Bus as a transport mode. Tables 5.4, 5.5 and 5.6. Box plot Figure 5.2.

3) Transport mode: Car and Bus
   - Similarity statistic of users that traversed their route using either a Car or a Bus as a transport mode. Tables 5.7, 5.8 and 5.9. Box plot Figure 5.3.

The overall mean and median values from the methods proposed in this study are, in general, moderately low. Significantly higher similarity levels can be seen in estimated trajectories of users that used a car as their transport mode rather than a bus. When computing the shortest path, the proposed methods did not take into account the additional time aspect of a bus route which contains n stops. Each bus stop is associated with an increase in the time needed to traverse a road segment. As the FCD data in this study was generated by mostly tracking delivery vehicles

and taxis, the calculated speed and consequently time attributes of the road segment, differed significantly from a typical bus commute on the same link.

In theory, the proposed methods aimed to generate a pool of alternative routes that contained the entire observed road network. Following, for each user's path metadata, find the alternative path with the highest spatiotemporal similarity. With that in mind, the expected similarity results were much higher as by iterating through all possible routes, there is bound to be a route with an overall similarity level greater or equal to 90% of the ground-truth path. The moderately low results, overall and per each transport mode, could be caused by the nature of the FCD data set, which served as a principal data source for the road network. As stated above, the FCD data was generated by mostly tracking delivery vehicles and taxis. The commute routes of service vehicle may differ from commute routes of personal as the prior tend to not deviate significantly from the main roads, which in general, provide the fastest routes from point A to point B. The daily commutes of the latter may commence or end in residential areas, which contain road segments that are generally defined as slow and narrow. Those links do not fit the profile of a road segment traversed by a service vehicle. With the assumption that service vehicles tend to use paths with higher mean velocities, it is likely that the start or end road segments of users associated with personal vehicles may have not been traversed by service vehicles and are as such not contained in the FCD data set. With the absence of such data, the number of alternative routes which may generate a higher similarity level of the observed ground-truth path decreases.

Considerably higher overall results generated in Case 2 (Method 4.2.3) in contrast to Case 1 (Method 4.2.2) suggest that an approach where only the length of the path is considered does not manage to precisely reconstruct the ground-truth route. The introduction of additional analysis parameters that spatio-temporary measure the validity of an alternative path enables the algorithm to find a route that has a higher level of similarity with the observed ground-truth user's trajectory. Moreover, higher overall median results generated in Case 2 (Method 4.2.3) in contrast to Case 3 (Method 4.2.4) indicate that the ground-truth path (users) in this study tended to follow the shortest path when travelling between two locations rather than the fastest path.

Table 5.1 Transport mode: Car; Case 1 similarity statistics, Method 4.2.2

Interval	Mean [%]	STD [%]	Min [%]	1st Qu. [%]	Median [%]	3rd Qu. [%]	Max [%]
00-06 h	44.88	27.34	0.10	23.51	39.42	66.93	99.96
06-09 h	44.24	26.70	0.12	23.18	39.96	64.81	99.96
09-14 h	44.95	26.79	0.10	23.78	40.67	66.12	99.99
14-18 h	44.38	26.56	0.13	23.51	39.86	64.73	99.99
18-00 h	45.65	26.75	0.12	25.10	40.84	66.62	99.99
Overall	44.85	26.74	0.10	23.89	40.36	65.74	99.99

Table 5.2 Transport mode: Car; Case 2 similarity statistics, Method 4.2.3

Interval	Mean [%]	STD [%]	Min [%]	1st Qu. [%]	Median [%]	3rd Qu. [%]	Max [%]
00-06 h	58.77	22.99	0.40	40.19	59.66	76.56	99.95
06-09 h	58.60	23.79	0.38	40.38	62.18	76.83	100
09-14 h	59.61	23.81	0.31	41.40	63.20	78.00	100
14-18 h	58.86	23.63	0.40	40.78	62.17	76.98	99.99
18-00 h	59.91	23.58	0.30	41.71	63.30	78.10	100
Overall	59.30	23.68	0.30	41.10	62.65	77.50	100

