



**Politecnico
di Torino**

Politecnico di Torino

Master degree in Civil Engineering

**Analysis of Vehicle Travel Times and Traffic
Congestion Through Map Matching of Floating
Car Data: An Application to the Turin Road
Network**

Candidate:

Petralia Davide

Supervisor:

Prof. Marco Diana

Academic Year 2024/2025

Contents

1	Introduction	3
1.1	Thesis objectives	3
1.2	Background	3
1.2.1	Characteristics of traffic stream	4
1.2.2	Externalities	5
1.3	Data Analysis	6
1.3.1	Road Graph	6
2	State of the Art	7
2.1	Map Matching	7
2.1.1	Geometric approach	8
2.1.2	Topological approach	9
2.1.3	Probabilistic approach	9
2.1.4	Advanced map-matching techniques	9
2.1.5	Errors associated with Map Matching processes	10
2.2	Observed travel time estimation using FCD with different methodologies	11
2.2.1	Data-driven methods	11
2.2.2	Model-based methods	16
2.2.3	Hybrid strategies	17
2.3	Travel time estimation under free flow conditions	18
2.4	Delay estimation in aggregate fashion	18
2.5	Indicators of congestion	19
2.5.1	Key Performance Indicator based on highly disaggregate level analysis using travel time	19
2.5.2	Performance Indicator based on qualitative analysis of average speeds	19
2.6	Contextualizing the Proposed Methodology within Current Research	20
3	Experimental setting	21
3.1	Study context: Turin	21
3.2	Mobility and road infrastructures in the study area	23
3.3	Floating car data used in the study area	28
3.3.1	Overview of Commercially Available Floating Car Data	28
3.3.2	Historical Car Data (HCD)	29
3.3.3	Floating Car Data (FCD)	33
3.3.4	Differences between HCD and FCD	35
3.4	Graph of Turin	35
4	Methodology, part 1: HCD data processing and map matching	38
4.1	HCD data pre-processing	38
4.2	HCD data matching process to the graph	40

4.2.1	Data cleaning and selection	41
4.2.2	Online map matching with ORS	42
4.2.3	Snapping on QGIS	43
4.2.4	Association of arc's corresponding attributes	44
4.3	HCD data cleaning of matched points	45
4.4	Computation of distances on the graph	47
4.5	Filtering out HCD observations with service stops	48
4.6	Statistics on the number of HCD observations on each arc	49
4.6.1	Distribution of the average number of observations per device	50
4.6.2	Distribution of the visits	51
5	Methodology, part 2: travel times derivation	53
5.1	Free-flow travel time computation	53
5.1.1	Free flow travel time using previous information from the graph	53
5.1.2	Free flow travel time based on HCD observations [19]	54
5.1.3	Selection of the most appropriate free flow travel time and comparison between the two different methodologies	55
5.2	Observed travel time computation from the FCD dataset	59
5.2.1	FCD dataset manipulation and selection of observed days	59
5.2.2	Vehicle level travel time derivation	62
5.3	Statistics on the number of FCD observations on each arc	64
5.3.1	Distribution of the visits	65
6	Results	67
6.1	Increase in travel times by vehicle and by arc	67
6.2	Comparison of data availability during peak hours vs entire-day data	69
6.3	Increase in travel times at arc level	69
6.3.1	Increase in travel times at arc level during peak hours	70
6.3.2	Increase in travel times at arc level during all day	75
6.4	Speed variation at arc level	80
6.4.1	Speed variation at arc level during peak hours	80
6.4.2	Speed variation at arc level during all day	83
6.4.3	Case study: Turin streets where the average speed is larger than 30 km/h	86
6.5	Increase of arc travel times at zonal level	88
6.5.1	Increase of arc travel times at zonal level during peak hours	91
6.5.2	Increase of arc travel times at zonal level during entire day	92
6.6	Increase of vehicular travel times at zonal level	92
6.6.1	Increase of vehicular travel times at zonal level during peak hours	93
6.6.2	Increase of vehicular travel times at zonal level during the whole day	94
6.7	Speed measure for O/D relations between the zones introduced in 6.5 <i>Increase of arc travel times at zonal level</i>	98
6.8	Increase in travel times for different travellers' categories	101
7	Conclusions	104

APPENDIX **108**

1	Map Matching code	108
2	Counting deviceId-Arco Occurrences in R	110
3	Computation of average number of visits on each arc	111

4	Computation of distances on the graph	112
5	Computation of points' relative position on the arcs	113
6	Calculation of mean travel time for each deviceId on each arc	115
7	Different categories of Map Matching methods	117
8	Association of deviceId to correct Trip	117

List of Figures

1	Map Matching [16]	7
2	Map Matching strategies, from [12]	8
3	FCD observations after map matching, from [17]	10
4	Distance-time proportion travel time estimation [18]	13
5	Link travel time using the spatial and temporal connectivity [18]	15
6	Plan view of Turin, can appreciate the grid in the central area	21
7	Extension of study area with municipalities, from Qgis elaboration	22
8	Extension of study area with road graph, from Qgis elaboration	23
9	Extension of Metropolitan City of Turin [6]	24
10	Motorization Rate of Metropolitan City of Turin [6]	25
11	Main attractor and generator poles [6]	26
12	Influence area of Turin-study [6]	27
13	Influence area of Turin-work [6]	28
14	Graph view on Qgis	35
15	Three dimension contingency table	39
16	Sampled strata	40
17	Aggregated HCD files, per month, per hourly interval, sorted by month	40
18	Maintained points (violet) and deleted points (green)	41
19	.csv files with only maintained points after selection process	42
20	Output of Python script, with corrected coordinates	43
21	Results of snapping process	44
22	Matched points after map matching process, from "matched_jan-apr.csv"	45
23	Output after counting script applied on R, from "matched_jan-apr.csv"	45
24	Cronological movements of each vehicle, from "merged_matched_processed.xlsx"	46
25	Difference column with sequence of movements of Device.Id 2822159	47
26	Output file after execution of Python code to calculate distances among points	47
27	Distribution of delta_time from 'distances.csv'	49
28	Arcs statistics	50
29	Distribution of 'average' column, from 'arco_stats.xlsx'	51
30	Number of deviceId visiting arcs distribution	52
31	List of points to be processed to determine free flow speed	54
32	Output of elaboration of free flow speed with the two different methodologies, from the file "points_processed.csv"	55
33	Free flow travel time of each arc, from 'archi_fftt.xlsx'	57
34	Relationship between 'delta_speed' and 'arc_length', from 'archi_fftt.xlsx'	58
35	Relationship between 'delta_speed' and 'arc_length'	59
36	Three points on the same arc	62

37	Notation on the arc considering points 1 and 2	63
38	Notation on the arc considering points 2 and 3	63
39	Result of described process above, from 'distances_FCD.xlsx'	64
40	Statistics on arcs, from 'arco_stats_FCD.xlsx'	65
41	Number of deviceId visiting arcs distribution	66
42	Structure of 'results_vehicle level.xlsx'	67
43	Structure of 'results_vehicle level.xlsx', with percentage of time wasted for each vehicle	68
44	Distribution of column 'Wasted_time_vehicle', from 'results_vehicle level.xlsx'	68
45	Distribution of column 'Wasted_percentage', from 'results_vehicle level.xlsx'	69
46	Structure of 'results_arc level_peak.xlsx'	70
47	Structure of 'results_arc level_peak.xlsx'	70
48	Relationship between arc_length and average time wasted on each arc	71
49	Visualization of arc level congestion, from Qgis layout	72
50	Visualization of arc level congestion in the central area of Turin, from Qgis layout	73
51	Hourly flow distribution measured on February, 13th, 2019, from [14]	74
52	Structure of 'results_arc level.xlsx'	76
53	Structure of 'results_arc level.xlsx' with new column 'Average_wasted_time_arc'	76
54	Visualization of arc level congestion, from Qgis layout	77
55	Visualization of arc level congestion in the central area of Turin, from Qgis layout	78
56	'results_arc level.xlsx', with calculus of speeds and their difference	80
57	Real speed deviation with respect to free flow speed during peak hours, expressed in percentage	81
58	Real speed deviation with respect to free flow speed during peak hours, expressed in percentage, zoom on Turin urban centre	82
59	Real speed deviation with respect to free flow speed, expressed in percentage	84
60	Real speed deviation with respect to free flow speed, expressed in percentage, zoom on Turin urban centre	85
61	Average speed on each arc, zoom on Turin	88
62	Zoning inside Turin	89
63	Zoning of the study area	90
64	Unit average time wasted for each zone during peak hours	91
65	Unit average time wasted for each zone considering all day	92
66	Unit average time wasted for each zone during peak hours, expressed in minute per vehicle	93
67	Total time wasted for each zone during peak hours, expressed in hours	94
68	Unit average time wasted for each zone, expressed in minute per vehicle per day	95
69	Total time wasted for each zone during all day, expressed in hours	96
70	Structure of 'confronto_zonadisaggregated.xlsx'	96
71	Graphic visualization of ratio	97
72	Classification of the accessibility measures reviewed from the literature, from [4].	98
73	Origin/Destination matrix, from 'origin_destination.xlsx'	98
74	Average speed matrix, from 'origin_destination.xlsx'	99
75	Desire lines of Turin	100
76	Desire lines towards Turin	101
77	Pivot table with info on type	102
78	Pivot table enriched with info on gender	102
79	Five entry table	103

80 Review of Map Matching methods, from [12] 117

List of Tables

1	Notation used in the paper	12
2	Percentage of free flow speed corresponding to different LOS and assumed traffic state	20
3	HCD Metadata	31
4	HCD Trip Details Metadata	33
5	FCD Metadata	34
6	Graph Metadata	36
7	Classified List of Significant Dates in 2019	39
8	Selected day for FCD data manipulation	60
9	10 most congested arcs in Turin during peak hours	75
10	10 most congested arcs in Turin	79
11	10 highest delta_speed arcs in Turin during peak hours	83
12	10 highest delta_speed arcs in Turin	86

Abstract (Italiano)

Al giorno d'oggi, le persone non viaggiano semplicemente per il piacere di spostarsi, ma perché devono svolgere diverse attività, spesso distribuite in aree differenti. Se non ci fossero attività da svolgere, o se tutte fossero concentrate in un unico luogo, le persone non avrebbero necessità di spostarsi. Questo concetto rappresenta il punto di partenza del nostro studio, che si propone di analizzare la congestione lungo le infrastrutture urbane.

Delineato il problema, l'obiettivo di questa ricerca è analizzare i tempi di percorrenza per identificare i tratti stradali più critici all'interno della Città Metropolitana di Torino. Questa analisi è pensata per supportare la gestione del traffico, sia nelle decisioni a lungo termine che in quelle operative a breve termine.

Per raggiungere questo obiettivo, è stata sviluppata una metodologia strutturata che prevede diversi passaggi fondamentali nell'analisi su larga scala dei floating car data e degli historical car data. Questi dati, raccolti nell'arco di un anno dall'azienda TIM, sono stati archiviati in numerosi file .csv contenenti informazioni essenziali sugli spostamenti dei veicoli in un'ampia area che comprende l'intera città di Torino, la sua cintura urbana e i comuni limitrofi. Grazie a questi dati, che includono le coordinate dei veicoli, i timestamp e la velocità, è stato possibile analizzare sia i tempi di percorrenza effettivi che i tempi di percorrenza in condizioni di flusso libero lungo i diversi tratti stradali.

I passaggi principali della metodologia comprendono la selezione dei dati, filtrandoli in base alla rete stradale disponibile, seguita dal map matching, ovvero l'associazione di ciascuna posizione registrata del veicolo al corretto segmento stradale. Una volta completata questa mappatura, è stata condotta un'analisi della congestione per valutare le dinamiche del traffico e le inefficienze della rete viaria.

I risultati dello studio forniscono informazioni a tre livelli di aggregazione. A livello più disaggregato, è stato misurato il tempo perso da ciascun veicolo a causa della congestione. A livello di singolo tratto stradale, è stato quantificato il tempo perso su ciascuna sezione specifica della rete. A livello zonale, è stato sviluppato un metodo per valutare i ritardi dovuti alla congestione nelle diverse aree di Torino, con una suddivisione basata sui quartieri della città e sui confini amministrativi dei comuni limitrofi.

Infine, i risultati di questo studio hanno il potenziale per diventare uno strumento utile per le amministrazioni pubbliche e i pianificatori dei trasporti, supportando il processo decisionale nella gestione della congestione stradale e del traffico.

Abstract

Nowadays, people do not travel merely for the pleasure of moving but because they need to carry out various activities, which are often spread across different areas. If there were no activities to perform, or if all activities were concentrated in a single location, people would not need to travel. This premise serves as the foundation for our study, which aims to analyse the congestion along urban infrastructures.

Having outlined the issue, the objective of this research is to analyse travel times to identify the most critical road segments within the Metropolitan City of Turin. This analysis is intended to support traffic management efforts, both in long-term planning and short-term operational decisions.

To achieve this, a structured methodology was developed, involving several key steps in the analysis of large-scale floating car data and historical car data. These datasets, collected over a one-year period by the company TIM, were stored in numerous .csv files containing essential information on vehicle movements across an extensive area, encompassing the entire city of Turin along with its urban belt and surrounding towns. Using this data, which includes vehicle coordinates, timestamps, and speeds, both observed travel times and free-flow travel times along different road segments were analysed.

The main steps of the methodology included data selection, where raw data was filtered based on the available road network, followed by map matching, where each recorded vehicle position was associated with the correct road segment. Once this mapping was completed, congestion analysis was conducted to assess traffic patterns and inefficiencies.

The analysis provide insights at three levels of aggregation. At the most disaggregated level, the time wasted by each individual vehicle due to congestion was measured. At the road segment level, the study quantified the time lost on each specific section of the network. At the zonal level, a method was developed to evaluate congestion-related delays within different areas of Turin, where zoning followed the city's neighbourhoods and the administrative boundaries of surrounding municipalities.

The results of this study have the potential to serve as a valuable tool for public administrators and transport planners, aiding in decision-making processes related to traffic congestion management.

1 Introduction

Nowadays, the rise of new technologies is helping us in managing a lot of aspects of our daily lives, among which of course we have the traffic management system. So far, we mostly used fixed sensors to measure the characteristics of traffic streams and to take decisions based on that, but as we know those devices are pretty limited to the specific location where they are placed for example. So, with the introduction of the Floating Car Data we have the possibility to record the position of each vehicle in real time and even taking real time decisions. The structure of this thesis is made in such a way that we can present the usefulness of those kind of data in the individuation of the arcs where most time is wasted due to congestion in Turin. In the following sections, we introduce some technical terms that could be useful in the better understanding of the topic faced during the thesis. In the second chapter a literature review of the most common method used to determine the travel time on each arc at aggregate and disaggregate level is presented, then methods to determine time waste during congestion. In the third one the dataset we used to implement our analysis is presented. In chapter four, the methodology is described, thus the process that led us from the data pre processing to the final data ready to be analysed. Here we also present the Map Matching process we carried out and the following data processing. In the fifth chapter, the analysis of travel times is carried out while in the sixth the results of the analysis are commented and graphically visualized.

1.1 Thesis objectives

Our studies are based on the knowledge of the so-called Floating Car Data (FCD) and Historical Car Data (HCD) of the private vehicles around Turin. They are recorded, by means of mobile devices, such as smartphones, and they basically report information on the position of the vehicle along the time; we can say these are raw data, and we will process them in order to get the main information, thus, speed and travel time on each arc.

The purpose is not only to process these data and get these details, but to make them interpretable. We will evaluate the travel time and speed of the same arcs under free flow conditions by means of HCD, in order to successively understand how much time is wasted due to congestion and what are the roads in which most of the time is wasted in Turin, by means of FCD. This is slightly different with respect to determine the most congested roads, in fact, the congestion is strictly related with the capacity of the infrastructures, and for small minor roads, even if few vehicles pass, we could reach congestion easily, instead, on the other hand, it would be more interesting to understand how much time is collectively wasted with respect to a standard travel time under free flow condition, for example to support transport policy makers wishing to prioritize interventions based on the loss of social welfare.

1.2 Background

We would like to give some technical definitions and introduce concepts that will be faced in the next chapters of the thesis to make the reader fully understand all the topics, most of those are taken from the Highway Capacity Manual [3].

1.2.1 Characteristics of traffic stream

First of all, traffic flow can be divided into two types. The first one is uninterrupted flow, which is a flow regulated only by vehicle-vehicle and vehicle-roadway interactions, meant as the geometric and environmental characteristics of the latter. The second one is interrupted flow, which is a condition in which the free movement of one vehicle is influenced not only by other vehicles or by the geometric characteristics of the road, but also by external factors, like the presence of intersection on the road, traffic signals, traffic signs, yield signs and so on. Uninterrupted and interrupted, however, describe the facility, not the flow conditions.

Knowing this information is important to appropriately choose the correct analysis procedure to estimate the capacity of the infrastructure, which is a fundamental concept to be catch. The capacity of a facility is the maximum hourly rate at which the vehicles are expected to traverse a determined section of a lane or roadway under standard conditions.