Table 5.3 Transport mode: Car; Case 3 similarity statistics, Method 4.2.4

Interval	Mean [%]	STD [%]	Min [%]	1st Qu. [%]	Median [%]	3rd Qu. [%]	Max [%]
00-06 h	46.77	27.34	0.41	27.04	33.33	71.94	99.94
06-09 h	50.69	26.37	0.49	31.20	41.29	74.93	99.99
09-14 h	52.28	26.62	0.42	32.21	44.13	77.01	100
14-18 h	51.28	26.45	0.43	31.43	42.19	75.77	99.99
18-00 h	51.70	26.67	0.40	31.78	41.47	76.89	100
Overall	51.44	26.60	0.40	31.57	42.10	76.25	100

Table 5.4 Transport mode: Bus; Case 1 similarity statistics, Method 4.2.2

Interval	Mean [%]	STD [%]	Min [%]	1st Qu. [%]	Median [%]	3rd Qu. [%]	Max [%]
00-06 h	25.83	19.86	0.62	10.93	22.87	34.60	96.03
06-09 h	21.60	12.15	0.13	10.72	21.20	31.60	97.98
09-14 h	21.24	11.99	0.10	10.08	21.96	31.75	80.95
14-18 h	21.04	11.86	0.10	10.34	21.77	31.17	94.68
18-00 h	21.11	12.38	0.09	10.51	21.51	31.88	99.45
Overall	21.23	12.13	0.09	10.35	21.82	31.36	99.45

Table 5.5 Transport mode: Bus; Case 2 similarity statistics, Method 4.2.3

Interval	Mean [%]	STD [%]	Min [%]	1st Qu. [%]	Median [%]	3rd Qu. [%]	Max [%]
00-06 h	44.85	17.94	0.81	35.48	54.64	58.90	78.74
06-09 h	44.77	16.25	0.38	35.25	43.05	58.81	99.63
09-14 h	44.64	16.42	0.36	35.28	42.85	58.73	92.59
14-18 h	43.63	16.12	0.35	34.44	41.23	58.16	97.65
18-00 h	43.98	15.83	0.34	34.91	40.96	58.20	92.90
Overall	44.23	16.19	0.34	34.98	41.96	58.44	99.63

Table 5.6 Transport mode: Bus; Case 3 similarity statistics, Method 4.2.4

Interval	Mean [%]	STD [%]	Min [%]	1st Qu. [%]	Median [%]	3rd Qu. [%]	Max [%]
00-06 h	22.41	9.57	0.83	13.79	23.20	32.04	41.40
06-09 h	24.08	9.31	0.36	16.44	24.87	32.92	90.61
09-14 h	24.46	9.55	0.38	16.47	25.67	33.19	79.16
14-18 h	24.11	9.42	0.36	16.22	24.96	33.00	77.40
18-00 h	23.89	9.15	0.34	16.20	24.81	32.94	64.18
Overall	24.17	9.39	0.34	16.31	25.12	33.08	90.61

Table 5.7 Transport mode: Car and Bus; Case 1 similarity statistics, Method 4.2.2

Interval	Mean [%]	STD [%]	Min [%]	1st Qu. [%]	Median [%]	3rd Qu. [%]	Max [%]
00-06 h	44.23	27.33	0.10	22.92	38.68	66.20	99.96
06-09 h	39.69	26.11	0.12	19.55	34.64	57.52	99.96
09-14 h	39.69	26.22	0.10	19.23	34.82	57.72	99.99
14-18 h	39.36	26.00	0.10	19.37	34.34	56.65	99.99
18-00 h	40.13	26.35	0.09	20.03	34.77	58.45	99.99
Overall	39.85	26.23	0.09	19.60	34.76	57.92	99.99

Table 5.8 Transport mode: Car and Bus; Case 2 similarity statistics, Method 4.2.3

Interval	Mean [%]	STD [%]	Min [%]	1st Qu. [%]	Median [%]	3rd Qu. [%]	Max [%]
00-06 h	58.29	22.97	0.40	39.77	59.16	78.85	99.95
06-09 h	55.82	23.15	0.38	38.29	58.22	72.59	100
09-14 h	56.29	23.23	0.31	38.89	58.65	73.39	100
14-18 h	55.58	23.09	0.35	38.33	57.76	72.29	99.99
18-00 h	56.33	23.05	0.30	38.71	58.43	73.16	100
Overall	56.11	23.14	0.30	38.64	58.32	73.01	100

Table 5.9 Transport mode: Car and Bus; Case 3 similarity statistics, Method 4.2.4

Interval	Mean [%]	STD [%]	Min [%]	1st Qu. [%]	Median [%]	3rd Qu. [%]	Max [%]
00-06 h	45.93	27.28	0.41	26.17	33.33	70.75	99.94
06-09 h	45.34	26.20	0.36	26.69	33.82	66.88	99.99
09-14 h	46.06	26.55	0.38	27.09	33.96	69.13	100
14-18 h	45.39	26.31	0.36	26.58	33.59	67.43	99.99
18-00 h	45.39	26.54	0.34	26.66	33.33	68.28	100
Overall	45.63	26.46	0.34	26.77	33.33	68.24	100

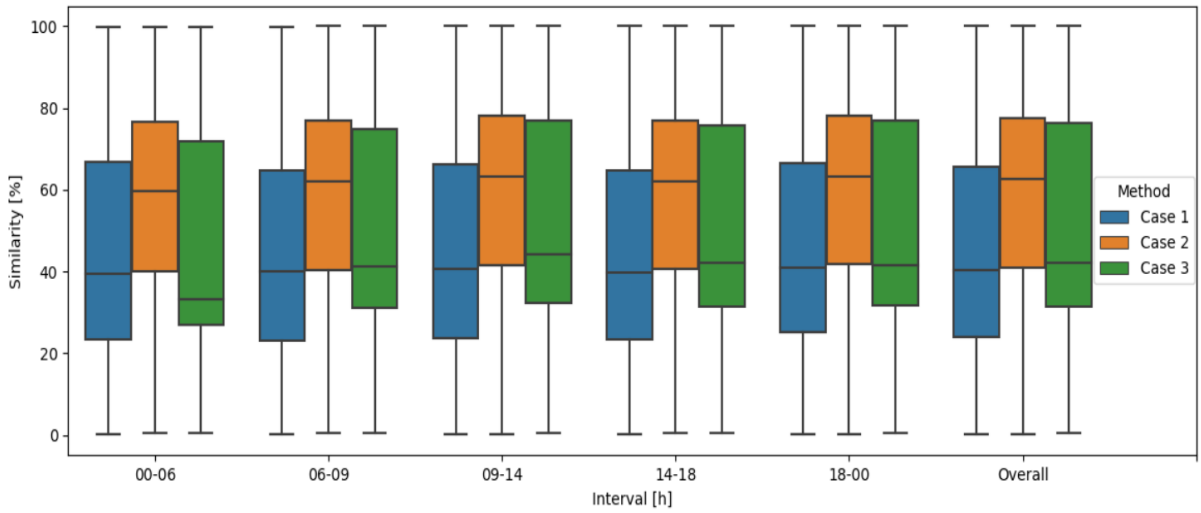


Figure 5.1: Transport mode: Car; Distribution of SM results per trajectory estimation method