We can state that capacity is a fixed characteristic of a facility, instead, to quantify the amount of traffic passing a section, we use the concept of:

- *Volume*: total number of vehicles that pass over a given section during a given time interval; it can be expressed as annual, daily, hourly or subhourly;
- *Flow rate*: the equivalent hourly rate at which vehicles pass over a given section during a given time interval less than 1 hour, usually is 15 min.

The difference between them is important. The volume in fact is the total number of vehicles observed in a given interval of time, the flow rate is expressed as a fraction of the volume, for example, if you observe 100 veh/15 min, you can assume a volume of 400 veh/h, but each 15 min, the flow can be different, so this distinction is important to understand if the flow rate is exceeding the capacity in that 15 minutes. This led us to define the so-called peak-hour factor (PHF), thus the ratio between the hourly volume and the highest flow rate within 1 hour.

$$PHF = \frac{V}{4 * V_{15}} \quad (1)$$

Where:

- V is the hourly volume;
- V_{15} is the highest flow rate in 15 minutes;

The knowledge of the PHF is not needed to determine peak flow rates when you have traffic counts, but if you don't have too much information, it is quite good to know it (usually it varies around 90%).

All this, to highlight the fact that when the volume is near to the capacity, we are reaching congestion, and this concept will be the basis for the analysis we will carry out in this thesis.

Although traffic volumes are good indicators, we can exploit also other indicators to understand if a segment is or not congested.

The *speed*, for example, is another important factor to be analysed. Literally, the speed is defined as the distance travelled in a unit of time, and by means of this we can estimate the travel time to travel a certain segment. For each segment is possible to define the free-flow speed, thus that speed adopted by the users, which is only influenced by the geometric characteristics of the roadway, so we can say the "optimal" speed; this has to be compared with the observed speed to understand the level of congestion.

Another factor is the *density* of the segment, thus the number of vehicles occupying a given length of the roadway in a certain instant of time. Those are the characteristics of the traffic flow which will be considered for our analysis, but considering only those, could be too simplistic, because, since we are prevailing in an urban context, we are in interrupted-flow conditions, so the most significant sources influencing the flow, are the traffic signals, so their presence will be considered as well.

1.2.2 Externalities

In the introduction, we mentioned some data on the effect of traffic on the whole society. We can basically refer to them in general as externalities [5]. In economics, an externality is a cost or benefit incurred or received by a third party who has no control over the factors that created that cost or benefit. The basic characteristic is that the agent causing the damage, usually is not paying any indemnity to those affected by the cost imposed, or he is not gaining any compensation for the generated benefit.

In the transport sector, there are mainly 5 categories of externalities, considering the viewpoint of the supplier, user and the collectivity:

- Congestion;
- Safety (accidents);
- Greenhouse gases emissions (GHG);
- Pollution;
- Noise;

We already introduced the congestion beforehand. The safety is referred to the increasing number of accidents and death due to the continuous expansion of the roads. While the difference between GHG emissions and pollutants emissions is slight but important, especially on a policy viewpoint. The GHG are the results of a process in which hydrocarbons are burnt, so if you know the quantity of fuel, with an equation you could estimate the quantity of CO₂ emitted, it's pretty simple.

On the other hand, concerning pollutants, their presence is due to the several processes that happen during the internal combustion process, e.g. atoms of carbon are not completely oxidized, therefore is coming out carbon monoxide (CO); when the combustion process does not happen correctly, there is the production of hydrocarbons, which are volatile, there are the so-called VOC (volatile organic compounds); the oxygen in the air could be recombined into ozone (O₃), and since in the engine does not enter pure oxygen but the air, which is in turn composed of nitrogen, we can have oxides of nitrogen; sulphur is added to the fuel for technical reasons, and is going to be combined with oxygen to form sulphur dioxide, which is again a poison; but right now we are missing the worst one, particulate matters, as PM₁₀, PM_{2,5}, where the number is giving the dimension of the particulate expressed in micrometres, the smaller is the worst is the effect on the health, this is coming out not only from the combustion of the engine but also from the operation of the vehicle (even electric vehicle emit particulate matter).

This distinction is important because we understand that we can control the pollutants by improving vehicle efficiency (better emission standard class) but the emission of GHG itself since is a chemical reaction cannot be controlled, it only depends on the kind of fuel. Noise is a very

harmful factor for the health of the people and hinders the smooth realization of their everyday activities. The main impacts from noise include hearing difficulties, frustration, radical changes in the behaviour of people, communication complications, fatigue, and difficulty in sleeping. These impacts can cause serious issues on human organization.

1.3 Data Analysis

We have raw data initially, and we will work on them to make them useful. Substantially, we want to transform data into information. As a general, when we have raw data, the first thing to do is to clean them, to eliminate outliers for example, then there is the process of analysis and at the end the final interpretation with the results.

At the level of analysis, we can have two alternatives: aggregate or disaggregate.

At aggregate level, raw data are gathered, maybe they are classified and then the analysis is carried out with respect to each class, to have a more general view.

For example, remaining on the thesis topic, we have for example one arc with a certain travel time during free flow condition. Assuming that there is a given number of vehicles travelling there within a certain interval of time, if we take their average speed and therefore their average travel time, then we can compare the latter with free-flow travel time and therefore assess the time lost due to congestion. This is an aggregate analysis at a level of the arc. An even more aggregate analysis involves the consideration of the whole network. There is a lot of macro-simulation software, in the field of transportation, that, starting from a series of input data, gives the flow on each arc and the total travel time as output. So by comparing the latter with a value taken under free flow conditions, we can estimate the total delay on the network.

At the disaggregate level, the analysis is carried out, considering the characteristics of each vehicle, in order to have a more detailed information, nevertheless this is a process which is “data hungry”, thus it needs a lot of data. Here, since we have a lot of data, we will carry out a disaggregate analysis, considering each vehicle on each arc with its own speed and travel time. This is consistent with a micro-simulation approach that is normally employed to study in details only limited portions of a road network.

1.3.1 Road Graph

The road infrastructure is represented by means of a set of edges and nodes which constitute the network graph. Each arc represents a road in the Network and the graph represents the topology rather than the real physical structure. Each vertex represents not only the physical intersections, but also points in which there is a change of speed, a change in the number, we can say a change in the physical or functional characteristics of the road.

Moreover, when we associate different attributes to each arc, such as the travel time, the speed, the number of lanes, the capacity, we can then analyse the graph and compute for example the shortest path between 2 points, the fastest path, or solve some vehicle routing problems (Travelling Salesman problem, Chinese Postman problem and so on). This to state that the representation of the network, it's not only used to represent the 'supply' in the transportation system, but can also be used to make analysis and prediction supporting the policy maker. At the same time, we cannot often represent all the roads present in the network on the graph, but only the main ones. The impact of this simplification has to be taken into account especially when we, for example, have to distribute the traffic flow along the network, because some arcs can result more congested than they effectively are, since roads are not represented and thus not loaded with their traffic.

2 State of the Art

In this chapter, we will examine current methodologies for measuring time wasted during congestion, using both disaggregate and aggregate approaches. First, we will review the Map Matching process as it is presented in literature, with its different typologies and then features of FCD (Floating Car Data) to identify potential challenges and limitations. At the end, most of the literature on low-frequency FCD has focused on link travel time estimation. Proposed methods are typically divided into two groups, one related more with the real travel time estimation, which is observed by means of FCD and one more related with the travel time under free flow speed conditions.

2.1 Map Matching

Map matching is a critical process in geographic information systems (GIS) and transportation analysis, aimed at aligning raw GPS trajectories with a digital road network. It consists of the integration of 2 data typologies: localization data and digital maps. In fact, as we know, GPS trajectories are associated with two types of errors, measurement errors, i.e. the recorded location can deviate from the true location due to noise from several sources; sampling errors, which refer to lost information between the recorded points; ignoring those errors can lead us to false analysis, so map matching is an approach to minimize these errors by matching the recorded points to the graph of the network. Over the years, researchers have proposed numerous methodologies to address the challenges posed by the inherent noise and imprecision of GPS data.

An example of Map Matching is reported in Figure 1.

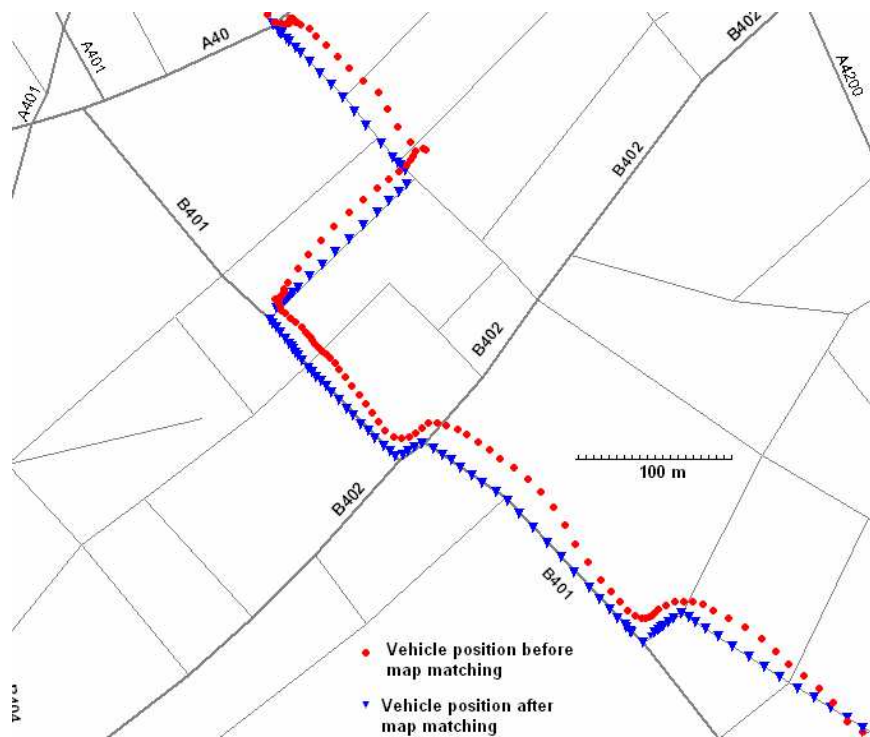


Figure 1: Map Matching [16]

Several approaches have been developed, each with distinct strengths and weaknesses. More in detail, in [12], at the paragraph 4, is reported a table which we will report in the Appendix

[7], providing an overview of the different categorizations of Map Matching methods. The most significant categorization is related with Use of Road Network and Trajectory details, divided into four groups: geometric approach, topological approach, probabilistic approach and advanced probabilistic or machine learning-based approaches. In this section, we will review the most influential works within each category. To sum up:

Category	Definition	Used Parameters	Papers/ Policies
Geometric Algorithms	Using defined geometry it derives the requisite information for matching the edge segments	Euclidean, Hausdorff, Frechet', Area, overlap, etc. function used for a curve to curve, point to curve or point to point matching	Geometrical (Abdallah et al., 2011)
Topological Algorithms	These methods use information from defined map topological knowledge	Node-degree, Valid traffic directions, speed, prohibited manoeuvres, distance, connected segments, etc.	Topological (Greenfeld, 2002) (Liu et al., 2017)(Schwertfeger & Yu, 2016)(Velaga et al., 2009)
Probabilistic Algorithms	Around the points, estimation of the confidence value is achieved through probabilities calculation	Use the confidence region around the trajectory points with closeness, heading, and connectivity to estimate the best matching segment.	Probabilistic (M. A. Qudus et al., 2007)(Jagadeesh & Srikanthan, 2015)

Figure 2: Map Matching strategies, from [12]

2.1.1 Geometric approach

A geometric map-matching algorithm relies solely on geometric information, such as the "shape" of line segments (i.e., the road centerlines that define the road network, rather than their connectivity). Various types of geometric map-matching algorithms have been developed, including point-to-point matching, point-to-curve matching, and curve-to-curve matching. These algorithms are versatile and can be applied to positioning data of any frequency, as they only require position fixes (x- and y-coordinates) and a base road network map as input. However, the accuracy of geometric map-matching algorithms is relatively low, with correct link identification rates ranging between 80% and 85%.

A review of different algorithms in this context is presented by [21]. The simplest one is the *simple search problem*, where the purpose becomes matching the given point to the "closest" node or shape point in the network. Several data structures and algorithms are available for identifying all points "near" a given point, a process often referred to as a range query. Once the nearby points are identified, calculating the distance to each node or shape point within a "reasonable" distance is straightforward (regardless of the distance metric used), and the closest point is selected. While this approach is relatively easy to implement and computationally efficient, it presents several practical challenges. One significant issue is its heavy reliance on how shape points are defined and utilized within the network, which can greatly influence the results.

Map matching can also be approached as a problem of *statistical estimation*. In this framework, a sequence of points is considered, constrained to lie on the network, and an attempt is made to fit a curve to them. This approach has been explored in numerous studies and is particularly appealing due to its elegance, especially when the "physics of motion" is modeled with simplicity—such as movement being restricted to a straight line. However, in most real-world applications, the physics of motion is heavily influenced or constrained by the network itself, making it challenging to accurately model and apply this approach effectively.

2.1.2 Topological approach

Since maps are usually represented as graphs, topological algorithms tend to preserve continuity in the matching, avoiding frequent errors. This approach considers both spatial details and route topology relationships through locations and path ties to candidates as the considerations of decision. The topology Map Matching approach steps can be divided into two sections:

1. Initial matching;
2. Subsequent matching.

Initial matching is the first step in the process, determining the section of the path to be matched through geometrical analysis. In the second step, subsequent matching selects candidate segments based on the outcomes of the initial matching. This step involves both road network analysis and further geometrical evaluation. Candidate links are then assigned a value, which is calculated based on three key components:

1. Alignment Angle in Direct Link and “Axis” Across Next Locations;
2. Relation Link and Orientation of Successive Points;
3. Closeness of Direct Link to Positioning Point;

After identifying the candidate segments, the best-suited candidate is selected through topological and geometric-based calculations. The candidate with the highest score is designated as the vehicle’s true position. While topological map-matching methods incorporate geometric strategies, they often struggle in more complex scenarios, such as handling low-precision data, large-scale positional datasets, or low sampling rates, where results may be unsatisfactory. Nonetheless, topological weight-based map-matching methods are noted for their simplicity of implementation and exhibit superior performance in terms of speed and accuracy.

2.1.3 Probabilistic approach

The probabilistic algorithm involves defining an elliptical or rectangular confidence region around a position fix obtained from a navigation sensor. Probabilistic map matching algorithms have been proposed to take advantage of statistical models such as Kalman filter, particle filters and Hidden Markov Model.

2.1.4 Advanced map-matching techniques

Advanced map-matching algorithms improve vehicle positioning by using refined techniques such as Kalman and Extended Kalman Filters, probabilistic models, and fuzzy logic. These methods address challenges in navigation, particularly in dense urban areas where GPS signals may be obstructed. Kalman Filters, for instance, help re-estimate positions by minimizing errors in relation to road network data, effectively reducing cross-track errors. Particle Filters apply probabilistic constraints to vehicle positions, enhancing accuracy even in urban environments. Some algorithms model the vehicle path as constrained road segments, allowing accurate positioning with fewer satellite signals. At intersections, these techniques integrate probabilistic models to select the correct road segment. Additionally, fuzzy logic-based models evaluate potential road segments using criteria like proximity, heading, and connectivity, refining the map-matching process. These

advanced approaches collectively enhance navigation accuracy by accounting for positioning errors and the complex topology of urban road networks.

The various map matching approaches offer distinct advantages depending on the application, but they are not without limitations. The accuracy of the results can be influenced by several factors, including the quality of GPS data and the complexity of the road network. These inaccuracies can lead to significant errors in the matching process, which are crucial to analyze in order to understand the main challenges of this technique.

2.1.5 Errors associated with Map Matching processes

We saw the different approaches with their strengths and weaknesses. However, as it is perfectly explained in [17], due to bandwidth limitations, the sampling frequency is often too low, leading to some problems which will be highlighted to have a clear context. A drawback of FCD is that probes are not in general generated at the start and end points of routes¹ and links.