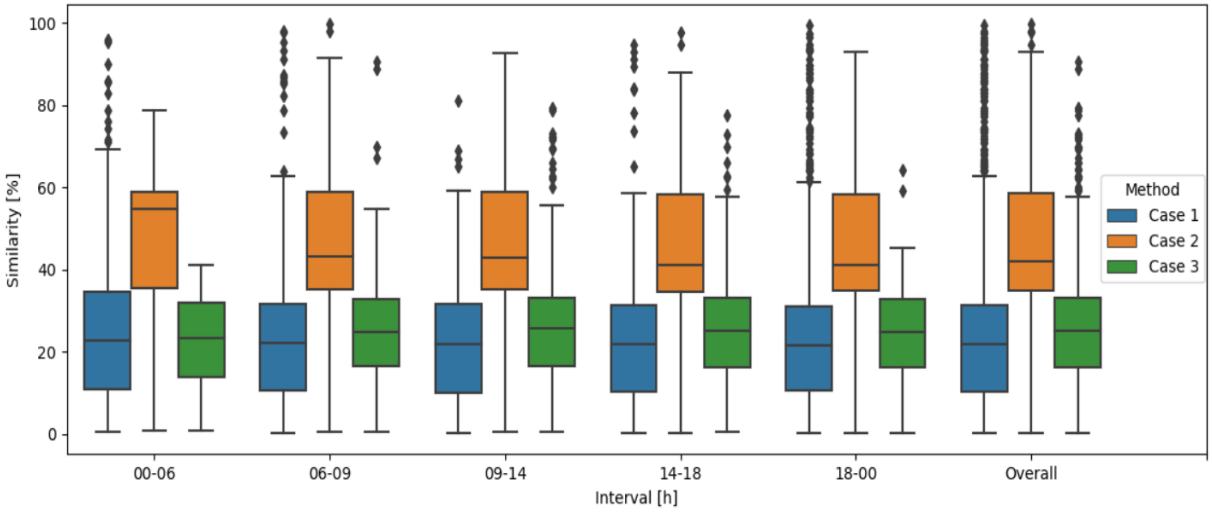


Figure 5.2: Transport mode: Bus; Distribution of SM results per trajectory estimation method

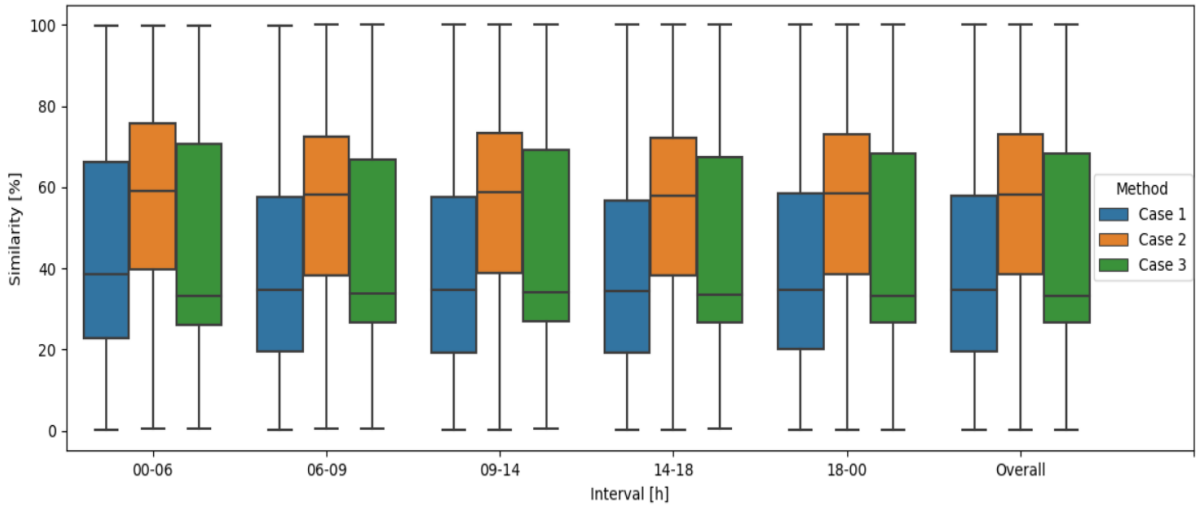


Figure 5.3: Transport mode: Car and Bus; Distribution of SM results per trajectory estimation method

Figures 5.4 and 5.5 illustrate results of the trajectory estimation methods for a same OD user. The purple line indicates the Case 2 method. The Case 2 method generated a pool of alternative paths with a traversed length similarity greater or equal to 90% of the ground-truth OD-pair and selected the path with the highest overall similarity. The green line represents the chosen estimated path from Case 3, where a similar process of trajectory estimation was conducted, with the difference of giving the priority to the time variable over the traversed length variable of the path. Lastly, Case 1 is coloured in orange. The Case 1 method selected a path with the highest traversed length similarity to the ground-truth path, no matter the overall accuracy. In Case 2, it is estimated that the user started the route in the proximity of the city of Hreljin and ended in the vicinity of the city Krasica. Case 1 and Case 3 estimated routes began on the outer edges of the cities of Kraljevica and Bakarac but ended in the middle of the road. It is important to note that the road network of the mentioned cities is not covered in the scope of the FCD data set. No coverage of the cities results in the shortest path algorithm not analysing its links. Thus, analysing the estimated path in method Case 1, one could assume the user began the route in the city of Hreljin and ended in the city of Krasica. The overall similarity of the methods in comparison to the ground-truth user's path: Case 1 - 56%, Case 2 - 98%, Case 3 - 78%.

Figure 5.4: Case 1 (Orange) and Case 2 (Purple) trajectory estimation results for a same user

Figure 5.5: Case 2 (Purple) and Case 3 (Green) trajectory estimation results for a same user

5.2. Traffic flow estimation

Without data on when the users started their routes, modelling the density and average speed of the observed road segment in a certain time frame is subject to a high degree of uncertainty. Thus, rather than estimating the standard traffic flow described in section 3.2, this Thesis utilized a method of indirectly estimating the state of the traffic flow by calculating the deviation of the mean vehicle velocity of all users from the free flow speed (FFS) of the observed link. The free-flow speed of a link was assumed to be equal to the speed limit of the road segment. The value of the speed limit of each link is described in the FCD data set. The mean velocity of the observed link is equal to the pre-processed value of the mean velocity of all FCD users that traversed the link in the observed time frame. The speed limit of a link is arguably the optimal speed that enables the traffic participants to traverse the observed road segment without congestions. Therefore, it is assumed that road segments that have a deviation value less than 90% of the FFS are congested and the rate of vehicles traversing it in a unit of time is low. On the other hand, road segments with a deviation value greater or equal to 90% of the FFS are experiencing a high rate of vehicles per unit of time and thus have no congestions. Table 5.10 and Figure 5.6 illustrate the overall distribution of velocity deviations on the observed network for each trajectory estimation method. The mean and median values in each time interval indicate that the overall traffic flow state of the network is within the right side range of the 90% FFS value, thus the overall network is subject to a free flow of vehicles. As visualised in Figure 5.6, the difference in the deviations of the observed methods is minimal. As each method used the same FCD mean velocity for the network links, the only difference could be observed if a method estimated a different trajectory for the same user in comparison to other methods. By traversing a different path, the user crossed a different set of road segments and

thus the deviation calculation algorithm analysed a different set of links.

Table 5.10 Case 1, 2 and 3 overall traffic flow estimation deviation results

Method	Mean [%]	STD [%]	Min [%]	1st Qu. [%]	Median [%]	3rd Qu. [%]	Max [%]
Case 1	97.95	30.57	3.33	77.61	97.86	118.02	248.68
Case 2	97.80	30.46	3.33	77.61	97.85	117.80	248.68
Case 3	96.45	30.97	3.33	77.61	96.48	116.11	248.68