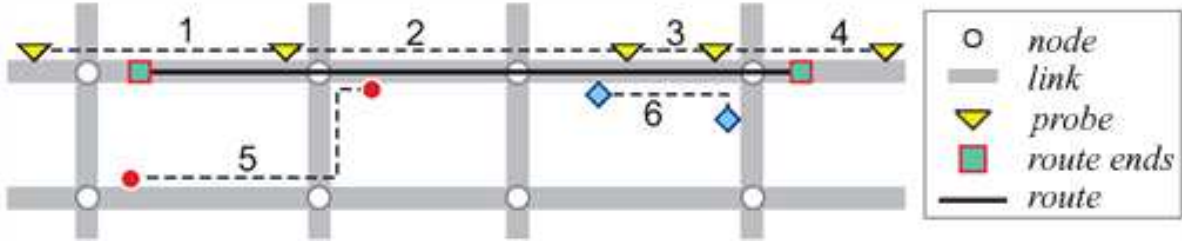


Figure 3: FCD observations after map matching, from [17]

In Figure 3, yellow triangles, red circles, and blue rhombuses represent three distinct vehicles whose movements have been recorded. A number is associated with each trajectory connecting the pairs of vehicles. There are several potential sources of bias when using FCD for travel time estimation:

- 1) **Incomplete Coverage of Route:** A FCD observation may cover only a fraction of the route, as illustrated in Figure 3 (see yellow triangles). Considering each observation in isolation, the travel time of the vehicle on other parts of the route is unknown and needs to be inferred; incorrect inference will lead to bias.
- 2) **Time-Based Sampling:** The sampling of vehicle trajectories is often triggered by time, which means that the distance between consecutive probes is short if the travel speed is low and vice versa. Furthermore, there will be more FCD observations in parts where the speed is lower. These factors need to be considered in the travel time estimation to avoid bias.
- 3) As the opposite of bias 1, an FCD observation may extend partially over the route and partially over adjacent links, as illustrated by observations 1, 4-6 in Figure 3. The precise allocation of travel time T between the route and the adjacent links is unknown. Incorrect allocation means that the travel speed on the adjacent links will spill over and bias the estimated route travel time.
- 4) **Non-Uniform Coverage of Route:** The coverage of FCD observations may vary along the route since vehicles may enter and leave via side streets (compare yellow triangles, blue

¹A route is defined as a sequence of links from an origin point to a destination point.

rhombuses and red circles Figure 3). If the travel speed is not homogenous on the route, the estimated route travel time will be biased if all observations are weighed equally.

- 5) Unknown Route Entry Time: Route travel times may be estimated for certain time intervals (e.g. 15-minute intervals) based on the time that a vehicle passes the start point of the route. For FCD observations covering the start point (e.g., observation 1 in Figure 3), the exact entry time is not known. For observations not covering the start point (e.g., observations 2-6 in Figure 3), the entry time is hypothetical and needs to be constructed. If the inference is inaccurate, the travel time profile across intervals may be biased.

2.2 Observed travel time estimation using FCD with different methodologies

Travel time estimation is a critical component of intelligent transportation systems, enabling applications such as traffic management, route optimization, and real-time navigation services. Over the years, a wide range of methodologies has been developed to estimate travel times with increasing accuracy and reliability. These approaches leverage various data sources, including GPS trajectories, historical traffic data, and sensor-based measurements, each offering distinct advantages and challenges. A review of different approaches is presented in [10]. They are classified according to the data exploited and the method applied. Travel time estimation is articulated through a series of steps involving three main phases:

- FCD cleaning and map matching;
- Link-based travel time estimation, known as travel time allocation;
- Route travel time estimation;

Of course we will go more into detail of the second step, which usually is structured in two main phases, the first related with travel time allocation for each vehicle and the final one is the processing of all those travel time by means of simple instruments as the average, weighted average or by means of more sophisticated methods which will be presented below. We can distinguish mainly data-driven methods, model-based methods or hybrid strategies. We remind that this is the part on which our methodology will rely to meet the purpose already declared in section 1.1.

2.2.1 Data-driven methods

Data-driven methods are approaches that rely primarily on large volumes of data to extract insights, recognize patterns, and make predictions. Unlike traditional model-based techniques, these methods do not rely on predefined rules or physical models but instead use statistical and machine learning algorithms to learn directly from the data. For example, as it is described in [15], the reference to measure the time wasted in congestion can be the travel time on the arcs; this step involves a detailed process to ensure the calculated travel time accurately reflects the duration taken to traverse a segment while distinguishing between various factors that may influence the overall time. To begin with, the GPS data is analyzed to identify the first recorded point within the buffer zone surrounding the starting node of the road segment and the last recorded point within the buffer zone of the ending node. They both mark the entry and exit time of the vehicle from the arc respectively and the total travel time is the difference between the two timestamps.

This total travel time inherently includes any intermediate stops made by the vehicle along the segment, such stops can occur for two primary reasons: traffic-related conditions such as congestion, traffic lights, or yielding or operational service stops, such as deliveries or loading and unloading activities. To refine the travel time analysis, the methodology makes a clear distinction between these two categories of stops. Stops lasting less than or equal to 120 seconds are attributed to traffic conditions, as they typically reflect temporary delays caused by the road congestion. Conversely, stops exceeding 120 seconds are classified as service stops, which are generally longer and operational in nature. This threshold of 120 seconds is chosen as a compromise based on research, despite it is acknowledged that some misclassification may occur for instance because of shorter service stops or exceptionally long congestion related delays. Once the service stops are identified, their cumulative duration, denoted as T_GPS_{-ss} , is subtracted from the total travel time to derive the net travel time along the segment. This net travel time excludes operational delays, providing a clearer picture of the time required to traverse the segment under prevailing traffic conditions. At this point, focusing on a certain time interval (8:00-8:59 am) the travel time of each vehicle has been compared with an estimated travel time under ideal conditions, better known as free flow condition. In this way, with the following definition of some indicators, it has been determined the extent to which the vehicles wasted time due to traffic conditions.

The approach we have explored adopts a more disaggregated perspective, focusing on individual components or detailed elements. However, shifting towards a more aggregated analysis, a link-based method is proposed in [17] to estimate the final route mean travel time. There are two main steps, which are allocating observed travel times to links, and estimating the travel time of each link. Referring to Table 1

Table 1: Notation used in the paper

τ_i	i -th travel time observation
s_i	first time stamp of the i -th observation
ρ_{ik}	fraction of link k covered by observation i
α_k	fraction of link k included in definition of the route
β_{ik}	fraction of a route link k covered by observation i
ℓ_k	length of link k

Here the travel time is computed by following the 2 aforementioned steps:

1. Travel Time Allocation: The observed travel time τ_i is allocated proportionally to the distance traversed on each link:

$$\tau_{ik} = \frac{\rho_{ik}\ell_k}{\sum_k \rho_{ik}\ell_k} \tau_i \quad (2)$$

2. Aggregation: For observations that are not overlapping a link entirely, the allocated travel time τ_{ik} is scaled to the whole link by the factor $1/\rho_{ik}$ i.e.

$$t_{ik} = \frac{\tau_{ik}}{\rho_{ik}} \quad (3)$$

This scaling assumes that travel speed is homogenous along the link. To acknowledge that observations with little overlap are less reliable, each observation is then weighted according

to the fraction of the link covered, and the average link travel time t_k is defined as the weighted mean over all observations:

$$\bar{t}_k = \frac{\sum_i \rho_{ik} t_{ik}}{\sum_i \rho_{ik}} \quad (4)$$

3. Route Mean Travel Time: To estimate the mean route travel time, the mean travel times of the links in the route are summed up. In case the route starts and ends at non-zero link offsets, link travel times are assigned proportionally to the overlapping length, i.e.,

$$\bar{T} = \sum_k \alpha_k \bar{t}_k \quad (5)$$

The last step has been reported only for knowledge.

Another more structured method which could be seen as a mixture of both disaggregate and aggregate analysis, is described in [18] to estimate travel time. We tested 4 different kinds of scenario:

- Scenario 1: no map matched point on a link;
- Scenario 2: only one matched point from one vehicle on the link;
- Scenario 3: more than one map matched point from one vehicle on the link;
- Scenario 4: more than one matched point from more vehicles on the link;

Method 1 considered map-matched points from adjacent links while estimating travel time to enhance the link coverage and to reduce the uncertainty in the estimation. Method 2 assimilated link travel times from Method-1 in both spatial and temporal ways.

Method 1-Distance and Time proportion Map-matched fixes from step-1 were employed to estimate link travel time. Figure 4 depicts a generic condition in which there are four map-matched points from two different vehicles on link AB.

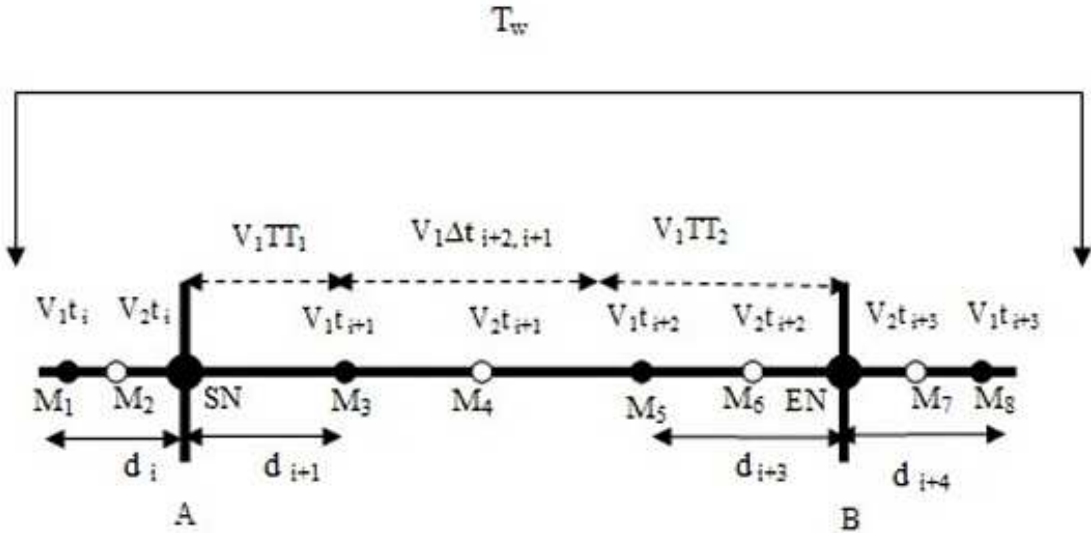


Figure 4: Distance-time proportion travel time estimation [18]

Where:

- M_1 - M_7 represent the map matched vehicles (one represented by black circles and the other by white circles);
- V_1 and V_2 represent respectively the speed of the two vehicles;
- A and B are respectively the start and the end node of the link;
- TT_1 and TT_2 are respectively the travel time to travel the first portion of the arc, defined below, and the travel time to travel the last portion, described below as well;
- $\Delta t_{i+2,i+1}$ is the difference between the timestamp associated with the two map matched points;
- t_i represents the timestamp of each map matched point;

One approach would have been to determine average speed on link AB based on speed measurements (from GPS) associated with these four map-matched points and then use average speed and the length of the link to estimate link travel time. Alternatively, the average speed for each of the vehicles could have been calculated and then employed to estimate two measurements of travel time. Both of these approaches however would likely result in an incorrect estimate of travel time, especially for an urban link where delays are more common at junctions due to traffic lights. To counter this effect, the solution is to divide the link in three portions:

- The first portion is between the starting node of the arc and the first matched vehicle;
- The second one is between the first matched point and the last one;
- The third one is between the last matched point and the end node of the arc;

The travel time is calculated with the following equations:

$$TT_{AB} = \Delta t_{i+2,i+1} + \sum_{i=1}^2 TT_i \quad (6)$$

$$TT_1 = (t_{i+1} - t_i) * \frac{d_{i+1}}{d_{i+1} + d_i} \quad (7)$$

$$TT_2 = (t_{i+3} - t_{i+2}) * \frac{d_{i+3}}{d_{i+1} + d_{i+4}} \quad (8)$$

This makes it possible to consider the effect of the downstream and upstream characteristics of the traffic. Developing each scenario, we have different situations:

- Scenario 1: For the first scenario when no map-matched point existed on a link, then speed limit data of the link was considered as being the 'average speed' to use in estimating the travel time. Otherwise we can employ link-based historical travel time that would be more accurate.
- Scenario 2: For the second the second scenario, when one map-matched point was available on a link, then the distance between the two nodes of link and vehicle speed recorded by GPS receiver was used for link travel time estimation.

- Scenario 3: For the third scenario when there are more than one map matched points from one vehicle on a link, the average speed of that vehicle (\bar{v}) is used to estimate the travel time for the first and last portion of the link. The link travel time therefore consisted on link travel time of each portion, denoted by TT_1, TT_2 and Δt .

$$TT_1 = \frac{\sqrt{(x_i - x_s)^2 + (y_i - y_s)^2}}{\bar{v}} \quad (9)$$

$$TT_2 = \frac{\sqrt{(x_f - x_n)^2 + (y_f - y_n)^2}}{\bar{v}} \quad (10)$$

Where x_s, y_s, x_f, y_f are the coordinates of starting and final node of the link respectively and x_i, y_i, x_n, y_n are the coordinates of the first and last matched point on the arc respectively.

- Scenario 4: it is carried out the same process as the third scenario, for each vehicle and then it is calculated an average.

Method 2-Spatial and Temporal moving average It was reasonable to assume that the link travel time among adjacent links may be correlated with each other for a given time window. For instance, the link travel time for link (AB) may be correlated with the link travel time for both links (DA) and (BC), especially for the same time window length (see Figure 5).

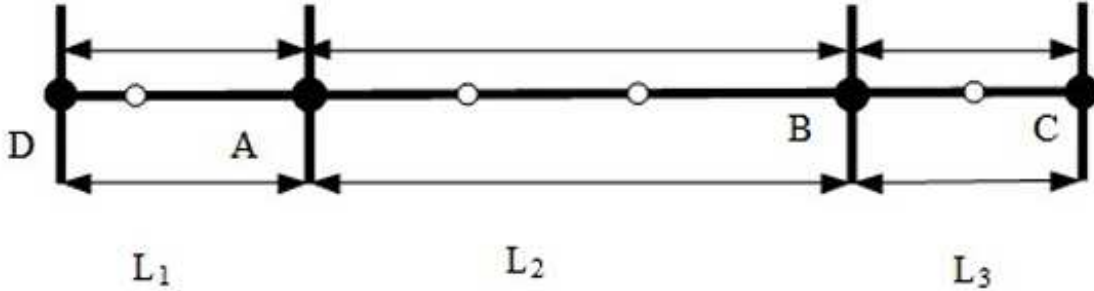


Figure 5: Link travel time using the spatial and temporal connectivity [18]

It was also rational to assume that link travel time of a link may be correlated over time. Especially, link travel time for link (AB) at time window (t) may be correlated with travel time at time window (t-1) and (t+1). Therefore, the estimation of link travel time could potentially be enhanced by link travel time input values from the adjacent links for the same time window (i.e. spatial component) and from the adjacent time windows for the same link (i.e. temporal component). Since vehicle trajectory data from GPS were used in estimating travel time, for the temporal component it was required to integrate the current travel times of a given link with those of the previous and following time windows. For the spatial component, the lengths and travel times of the connected links needed to be used. Therefore, the link travel time estimation based on spatial and temporal component was derived as follows:

- Spatial component: since link travel time for arc AB is correlated with those of link DA and BC, we perform a weighted average, where the weight is the length.

$$T_{AB}^s = \frac{T_{DA} * L_{DA} + T_{AB} * L_{AB} + T_{BC} * L_{BC}}{L_{DA} + L_{AB} + L_{BC}} \quad (11)$$

(b) Temporal component: average link travel time for AB over n_{th} consecutive time windows.

$$T_{AB}^t = \frac{1}{n} \sum_{k=-n}^n T_{AB}^{t-k} \quad (12)$$

Where n is the total number of time windows considered and $n=1$. By taking $n=1$ in (12) it is confirmed that link travel time is estimated by integrating the travel times of the current time window with the previous and the next one.

Then, total link travel time for AB is obtained as a weighted average between spatial and temporal component, according to two empirical coefficients:

$$TT_{AB} = \alpha * T_{AB}^s + \beta * T_{AB}^t \quad (13)$$

In which $\alpha + \beta = 1$ and $0 < \alpha < 1$ and $0 < \beta < 1$. Through empirical analysis $\alpha = 0.1$ and $\beta = 0.9$. It means the effect of spatial correlation in link travel time estimation is much weaker.

2.2.2 Model-based methods

A model-based method is an approach that uses predefined mathematical, physical, or theoretical models to represent and analyse a system or process. These methods are grounded in established principles, such as physical laws, traffic dynamics, or well-defined equations, to predict outcomes or understand behaviours within a given context. For example, a virtual floating car² method is proposed in [22]. The main concept is the proportional decomposition of the total route time in more link travel times. Basically, the general formula is:

$$T_{AB} = \rho_{AB} * T_{total}^{actual} \quad (14)$$

Where ρ_{AB} is the estimated proportion for the link AB calculated with the virtual floating car method. The process to estimate the proportion is the following:

1. Virtual floating car simulation: it is created a virtual floating car simulating the real behaviour of the GPS-equipped vehicle considering decision parameters as positions, speed, acceleration, deceleration and so on..
2. Computation of proportion time along the link: once we have carried out the simulation, we simply make the ratio between the time needed to travel the arc i and the total time needed to complete the route.

$$\rho_{AB} = \frac{T_{AB}^{virtual}}{T_{route}^{virtual}} \quad (15)$$

By simulating vehicle behavior with a virtual floating car, the approach compensates for gaps in real GPS data, leading to more reliable travel time estimates across all road segments.