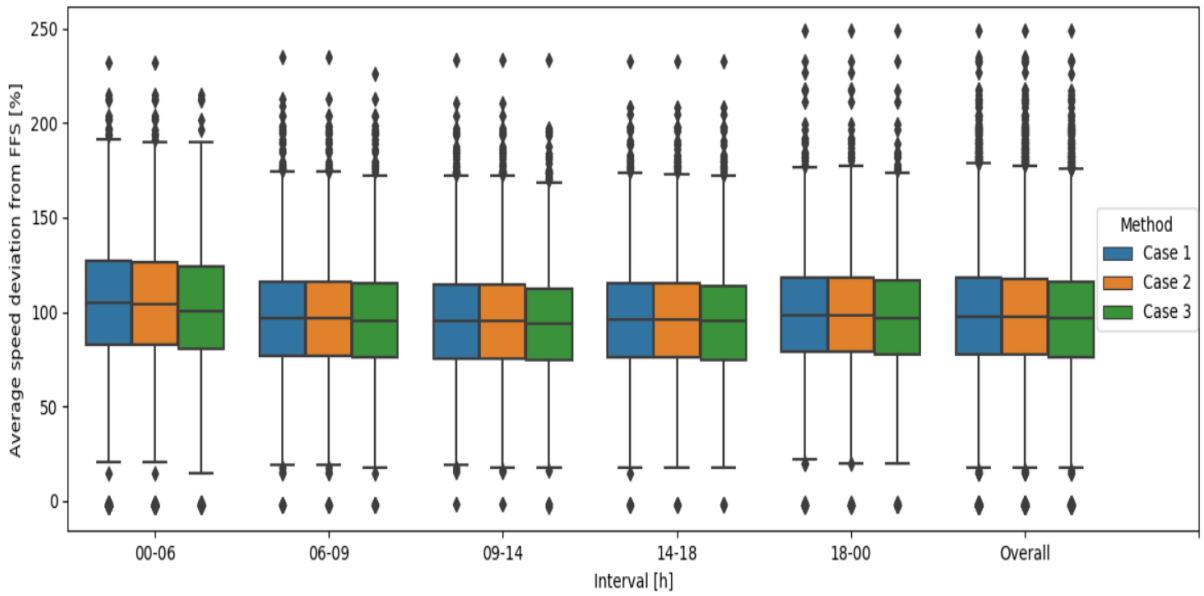


Figure 5.6: Case 1, 2 and 3 distribution of average speed deviation from FFS

6. Summary and conclusion

6.1. Summary

This Thesis considered trajectory estimation methods based on cellular and vehicular data to develop a traffic flow model of the observed network. The cellular network data is portrayed by an *OD* matrix. Each row in the OD matrix consists of a unique users trajectory metadata with an emphasis that for each user, only the source and destination cell is known. Thus the estimation of the user's trajectory marked the first step of developing a traffic flow model. The vehicular data set, generated by tracking the movement of approximately 4200 services vehicles within sixteen months across Croatia and its neighbouring regions, contains an n number of *FCD* for each road segment traversed by the tracked fleet. The tracked fleet was versatile and consisted of delivery vehicles and taxi cars. To estimate the whole cellular user trajectory, the *FCD* was processed in such a manner that the output resembled the complete road network traversed by the tracked fleet, with each road segment containing the average speed per observed time interval.

Three trajectory estimation methods were developed based on the data of the length and mean velocity of each road segment (link) of the observed road network. The general approach of estimating the user trajectories was to fetch a predefined set of alternative routes for each user and select the path with the highest spatiotemporal similarity with the ground-truth cellular OD-pair (user). Each method utilized Dijkstra's shortest path algorithm to generate a pool of alternative routes. The first two methods, *Case 1* and *Case 2*, differing in the approach of trajectory allocation by considering different parameters while computing the similarity measurement, assumed that individuals tend to follow the shortest path when travelling between locations. Therefore, when computing the shortest path algorithm, the weights of the graph were set to the road segments length. In contrast, method *Case 3* assumed that individuals tend to follow the time-wise shortest path when travelling between locations. Therefore, the weight of edges was equal to the time required to traverse the road segment. Additionally, in *Case 3*, the time values (weights) were adjusted accordingly to the link's road category. The fastest path, thus time-wise shortest, was presumed to be the path that uses road categories such as motorways, highways, etc. Hence, each road segment's required time to traverse its length was

modified so that with the increase of the theoretical speed of the road category, a decrease in the calculated time needed to traverse it was conducted.

The efficiency of the proposed methods was derived by measuring the spatial or spatiotemporal similarity of the observed methods estimated path with the observed ground-truth user's route. The *Case 1* similarity (SM) estimation algorithm iterated through the set of alternative trajectories and selected the one with the highest traversed length similarity in comparison to the observed ground-truth trajectory traversed length, not taking into account the temporal or velocity similarities. *Case 2* and *Case 3* SM estimation algorithms first generated a list of alternative paths which had a traversed length similarity for the prior method and the required time for crossing the route similarity for the latter, greater or equal to 90% of the respective ground-truth value. Such a list aimed to create a pool of paths with a high similarity level to the ground-truth trajectory. Furthermore, in contrast to *Case 1*, a spatial and temporal similarity comparison was computed for each alternative path in the newly generated list. The route with the highest overall similarity to the observed ground truth was allocated to the user. It is important to note that only trajectories with the same source and destination cell as the ground-truth OD-pair were analysed in all three methods. The results indicate that the *Case 2* method, which utilizes length as the weight in the graph and favours trajectories that have the highest overall similarity to the observed ground-truth path, is identified as the most satisfactory method. *Case 2* average mean similarity result is equal to 56.11%, while *Case 1* is at 39.85% and *Case 3* at 45.63%. Moreover, higher overall mean and median results generated in *Case 2* in contrast to *Case 3* indicate that the ground-truth path (users) in this study tended to follow the distance shortest path when travelling between two locations rather than the fastest travel time path.