²A virtual floating car is a simulation of a vehicle that mimics, in a virtual environment, the path and behavior of an actual GPS-equipped vehicle

2.2.3 Hybrid strategies

Hybrid strategies make use of both approaches described above, by combining available data with statistical instruments. For example, in [9] is presented a method for estimating time-dependent travel times tailored to urban logistics by leveraging Floating Car Data (FCD). The primary objective is to enhance routing accuracy in urban environments, where congestion and traffic variability demand reliable, time-sensitive travel times for each road segment. The study introduces various levels of FCD aggregation to calculate time-dependent travel times and employs a Data Mining approach to significantly reduce the data volume required for city logistics routing without sacrificing accuracy. The part we want to highlight is the method described in the third paragraph to compute the travel time of each arc. In fact here, it is suggested to define a time interval and inside this last, select all the available speeds, then, the speed selected will be the median. So it will be possible to calculate the travel time with a simple ratio between distance and speed.

The method presented in [13] is quite different with respect to the previous one. In fact here is presented a statistical model in which link travel times are calculated as the sum of the segment travel time (one link is further divided in more segments with constant characteristics of traffic, such as speed, flow or geometric characteristics) plus the intersection delay. The model is estimated using maximum likelihood. The travel time of a trip is assumed to consist in two parts:

- Link travel time;
- Delay at intersections and traffic signals;

A link is defined to be the road section between two adjacent intersections or traffic signals and as we already stated, they can be divided into more segments. While the links are largely determined by the inherent network structure, the number of segments per link depends on the traffic characteristics of the link. Segments are designed to capture homogeneous traffic behaviour. In this model the average speed of a vehicle can vary between segments but it is assumed to be constant along each segment. The travel time on a segment s is presented as the length of the segment, multiplied with the inverse speed or travel time rate X_s . The travel time rate may depend on observed and unobserved properties of the segment and conditions for the trip. The second component, thus delay at intersections is basically defined by means of a time penalty h_t , influenced by the type of traffic control present at intersections. Here the segment travel time rates and the turn penalties are modeled as stochastic variables.

$$X(\xi) = \mu_s(\xi) + \epsilon_s(\xi) \quad (16)$$

$$Z(\xi) = \mu_a(\xi) + \epsilon_a(\xi) \quad (17)$$

Where $\mu_s(\xi)$ and $\mu_a(\xi)$ are mean values vectors while $\epsilon_s(\xi)$ and $\epsilon_a(\xi)$ are stochastic error terms with $E[\epsilon_s(\xi)] = E[\epsilon_a(\xi)] = 0$.

Observation model We can calculate the travel time of one link as:

$$y_r = l_s * X(\xi) + a_r * Z(\xi) \quad (18)$$

Where l_s is the length of the segment and a_r is 1 if the intersection is visited, otherwise is 0. This for all the segments composing the link. Therefore, the link travel time is a linear combination of the travel time rates on all traversed segments and the penalties at all turns.

2.3 Travel time estimation under free flow conditions

As it is highlighted in [19], travel times, especially on urban road networks, are highly stochastic, so only basing the computation on mean values could lead to some misunderstanding situation. Here, therefore, is proposed a method in which distribution of the travel times on the whole road network is calculated as the sum of the single link travel time plus the turning delay on each intersection. The part we are mostly interested in, is the estimation of the link travel time. In particular, in the paragraph 3.1 is introduced a method to estimate the link travel time under ideal conditions, simply carrying out the ratio between the length of the arc and the speed of the vehicle.

$$t_{ij} = \frac{d_{ij}}{v_{ij}} \quad (19)$$

Here the challenge is to choose the most appropriate speed, which is the maximum speed registered on the arc. The choice is due to the fact that in the paper, the aim is to determine the travel time under ideal conditions, only influenced eventually by delays at the nearest intersection, so this could be a good method to review the travel time under ideal conditions, to be compared with the measured travel time. The speed is computed with a weighted average:

$$v_{ij} = \frac{\sum_{p=1}^u \sum_{r=1}^q w_{ij}^{r,p} \cdot v_{ij}^{r,p}}{\sum_{p=1}^u \sum_{r=1}^q w_{ij}^{r,p}} \quad (20)$$

Where the weights for speeds are calculated using a concept called the "degree of central tendency" which measures how close each GPS sampling point is to the center of the link. This approach assigns greater reliability to points near the center of the link, as these are less affected by slowdowns caused by traffic lights or turns at intersections, which mainly influence points closer to the link's endpoints. The formula for the weight is:

$$w_{ij}^{r,p} = 1 - |2 \cdot \theta_{ij}^{r,p} - 1| \quad (21)$$

Where $\theta_{ij}^{r,p}$ represents the relative position of the sampling point on the link, expressed as a value between 0 and 1:

- If $\theta_{ij}^{r,p}$ is close to 0.5, the point is central and has a higher weight;
- If $\theta_{ij}^{r,p}$ is near 0 or 1 (closer to the link edges), the weight decreases, reducing the influence of that speed on the final estimate, because probably that speed is affected by the presence of the intersection.

2.4 Delay estimation in aggregate fashion

Here in [20] it is presented a method to globally validate the use of FCD to support policy maker in taking decisions. The interesting part is more related to the determination of the total delay. In fact here is presented a more aggregate analysis, in which the delay is calculated as following:

$$Delay = (TT_{observed} - TT_{freeflow}) * Volume \quad (22)$$

It is pretty simple but I would say efficient. Here the only challenge is to define the correct speeds. The average observed speed is determined for each interval of 15 minutes while the free flow speed is the maximum observed speed during off peak hours after having removed the 20% highest speeds.

2.5 Indicators of congestion

This section explores key metrics used in literature to evaluate congestion levels on road networks. By analyzing these indicators, we can identify bottlenecks, assess travel delays, and understand the overall impact of congestion on traffic flow efficiency.

2.5.1 Key Performance Indicator based on highly disaggregate level analysis using travel time

In [15] a highly disaggregated analysis has been developed to assess traffic congestion, initially conducted at the arc level by evaluating the travel time of each individual vehicle. This analysis was then slightly aggregated at the link level, considering the combined travel times of all vehicles on the same arc. Initially, the time lost is defined as follows:

$$KPI_{j,k} = T_{0j} - (T_{GPS_{j,k}} - T_{GPS_{SS_{j,k}}}) \quad (23)$$

Where T_{0j} is the free flow travel time of arc j and $(T_{GPS_{j,k}} - T_{GPS_{SS_{j,k}}})$ is the observed travel time of vehicle k along arc j purified from the eventual service stop time. To consider also the length of the arc, because clearly the time is strongly depending on it, we can define a relative indicator:

$$RKPI_{j,k} = \frac{T_{0j} - (T_{GPS_{j,k}} - T_{GPS_{SS_{j,k}}})}{T_{0j}} \quad (24)$$

Negative values of those indicators respectively represent absolute or relative measures of the time potentially lost in congestion. Zero values indicate that vehicle was travelling at free flow speed. Can even occur cases in which there are positive values, because free flow travel time is an average estimation as we will see in 5.1, so it can happens a vehicle is travelling at higher speeds, especially during off-peak hours. Starting from those indicators, it is possible to aggregate the results at different scales, according to the specific transport policy questions that need to be answered.

Then, results are visualized by plotting for example the minimum RKPI for each arc j , for different hourly intervals, especially for peak hours. Through these maps we can eventually appreciate where are the most congested link to act on them.

2.5.2 Performance Indicator based on qualitative analysis of average speeds

In [2] average speed values were transformed into a qualitative 4-scale state parameter based on the Level of Service (LOS) definitions for urban roads by using raw FCD data sampled every 1 minute. After transforming average speeds into predefined states, a series of search algorithms were developed to detect critical patterns in urban traffic, depending on the number of segments considered.

LOS is a quantitative measure representing quality of service [3]. Generally, 6 different LOS states are defined for different road types, where LOS A represents the best operating condition and LOS F the worst. HCM defined LOS for urban roads as "*the reductions in travel speed as a percentage of the free-flow speed of the corridor*". Table 2 shows those percentages.

Table 2: Percentage of free flow speed corresponding to different LOS and assumed traffic state

LOS	Travel Speed as a Percentage of Base Free-flow Speed	Assumed Traffic State
A	>85%	1
B	67-85%	1
C	50-67%	2
D	40-50%	3
E	30-40%	3
F	<30%	4

Then, during the searches, the traffic state in each segment was compared against those of the following segment(s), in order to acquire different patterns such as bottleneck release, persistent congestion, etc. At the final stage, all segments were evaluated to detect the frequency and start point of the predefined critical patterns in the extensive FCD archive of the corridor.

2.6 Contextualizing the Proposed Methodology within Current Research

We have just seen some of the many methodologies present in literature. Our work is related with the three aspects, Map Matching, determination of observed travel time and determination of free flow travel time. Concerning Map Matching, we applied of course a geometric approach, because of its simplicity and efficiency, more or less like that described in [12] (point to line) but with different elaboration.

Concerning observed travel time, I would say the best method to measure it considering the efficiency and the simplicity is presented in [18], however, divide the arcs in three part could be really challenging and needs a big effort, which is not even related with the final purpose of this thesis, therefore we will opt for a more simple method, which we didn't find in any research. The calculus of travel time is based on a proportionality concept, thus, we know the timestamps and the distance travelled between the points and as a consequence with a proportionality calculus we can determine the total travel time to travel the current arc; this is carried out for each vehicle on each arc and then we perform a weighted average among all the travel times.

Concerning the evaluation of the travel time under free flow conditions, we followed two approaches, one based on what is described in [19] and the other based on usage of the already known data associated with the graph.

3 Experimental setting

Here we will give a clear context of the study area, reporting information related with the infrastructure and mobility, and then we describe all the data we used to carry out this work along with their features, involving the process that led us to have the complete dataset.

3.1 Study context: Turin

Turin (Torino in italian), located in the northwestern region of Italy, serves as the capital of the Piedmont region. Renowned for its historical significance, industrial heritage, and cultural prominence, Turin has emerged as a pivotal hub for urban mobility studies, making it an ideal case study for this research. With a population of approximately 900.000 residents within the city limits and over 1.7 million in its metropolitan area, Turin is one of Italy’s largest urban centres. This dense and diverse population drives significant demands for efficient transportation and logistics systems. The city’s urban layout is characterized by a blend of historic streets, modern thoroughfares, and green spaces, with a grid pattern especially in the urban central area, providing a unique backdrop for studying mobility patterns. Its compact centre contrasts sharply with sprawling suburban developments, creating distinct transportation challenges and opportunities.



Figure 6: Plan view of Turin, can appreciate the grid in the central area

The extension of the study area is about 1132.9 km², consisting of 62 municipalities and it has been delimited with a polygon created on the basis of the available graph, as it is shown in Figures 7 and 8:

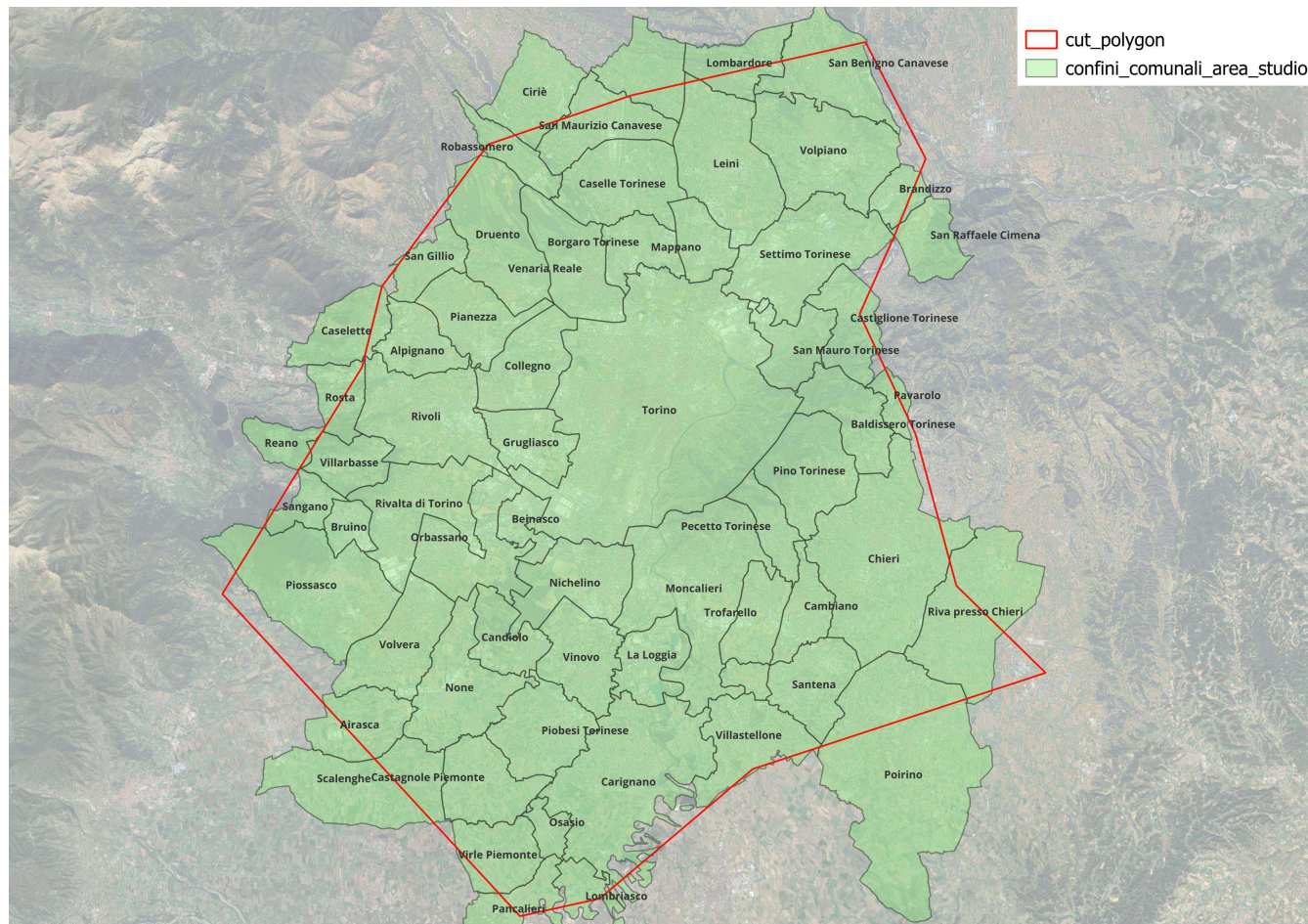


Figure 7: Extension of study area with municipalities, from Qgis elaboration

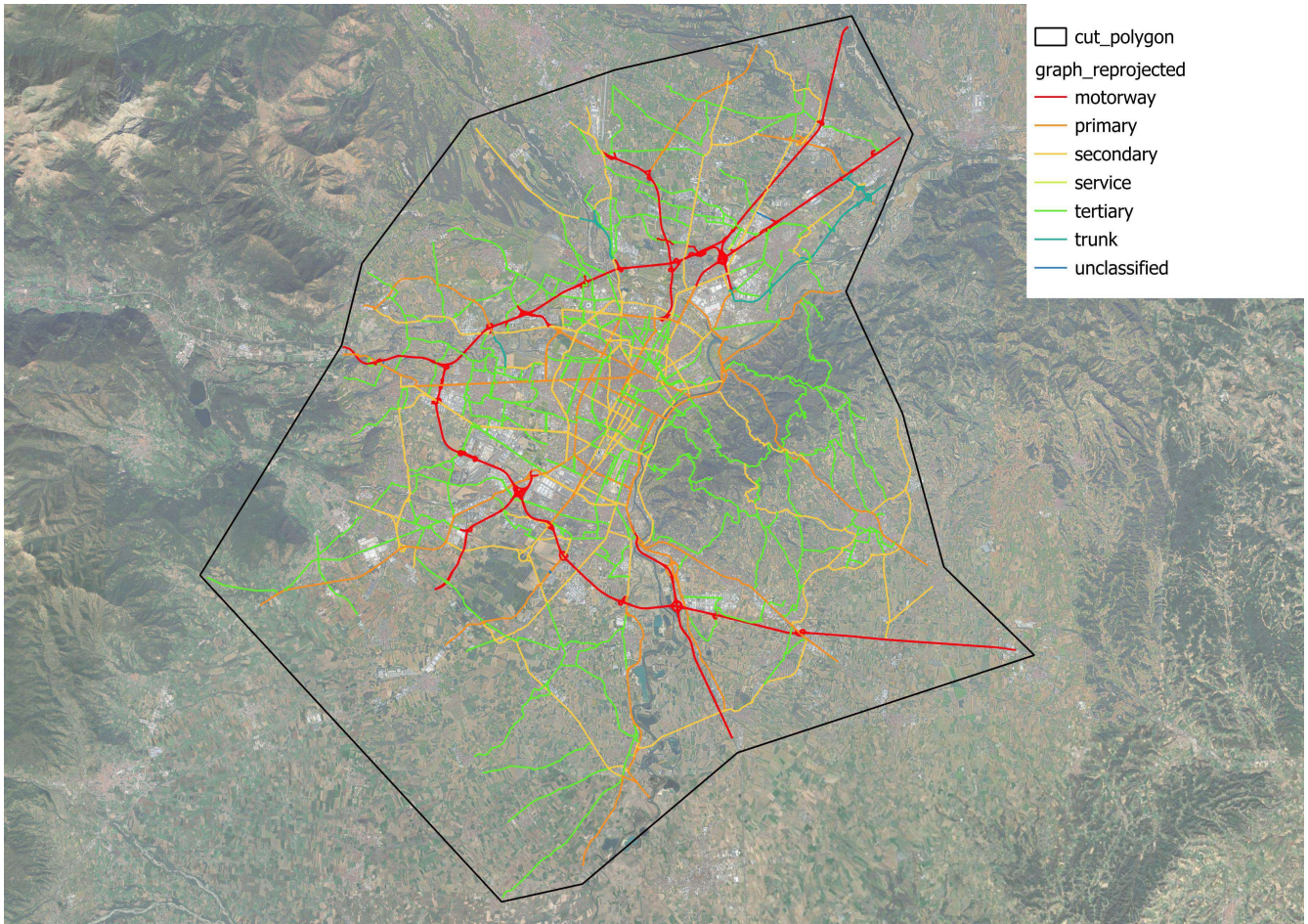


Figure 8: Extension of study area with road graph, from Qgis elaboration

Municipalities boundaries have been taken from the official Istat website.³

3.2 Mobility and road infrastructures in the study area

Some information about road network has been taken from PUMS (Sustainable Mobility Urban Plan [6]).