Without data on when the users started their routes, modelling the density and average speed of the observed road segment in a certain time frame is subject to a high degree of uncertainty. Rather than estimating the standard traffic flow, this Thesis utilizes a method of indirectly estimating the state of the traffic flow by calculating the deviation of the mean vehicle velocity of all users from the free flow speed (FFS) of the observed link. The links *FFS* was assumed to be equal to the speed limit of the road segment. The value of the speed limit of each link is described in the *FCD* data set. The mean velocity of the road segment is equal to the preprocessed value of the mean speed of all *FCD* users that traversed the link in the observed time frame. The speed limit of a road segment is arguably the optimal speed that enables the traffic participants to cross the road segment without congestions. Therefore, it was assumed that road segments with a deviation value less than 90% of the *FFS* are congested, and the rate of vehicles traversing it in a unit of time is low. On the other hand, road segments with a deviation value greater or equal to 90% of the *FFS* are experiencing a high rate of vehicles per unit of time and thus are not congested. The results of the traffic flow model indicate that the overall traffic flow state of the network is within the right side range of the 90% *FFS* value, thus the overall

network is subject to a free flow of vehicles.

6.2. Conclusion

With suitable algorithms, it is possible to ascertain a certain spatiotemporal logic for a given area by combining cellular and vehicular data. As shown in this Thesis, trajectory estimation of sparse cellular data is feasible by reconstructing a road network using the *FCD* data. Thenceforth, generating a pool of alternative trajectories of the reconstructed network with a shortest-path algorithm and conclusively computing an algorithm that allocates the user an alternative route with the highest spatial and temporal similarity to the observed ground-truth OD (user) trajectory. Developing a model which calculates flow estimates based on the user's estimated origin-destination matrix was one of the guiding interests behind this Thesis. Due to the cellular data broad time intervals (3 hours), allocating a user to a specific road segment in a definite time interval is subject to a high degree of uncertainty. Therefore, modelling the network traffic flow without data on how many users were on a unique road segment would result in an unrealistic representation of the network traffic flow. To surmount this obstacle, future studies could provide more emphasis to the user's location in the trajectory at a definite time interval. One approach might be to compute the shortest path algorithm on the road network within 15 minute time intervals, rather than the current 3 hour time intervals. Such an approach would specify in which time interval a user's route started. With the data on when the user's path began and the mean velocity and length of all the road segments the user crossed, one could estimate the user's location at a definite time interval with high certainty.

BIBLIOGRAPHY

[1] Fillekes M. Reconstructing Trajectories from Sparse Call Detail Records. Master's thesis, University of Tartu. 2014;.

[2] Hoteit S, Secci S, Sobolevsky S, Ratti C, Pujolle G. Estimating Human Trajectories and Hotspots through Mobile Phone Data. Computer Networks. 2014;64:296–307.

[3] Huerta R, Tsimring L. Contact Tracing and Epidemics Control in Social Networks. Physical Review E. 2002;66.

[4] Vieira MR, Frias-Martinez V, Oliver N, Frias-Martinez E. Characterizing Dense Urban Areas from Mobile Phone-Call Data: Discovery and Social Dynamics. Proc IEEE SocialCom. 2010;p. 241–248.

[5] Zang H, Bolot J. Mining call and mobility data to improve paging efficiency in cellular networks. Proceedings of the 13th Annual International Conference on Mobile Computing and Networking, MOBICOM 2007, Montréal, Québec, Canada, September 9-14, 2007. 2007;.

[6] Ricciato F, Widhalm P, Pantisano F, Craglia M. Beyond the single-operator, CDR-only paradigm: An interoperable framework for mobile phone network data analyses and population density estimation. Pervasive and Mobile Computing. 2017;35:65–82.