The metropolitan mobility system in Turin is based on a complex infrastructure network, combining historical elements with more recent developments. The road network of the Metropolitan City is centered around major highways, including the A4 Torino-Milano-Venezia-Trieste (1932), the A5 Torino-Ivrea-Aosta-Monte Bianco with a branch to Santhià (1961), the A6 Torino-Savona (1960), the A21 Torino-Piacenza-Brescia (1968-69), and the A32 Torino-Bardonecchia with the Fréjus tunnel (1992-94). These routes converge on the A55 ring road (1976), which also includes the branch to Pinerolo (1992-2006) and the highway connection to Caselle Airport (RA10). In total, the highway network within the metropolitan boundaries spans 316 km, approximately 60 km of which are part of the ring road. The ordinary road network, consisting of a limited number of state roads that follow historical routes and over 300 provincial roads (around 450 if including branches), plays a complementary role to the aforementioned main highways. The network is further complemented by numerous local roads that densely connect the entire plains and hill areas,

³<https://www.istat.it/notizia/confini-delle-unita-amministrative-a-fini-statistici-al-1-gennaio-2018-2/>

supporting short- and medium-range inter-municipal mobility. Overall, the ordinary road network extends for approximately 5600 km.

ESTESA - Città Metropolitana			
Classe	Totale	Ambito urbano	Ambito extraurbano
	<i>km</i>	<i>km</i>	<i>km</i>
Autostrade	316	66	249
Principali	479	242	237
Secondarie	932	365	566
Complementari	647	341	307
Locali	3.609	1.793	1.816
TOTALE	5.982	2.807	3.175

Figure 9: Extension of Metropolitan City of Turin [6]

The analysis of the circulating vehicle fleet was conducted using data collected by the Automobile Club d'Italia (ACI) and included in an annually published study called Autoritratto⁴. This study provides detailed information on the Italian vehicle fleet according to various spatial and temporal aggregations (e.g., vehicles by region or province, year of registration) and vehicle characteristics (e.g., passenger cars, commercial vehicles, buses, fuel type, engine capacity, emission class).

Focusing on the Metropolitan City of Turin for the year 2019: The total circulating vehicle fleet is of about 1.9 million, of which 77% is represented by passenger cars. If we relate those data to the resident population of 2.25 million in December 2019, we have high motorization rate, of 657 passenger cars and 848 vehicles for each 1000 inhabitants.

⁴<https://www.aci.it/laci/studi-e-ricerche/dati-e-statistiche/autoritratto.html>

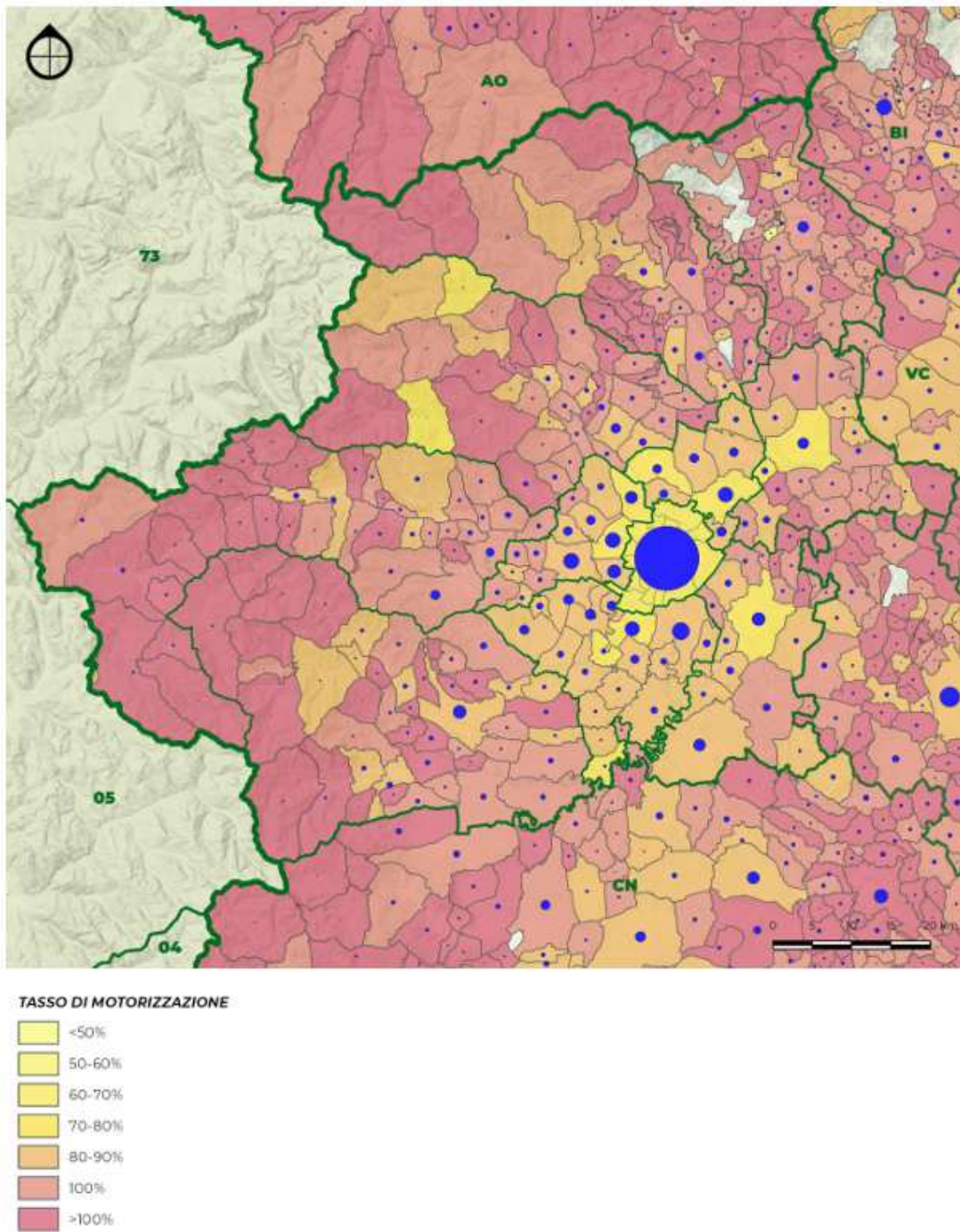


Figure 10: Motorization Rate of Metropolitan City of Turin [6]

It can be seen in Figure 10, as we could expect, that the rate increases as we move towards the external areas. Blue circles are proportional to the number of inhabitants of the municipality.

To this high motorization rate, there naturally corresponds a high number of interchange trips using private vehicles, primarily driven by reasons such as work and study. We can then distinguish net mobility-generating municipalities, those where the number of residents commuting outside the municipal boundaries for study and work exceeds the number of incoming commuters for the same reasons and net mobility-attracting municipalities, with more individuals entering than leaving for work or study. This distinction is illustrated in Figure 11, where attracting municipalities are highlighted in red and generating ones in blue.

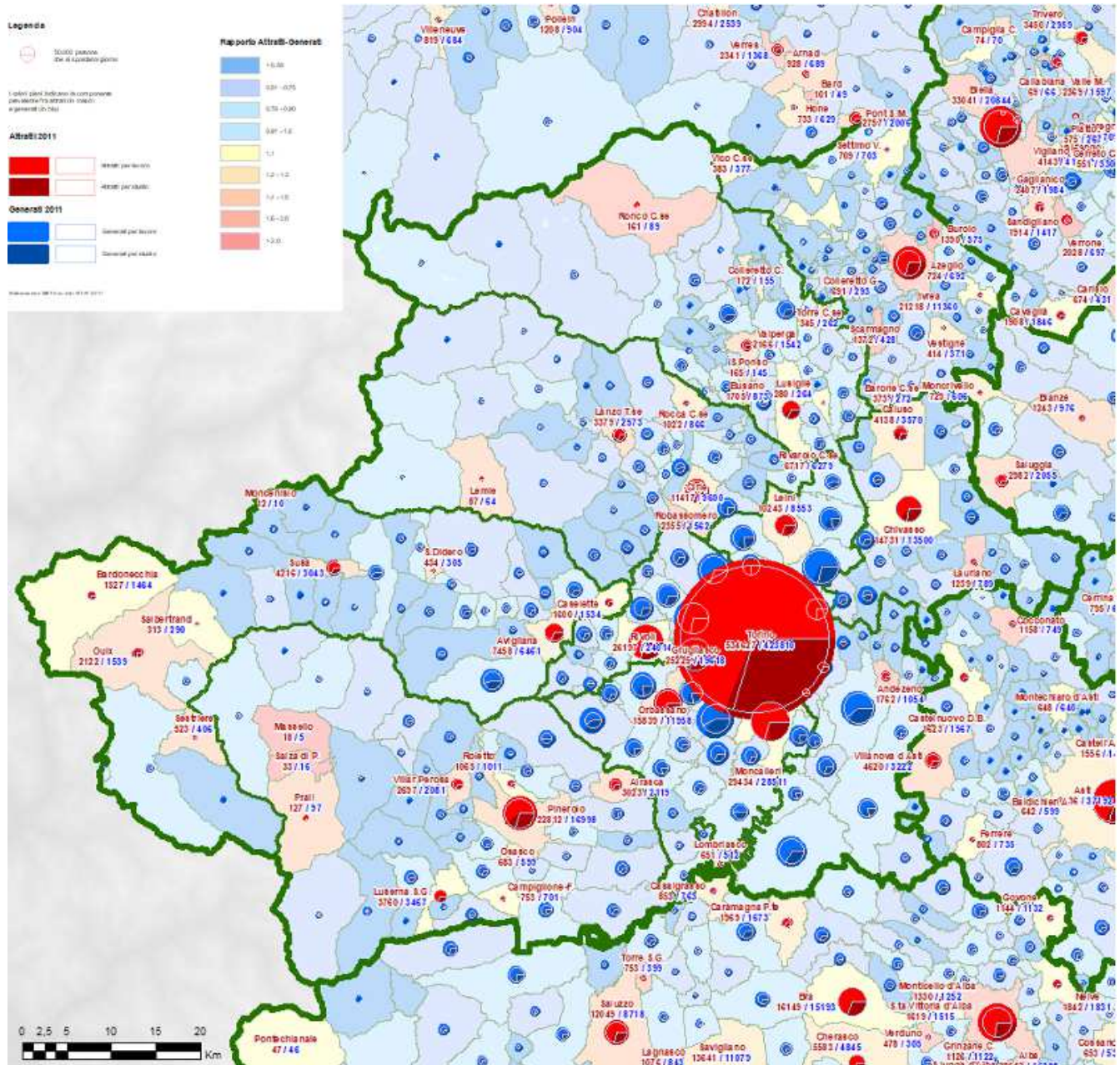


Figure 11: Main attractor and generator poles [6]

As observed, the primary metropolitan-level attractor is clearly the City of Turin, which draws over half a million systematic inbound trips, while generating slightly more than 400,000 outbound trips. Going more into detail, within the city of Turin, it is possible to identify its area of influence

based on two primary reasons: study (Figure 12) and work (Figure 13). These distinct purposes shape commuting patterns, highlighting the city's role as a central hub for both educational and employment opportunities in the metropolitan area.

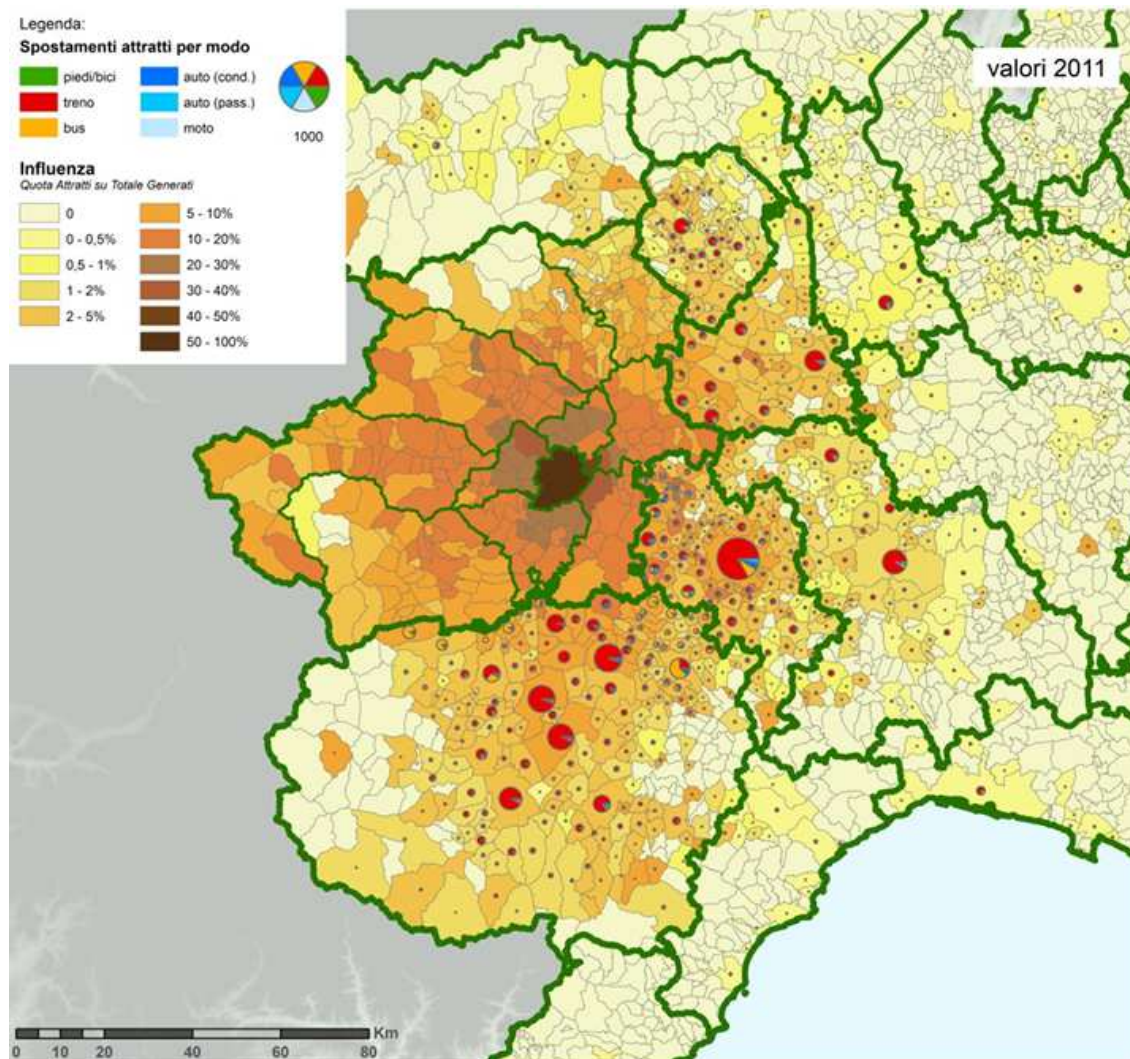


Figure 12: Influence area of Turin-study [6]

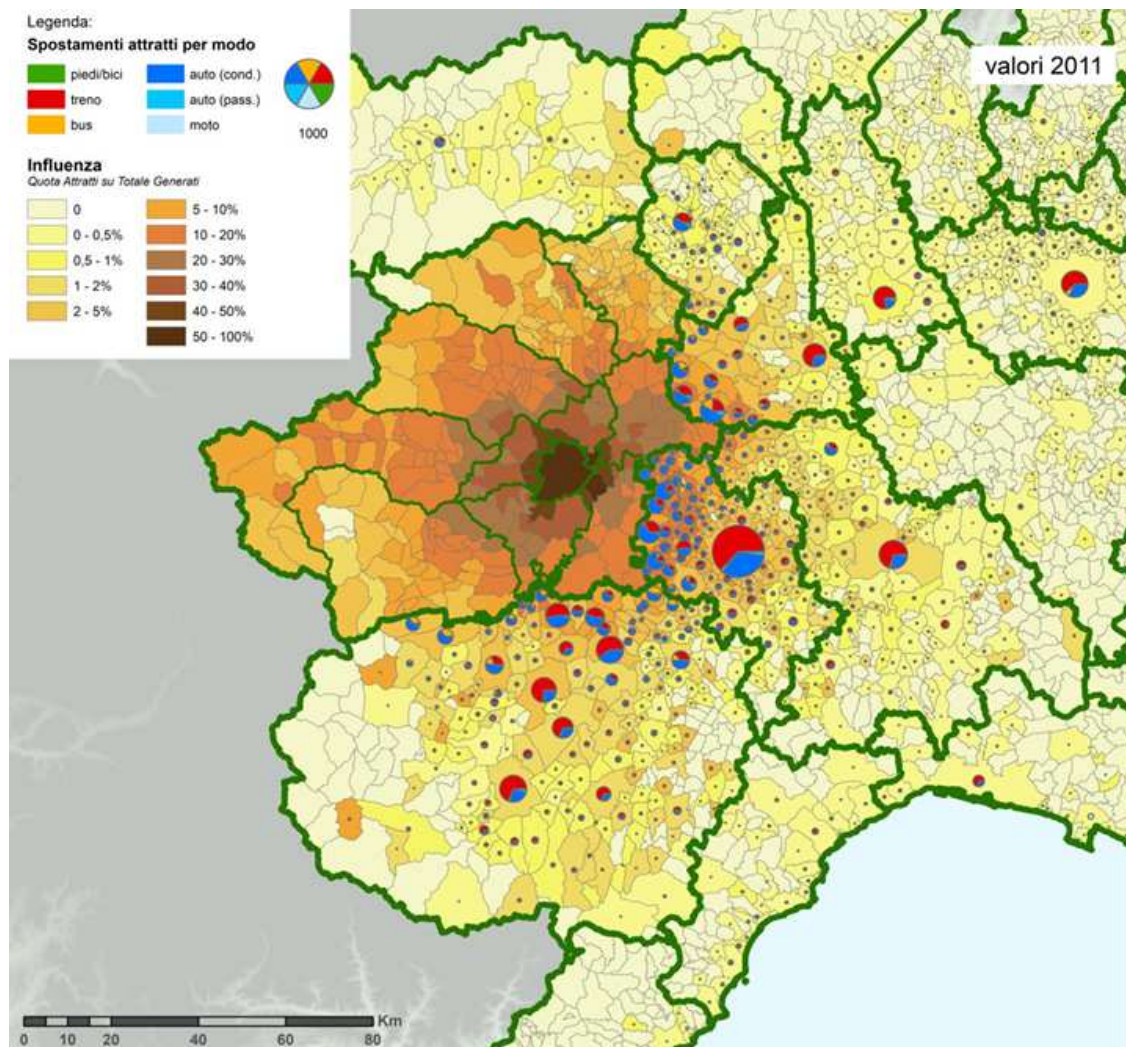


Figure 13: Influence area of Turin-work [6]

3.3 Floating car data used in the study area

Through an agreement between TIM and the Municipality of Turin, trips to and from Turin were recorded between January 15th, 2019, and January 17th, 2020. This initiative registered a total of 120,000 trips per day and 21,000 vehicles per day. These data were made possible by leveraging GPS coordinates and timestamps collected by TIM from individuals subscribed to the company's services.

3.3.1 Overview of Commercially Available Floating Car Data

A 'Floating Car' is defined as a vehicle that provides position and kinematics data, such as speed and direction of travel, to gather traffic information for Intelligent Transportation System (ITS) applications. These data can be processed by remote control centers for multiple purposes, for example, to monitor road traffic conditions and prevent congestion to inform fleets of vehicles cooperatively driving, to detect possible in-vehicle malfunctions, to collect traffic statistics, to get car maintenance tips and service information. The current need for FCD could be satisfied by

using mass-market users' devices, such as smartphones, and available technologies, such as cellular and Wi-Fi networks. This conviction is shared with the scientific community and many running initiatives are pushing toward enabling users' portable devices, such as tablets and phones, to access in-vehicle telematics and to monitor in-vehicle services through low-cost devices and open-source software.

Nowadays, there is a plenty of companies furnishing these kind of the data in the field of ITS, like TomTom, INRIX, Google Maps, Teralytics and so on. All of them acquire data from mobile devices and GPS devices placed on vehicles. In particular, also telephonic operators like TIM, Vodafone, Movistar and so on collect those data and sell them to private or public institutions (see Section 3.3 *Floating car data used in the study area*). In particular, the process begins with data collection through various sources within their network. Base transceiver stations (BTS), for example, record signals sent by mobile devices, providing an approximate location based on the tower's position. As devices move between towers, "handover" events are recorded, enabling the tracking of movement patterns. Additionally, network events like internet connections, SMS, and calls contribute to location data, which can be further enhanced with Wi-Fi hotspot and offload information. To ensure user privacy and comply with regulations like GDPR, the collected data undergoes anonymization and aggregation. Anonymization removes personal identifiers such as phone numbers, while aggregation combines data from multiple users into statistical insights, ensuring that individual movements cannot be traced. The raw data is then processed using advanced techniques. Filters remove erroneous signals, while localization algorithms, including triangulation or GPS-based methods when available, improve positional accuracy. The resulting data is analyzed to identify traffic flows, congestion points, and frequently used routes. The insights derived from these data serve a variety of applications among which the purpose of this thesis declared in 1.1 *Thesis objectives*.

3.3.2 Historical Car Data (HCD)

Historical Car Data, refers to vehicle data collected in batch mode over an extended period.

In particular, their aim is to capture aggregate information such as average speed, traffic volumes and travel times on various road segments. This is a useful tool to plan and to manage the traffic because it makes understand the main trends. Moreover, it's a cost-effective method, and their big advantage is that they provide continuous data across the entire road network, unlike sensors, which are limited to the specific location where they are installed.

Referring to our dataset, in the document we have, named "Gestione invio dati HCD e FCD" are described the procedures and the metadata related with the acquisition of Historical Car Data and Floating Car Data.

HCD are extracted once per day, not before 09:00 AM and refer to data belonging to the previous day. For each day recorded, there is one heading representing the origin/destination of the trips carried out during that specific day and n details representing the successive movements of the vehicle during each trip. The heading consists of 368 .csv files named in the format `polito_viaggi_YYYY_MM_DD`, where YYYY represents the year, MM the month, and DD the day. Each file contains hundreds of thousands of records, with each record representing a trip along with the associated origin and destination coordinates during that day. The data spans the period from December 18th, 2018, to January 14th, 2020. 366 files concern the whole year between January 14th, 2019 and January 14th, 2020. 2 files more concerns December 18th, 2018 and December 19th, 2018. These files contain the following information:

- Trip numerical identifier;

- Device identifier;
- Trip start timestamp (UTC), format YYYY-MM-DD HH24:MI:SS;
- Trip end timestamp (UTC), format YYYY-MM-DD HH24:MI:SS;
- Trip start latitude and longitude, with 6 significant decimal digits after the separator which is the dot (.);
- Trip end latitude and longitude, with 6 significant decimal digits after the separator which is the dot (.);
- Starting address in textual format: State—Region—Province—Municipality—Address;
- ISTAT code of starting municipality as result of census of 2011;
- ACE code of starting zone (Census area);
- Arrival address in textual format: State—Region—Province—Municipality—Address;
- ISTAT code of arrival municipality as result of census of 2011;
- ACE code of arrival zone (Census area);
- Total km travelled, with minimum resolution of 100 m;
- Average speed in km/h;
- Total number of samples that define the travel detail;
- Vehicle typology (1:autovehicle,2:commercial vehicle);
- Vehicle brand, if not known "NULL";
- Vehicle model in textual format, if not known "NULL";
- Owner sex if physical person ("M" or "F"), "S" if legal person, if not known "NULL";
- Owner age, "S" for legal person, if not known "NULL";
- Starting weather conditions: temperature (celsius), precipitation (0-1-2-3 depending on the intensity), textual description;
- Arrival weather conditions: temperature (celsius), precipitation (0-1-2-3 depending on the intensity), textual description;

The metadata are described in Table 3.

Table 3: HCD Metadata

Field	Rules
Trip_id	From 1 to 20 characters
Device_Id	From 1 to 20 characters
Datetime_partenza	(format UTC: YYYY-MM-DD HH24:MI:SS)
Datetime_arrivo	(format UTC: YYYY-MM-DD HH24:MI:SS)
Lat_partenza	6 significant decimal numbers, separator is dot
Lon_partenza	6 significant decimal numbers, separator is dot
Lat_arrivo	6 significant decimal numbers, separator is dot
Lon_arrivo	6 significant decimal numbers, separator is dot
Indirizzo_partenza	State Region Province Municipality Address, separator is
Codice Istat comune partenza	ISTAT census 2011, from 4 to 6 digits
Codice ACE partenza	Code that identifies univocally the census area, if present, within the municipal territory, from 0 to 3 digits
Indirizzo_arrivo	State Region Province Municipality Address, separator is
Codice Istat comune arrivo	ISTAT census 2011, from 4 to 6 digits
Codice ACE arrivo	Code that identifies univocally the census area, if present, within the municipal territory, from 0 to 3 digits
km_percorsi	Precision of 2 decimals digits, separator is .
speedKmh	Speed on the arc, expressed in km/h
Nsamples	Number of samples constituting the trip details
Tipologia	1 car 2 fleet
Marca_veicolo	Es. Fiat, Toyota...
Modello_veicolo	Es. Panda, Clio, Golf
Sesso	M or F for physical persons
Età intestatario	If not known 0
Temperatura alla partenza	Up to 1 decimal digit, separator is ., if not known 99
Precipitazioni alla partenza	0:no rain, 1: 1mm/h, 2: 4 mm/h, 3: >4 mm/h, 9: not known
Condizioni meteo partenza	If absent "UNK"
Temperatura all'arrivo	Up to 1 decimal digit, separator is ., if not known 99
Precipitazioni all'arrivo	0:no rain, 1: 1mm/h, 2: 4 mm/h, 3: >4 mm/h, 9: not known
Condizioni meteo arrivo	If absent "UNK"

Additionally, for each trip on each day, the dataset includes the sequence of recorded positions, stored in 8,832 (since $24 \times 368 = 8832$) .csv files with hundreds of thousands of records, named in the format `polito_dett_YYYY_MM_DD_HH.csv`, where YYYY represents the year, MM the month, DD the day, and HH the hour interval. Data refer to travels which started or ended or crossed the interest area on the days between December 18th, 2018, to January 14th, 2020. These are the information described:

- Trip numerical identifier;
- Device identifier;
- Timestamp in format UTC, YYYY-MM-DD HH:MI:SS;
- Latitude and longitude, 6 decimal digits, separator is . ;
- Address in textual format: State—Region—Province—Municipality—Address;
- ISTAT code of municipality as result of census of 2011;
- ACE code of Census area;
- Road class ("U", "E", "A", "X": urban, rural, motorway, other);
- Weather condition (textual and numeric (0-1-2-3) depending on the precipitation intensity) aggregated/updated according to the following rules:
 - If the position remain within the same municipality, each 30 minutes;
 - Sampling in municipality different from the previous one;
- Speed in km/h;
- Gps signal quality (3 excellent, 2 sufficient, 1 poor);

The metadata is described in Table 4.

Table 4: HCD Trip Details Metadata

Field	Rules
Trip_id	From 1 to 20 characters
Device_Id	From 1 to 20 characters
Datetime	(format UTC: YYYY-MM-DD HH24:MI:SS)
Lat	6 significant decimal numbers, separator is dot
Lon	6 significant decimal numbers, separator is dot
Indirizzo	State Region Province Municipality Address, separator is
Codice Istat comune	ISTAT census 2011, from 4 to 6 digits
Codice ACE	Code that identifies univocally the census area, if present, within the municipal territory, from 0 to 3 digits
Tipologia strada	U:urban E:rural A:motorway X:other
Temperatura	Up to 1 decimal digit, if not known 99
Precipitazioni	0:no rain, 1: 1mm/h, 2: 4 mm/h, 3: >4 mm/h, 9: not known
Condizioni meteo	If absent "UNK"
speedKmh	Speed on the arc, expressed in km/h
Hdop	Gps signal quality: 3 excellent, 2 sufficient, 1 poor

3.3.3 Floating Car Data (FCD)

Floating Car Data refers to data collected in near real time from vehicles equipped with GPS devices as they are moving throughout the road network, they provide real time data which makes understand the current traffic conditions and can support the policy maker especially in short term decisions, or they can even be used to assess the effectiveness of regulatory actions on traffic. Here as well we have a set of 264,383 files with thousands of records each one. Data was made available in format .csv, they have been sampled with a 2 minutes frequency and therefore are named with the format "VST_POLITO_YYYYMMDD_hhmmss.csv" where YYYY represents the year, MM the month, DD the day, hh the hour, mm the minute and ss the second. They have a temporal extension which starts from January,14th 2019 to January, 17th 2020. The big number is due to the fact that in one hour of each day we have about thirty file, since they have been sampled every 2 minutes. Therefore in one day we have $30 \cdot 24 = 720$ files. There are 366 entire days sampled, so $720 \cdot 366 = 263,520$ files plus those coming from January,14th 2019 that is sampled only from 09:46:57 ahead and January, 17th 2020 that is sampled only until 11:09:44. They have been made available immediately after the acquisition, without any filtering activity. Each sample has the following characteristics:

- Request identifier;
- Device identifier;
- Datetime of recording, same format as before;
- Latitude and longitude;
- Speed expressed in km/h;
- Direction of motion expressed with a number between 0 and 360, pointing the angle with respect to the North;
- Gps signal quality, expressed in hdop decimals, between 0 (excellent) and 150 (poor);
- Engine state (1 turned on, 0 turned off);
- Eventual other optional informations;
- Vehicle typology (1 autovehicle, 2 commercial vehicle)

As well as before, the metadata is described in Table 5.

Table 5: FCD Metadata

Field	Rules
idRequest	From 1 to 20 characters
DeviceId	From 1 to 20 characters
dateTime	format UTC: YYYY-MM-DD HH24:MI:SS
latitude	6 significant decimal numbers, separator is dot
longitude	6 significant decimal numbers, separator is dot
speedKmh	Speed on the arc, expressed in km/h
heading	From 0 to 360, with increment of 4 degrees
accuracyDop	From 0 to 150
EngineStatus	1 engine on 0 engine off
Type	1 car 2 fleet

3.3.4 Differences between HCD and FCD

To briefly highlight the main differences between the two types of data, as they might appear similar at first glance, the key distinction lies in the period of data collection. Specifically, while HCD (as detailed in Section 3.3.2 *Historical Car Data (HCD)*) are collected offline, typically on the following day, FCD are gathered in near real time. This means that FCD are not instantly available but are accessible shortly after collection, making them suitable for obtaining real-time information.

Another significant difference lies in the structure of the generated files. For HCD, data is typically split across multiple files: one file identifies the trip's origin and destination (`polito_viaggi_YYYY_MM_DD`) while additional files record the subsequent positions along the route (`polito_dett_YYYY_MM_DD_HH.csv`). In contrast, FCD consolidates all consecutive positions of each vehicle into a single file (`VST_POLITO_YYYYMM`). This difference likely reflects the nature of their processing; HCD are usually post-processed, whereas FCD are collected and utilized as-is due to their near real-time acquisition.

3.4 Graph of Turin

Beyond the GPS traces, we have also the graph (see Section 1.3.1 *Road Graph*) of the main roads in Turin and province.