[7] Lind A, Hadachi A, Piksarv P, Batrashev O. Spatio-temporal mobility analysis for community detection in the mobile networks using CDR data. 2017 9th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT) IEEE. 2017;p. 250–255.

[8] Lind A, Hadachi A, Piksarv P, Batrashev O. Human mobility characterization from cellular network data. Communications of the ACM. 2013;56(1):74–82.

[9] Chen G, Viana A, Fiore M, Sarraute C. Complete trajectory reconstruction from sparse mobile phone data. EPJ Data Science. 2019;.

[10] Ahas R, Silm S, Saluveer E, Järv O. Modelling Home and Work Locations of Populations Using Passive Mobile Positioning Data. Location based services and TeleCartography II: from sensor fusion to context models Springer, Berlin. 2009;p. 301–315.

[11] Schlaich J, Otterstätter T, Friedrich M. Generating Trajectories from Mobile Phone Data. Proceedings of the 89th Annual Meeting Compendium of Papers. 2010;(242).

[12] Leontiadis I, Lima A, Kwak H, Stanojevic R, Wetherall D, Papagiannaki K. From Cells to Streets: Estimating Mobile Paths with Cellular-Side Data. Proceedings of the 10th ACM International on Conference on emerging Networking Experiments and Technologies. 2014;p. 121–132.

[13] Wu C, Thai J, Yadlowsky S, Pozdnoukhov A, Bayen A. Cellpath: Fusion of Cellular and Traffic Sensor Data for Route Flow Estimation Via Convex Optimization. Transportation Research Part C: Emerging Technologies. 2015;59:111–128.

[14] Wilson RJ. Introduction to Graph Theory (4th Edition). Addison Wesley; 1996.

[15] Erdelić T. Metode za dinamičko određivanje staza na laboratorijskom sustavu s više vozila. Sveučilište u Zagrebu, Fakultet Elektrotehnike i Računarstva; 2014. Preuzeto sa: https://repozitorij.fer.unizg.hr/islandora/object/fer:784. [Accessed: June, 2021.].

[16] Diestel R. Graph theory. 173. Springer-Verlag Berlin Heidelberg; 2005.

[17] Dadić I, Kos G, Ševrović M. Teorija Prometnog Toka. 3. Fakultet Prometnih Znanosti; 2014.

[18] Rožić L, Carić T, Matulin M, Ravlić M, Fosin J, Milošević A, et al. Tehnički izvještaj rezultata eksperimentalnog razvoja projekta SORDITO. Sveučilište u Zagrebu, Fakultet prometnih Znanosti; 2016.

[19] Erdelić T, Ravlić M. SORDITO-System for Route Optimization in Dynamic Transport Environment. Promet-Traffic&Transportation. 2016;28(2):193–194.

[20] Breyer N, Gundlegård D, Rydergren C. Cellpath Routing and Route Traffic Flow Estimation Based on Cellular Network Data. Journal of Urban Technology. 2017;25(2):85–104.

LIST OF FIGURES

LIST OF TABLES

University of Zagreb
Faculty of Transport and Traffic Sciences
10000 Zagreb
Vukelićeva 4

DECLARATION OF ACADEMIC INTEGRITY AND CONSENT

I declare and confirm by my signature that this graduate thesis is an exclusive result of my own work based on my research and relies on published literature, as can be seen by my notes and references.

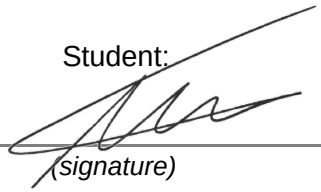
I declare that no part of the thesis is written in an illegal manner, nor is copied from unreferenced work, and does not infringe upon anyone's copyright.

I also declare that no part of the thesis was used for any other work in any other higher education, scientific or educational institution.

I hereby confirm and give my consent for the publication of my graduate thesis titled **Origin-Destination Flow and Traffic Parameter Estimation Based on Cellular Network Data and Vehicle Movement Historical Records** on the website and the repository of the Faculty of Transport and Traffic Sciences and the Digital Academic Repository (DAR) at the National and University Library in Zagreb.

In Zagreb, 02 September 2021

Student:



(signature)