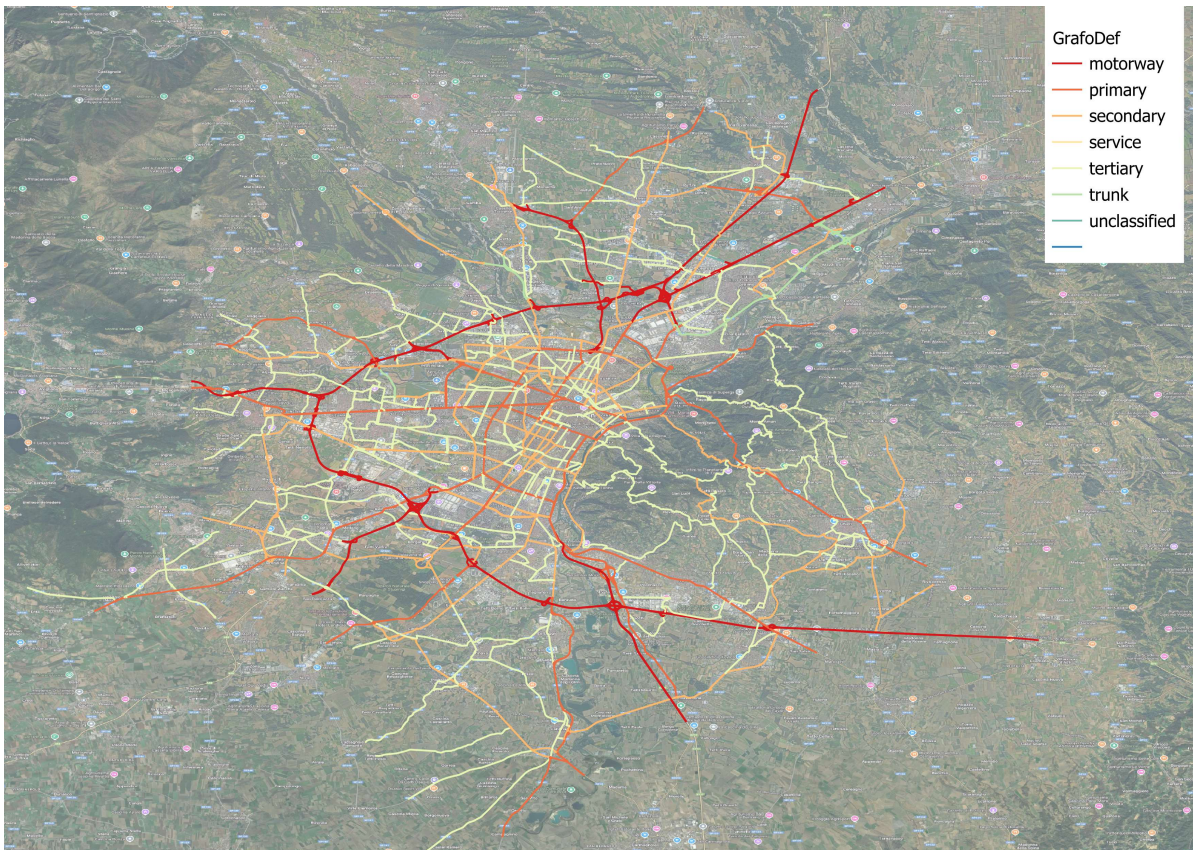


Figure 14: Graph view on Qgis

This has been taken from the master thesis [14] whereby is described the process to extract the graph. Here, to sum up we report only the main details. The input data are:

- Open Street Map⁵ road network available on the Geofabrik⁶ website;
- The graph "DatiSVR2019_su_ElementoStradaleBDTRE"⁷ available on Geoportal of region Piemonte;

OSM geometries are represented with good accuracy and include information about the direction of travel. However, they do not always provide data on travel speed or the technical-functional classification of the roads. The BD TRE graph, on the other hand, contains information updated to 2019, including the average daily traffic (ADT), maximum, and average travel speeds. However, its representation is less accurate. Their characteristics has been mixed to get the following attributes of each arc:

- Identifier number of the arc;
- Identifier number of starting node;
- Identifier number of end node;
- Direction;
- Length;
- Class (displayed in the Figure 14);
- Travelling speed (km/h);
- Capacity;
- Number of lanes;

The final graph will have therefore a number of 7549 arcs representing the main infrastructures in the area of Turin. In Table 6 we represent the metadata.

Table 6: Graph Metadata

Field	Rules
Arco	Identifier of the arc, from 1 to 4 characters
Nodo.i	Starting node identifier, from 1 to 4 characters
Nodo.f	Final node identifier, from 1 to 4 characters
Centroid	Null
Dir	F if it's one-way or B if it's two-way
L [m]	Length of the arc expressed in meters
Main class	motorway;primary;secondary;tertiary;trunk;unclassified;service
Vf [km/h]	Average traveling speed on that arc
C [ve/h]	Capacity of the arc expressed in vehicles per hour
N_corsie	Number of lanes

⁵OpenStreetMap (abbreviated OSM) is a website that uses an open geographic database which is updated and maintained by a community of volunteers via open collaboration

⁶<https://download.geofabrik.de/europe/italy/nord-ovest.html>

⁷https://www.geoportale.piemonte.it/geonetwork/srv/ita/catalog.search/metadata/r_piemon : 2bb551d2-bad8-488f-9070-07f5a65b5f11

We have it available as a file named 'graph_reprojected.shp' that we imported on Qgis to perform our analysis, and further we exported the attribute table as an Excel file named 'grafo.csv' to process it as it will be showed in Chapter 5 *Methodology, part 2: travel times derivation*.

4 Methodology, part 1: HCD data processing and map matching

Here we will focus on the process that led us to the dataset to be analysed for the purposes of our thesis, including data pre-processing. As we already said, in the previous Section 3 *Experimental setting* we have two kinds of data, thus HCD and FCD. We will exploit them for different purposes. In particular, HCD will be used to develop a methodology with which will be determined the free flow speed on the arcs. Whereas, FCD will be used to meet the purpose of this thesis, explained in Section 1.1 *Thesis objectives*, thus determine the arcs where most time is wasted due to congestion.

4.1 HCD data pre-processing

As described in Section 3.3.2 *Historical Car Data (HCD)*, our dataset covers one year of data. For computational and efficiency reasons, we could not use the entire dataset for our analysis. Thus, we began by cleaning and selecting relevant data. To select dates for free flow sampling, I will take one day every $365/n$ at constant intervals, focusing on hours between 10 PM and 6 AM, where n is fixed in such a way to have at least 10 days for each month, so $n=120$. Concerning the determination of congestion, we used an equal probability stratified sampling. This sampling technique ensures that each stratum (subgroup) is represented equally in terms of probability, regardless of the size or other characteristics of the strata.

Firstly, we excluded particular days like holidays, snowfalls and strikes. In Table 7 we reported what days have been excluded and the reason.

Table 7: Classified List of Significant Dates in 2019

Date	Reason
01/01/2019	holiday
06/01/2019	holiday
16/01/2019	strike
23/01/2019	snowfall
31/01/2019	snowfall
01/02/2019	snowfall
08/03/2019	strike
21/04/2019	holiday
22/04/2019	holiday
25/04/2019	holiday
27/04/2019	strike
01/05/2019	holiday
31/05/2019	strike
02/06/2019	holiday
24/07/2019	strike
14/10/2019	strike
25/10/2019	strike
22/11/2019	strike
08/12/2019	holiday
13/12/2019	snowfall, strike
25/12/2019	holiday
26/12/2019	holiday

Then, we considered as stratifying variables: Season (4 categories), a combination of August/school calendar (3 categories: from early September to mid-June / from mid-June to the end of July and August plus early September) and day of the week (7 categories). We therefore constructed a three dimensional contingency table counting the days lying on each strata, that we report in Figure 15:

School Calendar	August							Early Sept to Mid-June							Mid-June to End-July						
Day of Week	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
Season																					
Autumn	0	0	0	0	0	0	0	10	11	11	11	8	11	11	2	2	2	2	2	2	2
Spring	0	0	0	0	0	0	0	12	13	12	12	12	12	12	0	0	0	0	0	0	0
Summer	4	4	4	4	5	5	4	0	0	0	0	0	0	0	9	9	8	8	8	9	8
Winter	0	0	0	0	0	0	0	14	14	10	11	10	12	12	0	0	0	0	0	0	0

Figure 15: Three dimension contingency table

Now, since we want to maintain the same proportion of each strata in the sampling process, we would sample around 10 days for each month, so we assume $n=10*12=120$ and so, applied the following formula to each cell:

$$Sample = \frac{Cell\ frequency}{Total\ population} * n \tag{25}$$

Where the total population is 344 days resulting from the 366 normal days minus the excluded days in Table 7. The resulting table is shown in Figure 16:

School Calendar	August							Early Sept to Mid-June							Mid-June to End-July						
Day of Week	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
Season																					
Autumn	0	0	0	0	0	0	0	3	4	4	4	3	4	4	1	1	1	1	1	1	1
Spring	0	0	0	0	0	0	0	4	5	4	4	4	4	4	0	0	0	0	0	0	0
Summer	1	1	1	1	2	2	1	0	0	0	0	0	0	0	3	3	3	3	3	3	3
Winter	0	0	0	0	0	0	0	5	5	3	4	3	4	4	0	0	0	0	0	0	0

Figure 16: Sampled strata

For each stratum, we selected random day. Focusing on free flow speed, the process above described, resulted in 102 files, named using the format "MMDDYYYY_HCD.csv," where MM represents the month, DD the day, and YYYY the year. To reduce the dataset size and focus on relevant information, we removed several columns deemed unnecessary for our purposes: 'Indirizzo', 'Codice Istat comune', 'Codice ACE', 'Meteo Temperatura', 'Precipitazioni', 'Condizioni meteo', and 'Hdop'. Additionally, since each file contained data from multiple days, we filtered out records to retain only those corresponding to the reference day for each file. After this step, each file contained approximately 800,000 records.

Given that our primary objective with this dataset is to estimate the free-flow speed, we implemented an additional filtering process. For each day, we retained only data recorded during nighttime hours (from 10:00 PM to 6:00 AM). This restriction is based on the premise that traffic flow is generally lower during these hours, making speeds more reflective of free-flow conditions. To apply this filter, we utilized the Datetime column, which contains timestamp information, to isolate records within the specified time window.

The final result of this preprocessing is a collection of 102 files, each containing approximately 60,000 records, ready for further analysis.

Furthermore, since we needed to process the data in QGIS, we combined the 102 individual files into seven aggregated datasets because of Excel limits (1 million records), each one containing data from one or more months, we can see them in Figure 17.

HCD_01_2019.csv	30/12/2024 11:19	Microsoft Excel Co...	45,689 KB
HCD_02-03_2019.csv	30/12/2024 11:21	Microsoft Excel Co...	57,468 KB
HCD_04-05_2019.csv	30/12/2024 11:22	Microsoft Excel Co...	68,816 KB
HCD_06-07_2019.csv	30/12/2024 11:23	Microsoft Excel Co...	72,138 KB
HCD_08-09_2019.csv	30/12/2024 11:23	Microsoft Excel Co...	66,706 KB
HCD_10-11_2019.csv	30/12/2024 11:24	Microsoft Excel Co...	65,164 KB
HCD_12_2019.csv	30/12/2024 11:24	Microsoft Excel Co...	28,070 KB

Figure 17: Aggregated HCD files, per month, per hourly interval, sorted by month

The naming format here is "HCD_mm.yyyy.csv" where mm stands for the month period and yyyy for the year. They consist of about 6 million records.

4.2 HCD data matching process to the graph

To perform the map-matching process, we followed a series of structured steps.

4.2.1 Data cleaning and selection

The initial step involved importing the above mentioned files introduced in 4.1 *HCD data pre-processing* into QGIS. It is important to note that the coordinates in the original files are expressed in the WGS 84 geographic coordinate system (EPSG:4326), which uses latitude and longitude in degrees. Since the subsequent analysis requires distances to be expressed in meters, the first step was to reproject the points into a projected coordinate system, specifically EPSG:32632 (WGS 84 / UTM zone 32N). This system, based on the Universal Transverse Mercator (UTM) projection for Zone 32N in the Northern Hemisphere, utilizes the WGS 84 datum and represents coordinates in meters within a flat planar system, enabling precise distance calculations.

Furthermore, given that the extent of the graph was smaller than the distribution of the points, we created a polygon encompassing the graph (see Section 3.1 *Study context: Turin*) and excluded all points located outside of this boundary as you can see in Figure 18.

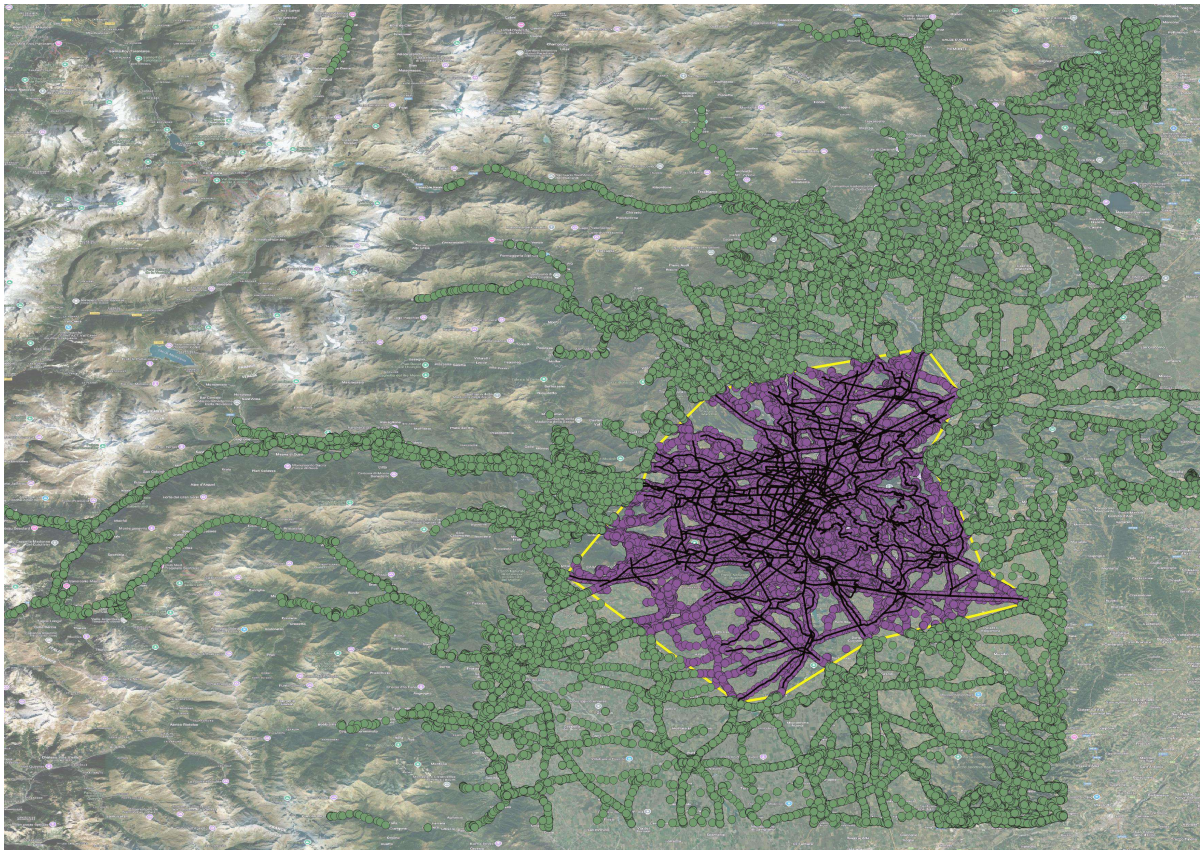


Figure 18: Maintained points (violet) and deleted points (green)

The yellow lines represent the polygon boundary, outside of which the points were removed. Once this pre-processing step was completed, we created new files containing only the violet points in Figure 18 displayed in Figure 19 by exporting them from Qgis to .csv and proceeded with the actual map-matching.








 in_polygon_01.csv	30/12/2024 16:54	Microsoft Excel Co...	33,819 KB
 in_polygon_02-03.csv	30/12/2024 16:54	Microsoft Excel Co...	43,298 KB
 in_polygon_04-05.csv	30/12/2024 16:55	Microsoft Excel Co...	52,501 KB
 in_polygon_06-07.csv	30/12/2024 16:56	Microsoft Excel Co...	52,066 KB
 in_polygon_08-09.csv	30/12/2024 16:56	Microsoft Excel Co...	46,740 KB
 in_polygon_10-11.csv	30/12/2024 16:56	Microsoft Excel Co...	49,022 KB
 in_polygon_12.csv	30/12/2024 16:56	Microsoft Excel Co...	20,370 KB

Figure 19: .csv files with only maintained points after selection process

Those files, since they contain less records than the files showed in Figure 17, have been further aggregated in 3 files, which are "in_polygon_jan-apr.csv", "in_polygon_may-aug.csv" and "in_polygon_sep-dec.csv" consisting of 3,108,779 points (respectively 1,014,135 records, 1,046,812 records and 1,047,832 records).

4.2.2 Online map matching with ORS

The first approach we employed utilized an online map-matching service provider, specifically Open Route Service (ORS)⁸. ORS returns a list of points snapped to the nearest edge in the routing graph taken from OpenStreetMap⁹ (OSM). In case an appropriate snapping point cannot be found within the specified search radius, "null" is returned. In the latter case we managed to maintain the same coordinates. We developed a Python script, detailed in Appendix [1], to facilitate this process. The script begins by reading the files "in_polygon_jan-apr.csv", "in_polygon_may-aug.csv" and "in_polygon_sep-dec.csv" one at a time, introduced at the end of the section 4.2.1 *Data cleaning and selection* and sending the coordinates to the ORS API. Using road data from OSM, the API¹⁰ then snaps each GPS point to the nearest road edge based on three main criteria:

- Minimum distance: each point is associated to the closest arc;
- Transport mode: in this case, the API is configured for the option "driving-car", so it only considers the network suitable for vehicles;
- Direction and topology: the API takes into account the direction and the topology of the road to place correctly the point on the segment;

Following this step, the script extracts corrected coordinates for each point. If the API fails to generate corrected coordinates, the respective fields are left empty, and we retain the original coordinates from the dataset. The new coordinates are appended as additional columns to the same file with the same structure and information as the original. The generated files are "in_polygon_aftercut_afterpython_jan-apr.csv", "in_polygon_aftercut_afterpython_may-aug.csv" and "in_polygon_aftercut_afterpython_sep-dec.csv" with same number of records as before, but with improved coordinates.

⁸<https://openrouteservice.org/dev/#/api-docs/v2/snap/profile/post>

⁹<https://www.openstreetmap.org/exportmap=9/45.064/8.042>

¹⁰An application programming interface (API) is a connection between computers or between computer programs. In contrast to a user interface, which connects a computer to a person, an application programming interface connects computers or pieces of software to each other.

	A	B	C	D	E	F	G	H	I
	Trip_id	Device_Id	Datetime	Latitudine	Longitudin	Tipologia	SpeedKmh	corrected_longitude	corrected_latitude
2	6022293121	7277245	08/05/2019 00:00:02	44.96998	7.681228	A	89	7.681227	44.969962
3	6022293121	7277245	08/05/2019 00:00:42	44.97234	7.668662	E	90	7.668653	44.972324
4	6022293121	7277245	08/05/2019 00:01:22	44.97585	7.656648	E	89	7.656605	44.975796
5	6022293121	7277245	08/05/2019 00:02:02	44.98142	7.646197	A	90	7.646152	44.981385
5	6022293121	7277245	08/05/2019 00:02:42	44.9876	7.636528	A	90	7.63648	44.987561
7	6022293121	7277245	08/05/2019 00:03:22	44.99424	7.627483	A	90	7.627477	44.99424
3	6022293121	7277245	08/05/2019 00:04:02	45.0028	7.622577	A	90	7.622544	45.002794
3	6022293121	7277245	08/05/2019 00:04:42	45.00963	7.615048	A	90	7.615054	45.009669
0	6022293121	7277245	08/05/2019 00:05:22	45.01339	7.60365	A	90	7.603621	45.013371
1	6022293121	7277245	08/05/2019 00:06:02	45.02184	7.598468	A	89	7.598464	45.021836
2	6022293121	7277245	08/05/2019 00:10:48	45.0377	7.566027	E	17	7.566014	45.0377
3	6022293121	7277245	08/05/2019 00:06:42	45.02972	7.591655	A	89	7.591616	45.029705

Figure 20: Output of Python script, with corrected coordinates

4.2.3 Snapping on QGIS

The second step involved further processing in QGIS. We imported the last three files "in_polygon_aftercut_afterpython_jan-apr.csv", "in_polygon_aftercut_afterpython_may-aug.csv" and "in_polygon_aftercut_afterpython_sep-dec.csv" introduced at the end of section 4.2.2 *Online map matching with ORS* containing the corrected coordinates and refined the map-matching results. Since the ORS API utilized a graph derived from OpenStreetMap, which differs from our graph (whose description is given in 3.4 *Graph of Turin*), the points were not perfectly aligned. The difference is given by the fact that our graph contains far less arcs with respect to the OSM graph, and moreover, since not all the links contained data on travel speed or the technical classification of the road, sometimes elements coming from the BDTRE graph have been embedded, as described in [14]. To resolve the above mentioned lack of alignment, we used the 'Snap Geometries to Layer' tool in QGIS, snapping points to the nearest road segment within a small tolerance (around 3 meters) to ensure points remained on major roads represented in our graph (otherwise, especially in central urban area, the points on minor roads would have been anchored to the graph). This adjustment was not only for visual consistency but also essential for associating each point with its corresponding road segment.

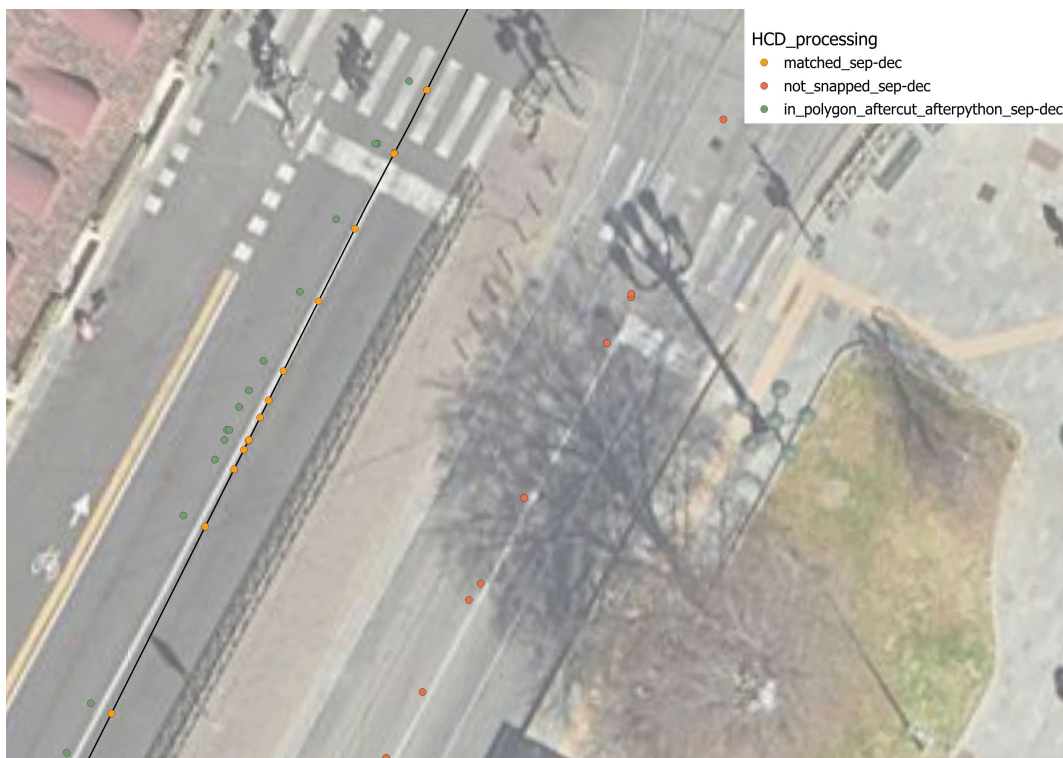


Figure 21: Results of snapping process

Given the differences between the OSM graph and our graph, not all the points can be associated to the graph's arch, as it can be seen in Figure 21 and as we will better discuss in the next section. We can say the process is more or less based on a point-to-curve geometric algorithm, based on what is described in [12].

4.2.4 Association of arc's corresponding attributes

The final step in the map-matching process was to correctly associate each point with the corresponding road segment. We used the 'Join Attributes by Nearest' function in QGIS to create new shapefiles. These files retained the original point geometries but enriched the attribute table with data from the graph, providing vehicle and road segment information for each point. We generated two shapefiles for each temporal interval, "matched_jan-apr.shp" and "not_snapped_jan-apr.shp", "matched_may-aug.shp" and "not_snapped_may-aug.shp" and "matched_sep-dec.shp" and "not_snapped_sep-dec.shp" among which only the matched ones were later exported as .csv files with the same name but different extension clearly. The "matched" file contains all points successfully associated with our graph, in detail, they contain respectively 674,581 points, 684,859 points and 678,806 points. Therefore, about 65% of the HCD points selected in Section 4.2.1 *Data cleaning and selection* have been matched.

In this dataset, a new column, 'Arco' was added to store the identifier of the arc on which each point is located. Conversely, the "not_snapped" file includes the points that were excluded from the matching process. These points were likely associated with other roads, as their positions were too distant from our network to be reliably mapped. In Figure 22 we display an example of the output file after the map matching process.

Trip_id	Device_Id	Datetime	Tipologia	SpeedKmh	L [m]	Main_Clas:Vf[km/h]	C [ve/h]	N_Corsie	Arco	East	North	
5.64E+09	5116563	14/01/2019 00:00:01 U		64	940.04	primary	59	4000	2	833	406030.906	4996741.767
5.64E+09	4321532	14/01/2019 00:00:02 U		24	280.67	tertiary	43	5850	2	3043	387010.175	4992120.926
5.64E+09	4436464	14/01/2019 00:00:05 U		45	378.44	secondary	59	2000	2	5714	391789.078	4994848.854
5.64E+09	3980498	14/01/2019 00:00:09 E		71	1426.76	primary	61	2000	2	5958	392621.315	4994157.456
5.64E+09	2461876	14/01/2019 00:00:10 U		54	700.76	tertiary	30	5844	2	1929	396013.395	4994281.142
5.64E+09	3935170	14/01/2019 00:00:11 E		85	1968.59	primary	65	4000	2	6771	395153.122	4976755.449
5.64E+09	4321532	14/01/2019 00:00:12 U		50	280.67	tertiary	43	5850	2	3043	387017.617	4992130.837
5.64E+09	4420399	14/01/2019 00:00:17 U		52	230.34	secondary	39	6816	2	3456	396882.866	4993606.156
5.64E+09	4198436	14/01/2019 00:00:24 U		43	354.89	tertiary	43	4426	2	108	387137.742	4992611.023
5.64E+09	5827484	14/01/2019 00:00:25 E		79	553.93	primary	73	4000	2	5963	395094.548	4980607.719
5.64E+09	5073833	14/01/2019 00:00:26 U		42	21.4	tertiary	45	6512	2	1646	397313.231	4995162.242
5.64E+09	4293470	14/01/2019 00:00:27 A		78	2858.92	motorway	97	3600	3	5913	385329.128	4993064.474

Figure 22: Matched points after map matching process, from "matched_jan-apr.csv"

We note in particular the most important features, thus the coordinates, expressed in the projected reference system WGS84/UTM 32N, the identifier of the corresponding arc under the column 'Arco' and the timestamps under the column 'Datetime'.

4.3 HCD data cleaning of matched points

With the matched points dataset ready, we needed to further refine the data to support our thesis analysis. Specifically, we focused on identifying and excluding the following cases:

- Cases in which there is only 1 point of the same deviceId on the same arc, because it is a useless information that could fake our analysis;
- Cases in which the vehicle's speed is zero;
- Cases in which the same deviceId passes through the same arc more than once in few minutes.

Since our objective is to accurately measure travel time on each road segment, these cases could compromise the validity of our results. In the first case, a single point does not provide any information about travel time. In the second case, we risk to get distances from points equal to zero, due to the fact that the vehicle is not moving along time. Instead, in the third case, repeated traversals in a short period could skew our analysis.

To address the first scenario, we implemented an R script (see Appendix 2) that counts the occurrences of each deviceId on each road segment during the same day. For example, if deviceId *5116563* appears only once on segment *833*, this point is removed from the dataset. We applied this script to the three files "matched_jan-apr.csv", "matched_may-aug.csv" and "matched_sep-dec.csv" introduced in 4.2.4 *Association of arc's corresponding attributes* and get the same file, with a new column reporting the above mentioned count, as it is showed in Figure 23.

Device_Id	Datetime	L [m]	Arco	East	North	counts
5116563	2019/01/14 00:00:01	940.04	833	406030.906	4996741.77	1
4321532	2019/01/14 00:00:02	280.67	3043	387010.175	4992120.93	2
4436464	2019/01/14 00:00:05	378.44	5714	391789.078	4994848.85	1
3980498	2019/01/14 00:00:09	1426.76	5958	392621.315	4994157.46	1
2461876	2019/01/14 00:00:10	700.76	1929	396013.395	4994281.14	1
3935170	2019/01/14 00:00:11	1968.59	6771	395153.122	4976755.45	2
4321532	2019/01/14 00:00:12	280.67	3043	387017.617	4992130.84	2
4420399	2019/01/14 00:00:17	230.34	3456	396882.866	4993606.16	1
4198436	2019/01/14 00:00:24	354.89	108	387137.742	4992611.02	2

Figure 23: Output after counting script applied on R, from "matched_jan-apr.csv"

We have hidden some useless column referring to the scope of this representation because otherwise the image would have been too large. In every file, we added one more sheet ('filtered') in which we filtered and maintained only the records corresponding with count>1 strictly. In that way we only maintained the vehicles which are recorded on the same arc at least two times and solved the first issue. To do this, we had to create three files with extension .xlsx named respectively "matched_jan-apr.xlsx", "matched_may-aug.xlsx" and "matched_sep-dec.xlsx" with respectively 172,550 records, 189,000 records and 173,000 records.

After completing this process, and considering the significant number of records that were deleted, we simplified the workflow by merging all three files into a single consolidated file named "merged_matched_processed.xlsx", which now contains a total of 534,988 records.

Then we filtered out and removed all the records corresponding to a speed equal to zero.

To handle the second scenario, we developed a more complex yet intuitive approach. We first sorted the data by Day, Device.Id and Datetime, creating a chronological sequence of each point's movements for each day.

Device_Id	Datetime	SpeedKmh	L [m]	Main_Clas	Arco	East	North	Giorno	counts	diff_arco	condition	test
2822152	2019/01/14 03:45:43	87	1617.26	motorway	7226	394180.2	4997702	14/01/2019	2	201	no	FALSO
2822152	2019/01/14 03:56:28	89	1637.94	motorway	7427	406540	5002813	14/01/2019	2	0	si	FALSO
2822152	2019/01/14 03:56:49	89	1637.94	motorway	7427	406958.4	5003098	14/01/2019	2	-1331	no	FALSO
2822159	2019/01/14 05:17:32	76	1328.67	tertiary	6096	385923.1	4986433	14/01/2019	2	-29	no	VERO
2822159	2019/01/14 05:19:30	65	1723.24	tertiary	6067	387079.8	4986814	14/01/2019	2	0	si	FALSO
2822159	2019/01/14 05:19:33	65	1723.24	tertiary	6067	387115.9	4986809	14/01/2019	2	889	no	FALSO
2822159	2019/01/14 05:35:30	0	593.23	tertiary	6956	393855.7	4989426	14/01/2019	2	0	si	FALSO
2822159	2019/01/14 05:35:51	15	593.23	tertiary	6956	393866.4	4989435	14/01/2019	2	-860	no	FALSO
2822159	2019/01/14 22:18:04	63	1328.67	tertiary	6096	385893.6	4986717	14/01/2019	2	-5765	no	VERO
2822159	2019/01/14 22:23:58	0	126.22	tertiary	331	386522.3	4984508	14/01/2019	2	0	si	FALSO
2822159	2019/01/14 22:26:31	0	126.22	tertiary	331	386518.3	4984509	14/01/2019	2	6918	no	FALSO
2822165	2019/01/14 05:10:32	85	3175.18	motorway	7249	392420.8	4982538	14/01/2019	2	0	si	FALSO

Figure 24: Cronological movements of each vehicle, from "merged_matched_processed.xlsx"

If we look at the sequence of movements of Device.Id 2822159 in Figure 24, we can notice that it passes through the arc 6096 two times in the same day, but during two different hourly intervals, this could significantly distort travel time calculations, so we have to exclude these cases.

Considered the big dataset, we cannot proceed manually, so what we did, basically, has been to create a new column reporting the difference between the current and previous arc ('diff_arco'). For each row, if we see a 0 value, it means the vehicle is still on the same arc, if you see a certain numeric value it means the Device.Id has changed arc. More in the detail, if you see a numeric value and then a 0 value, the vehicle has changed arc and then it is travelling on the last one, but the cases in which there are two consecutive numeric values it means the arc x has been visited, then the vehicle has been on another arc y and of course, sooner or later, the vehicle passed again through arc x because we remind that we only maintained the cases in which the same arc has been visited at least two times by the same vehicle. Looking to the 'diff_arco' column corresponding to the data displayed in Figure 24 we can see what we just said in Figure 25:

Device_Id	Datetime	SpeedKmh	L [m]	Main_Clas	Arco	East	North	Giorno	counts	diff_arco	condition	test
2822152	2019/01/14 03:56:49	89	1637.94	motorway	7427	406958.4	5003098	14/01/2019	2	-1331	no	FALSO
2822159	2019/01/14 05:17:32	76	1328.67	tertiary	6096	385923.1	4986433	14/01/2019	2	-29	no	VERO
2822159	2019/01/14 05:19:30	65	1723.24	tertiary	6067	387079.8	4986814	14/01/2019	2	0	si	FALSO
2822159	2019/01/14 05:19:33	65	1723.24	tertiary	6067	387115.9	4986809	14/01/2019	2	889	no	FALSO
2822159	2019/01/14 05:35:30	0	593.23	tertiary	6956	393855.7	4989426	14/01/2019	2	0	si	FALSO
2822159	2019/01/14 05:35:51	15	593.23	tertiary	6956	393866.4	4989435	14/01/2019	2	-860	no	FALSO
2822159	2019/01/14 22:18:04	63	1328.67	tertiary	6096	385893.6	4986717	14/01/2019	2	-5765	no	VERO
2822159	2019/01/14 22:23:58	0	126.22	tertiary	331	386522.3	4984508	14/01/2019	2	0	si	FALSO
2822159	2019/01/14 22:26:31	0	126.22	tertiary	331	386518.3	4984509	14/01/2019	2	6918	no	FALSO

Figure 25: Difference column with sequence of movements of Device_Id 2822159

To explain how we automatize the process: under the 'condition' column we plotted "si" if the difference is 0, otherwise "no". Under 'test' column we plotted 'FALSO' if there are two consecutive 0 or one number and a 0 and 'VERO' when there are two consecutive numbers. Given what we said, the points corresponding to 'VERO' should be removed from our dataset. We carried out this process in the 'filtered' sheet of the same excel file.

Once did this, only the last sheet of the file has been exported in .csv "merged_matched_processed.csv" with 464,338 records left, and consequently imported in QGIS. Now we do not have anymore points which could fake the analysis.

4.4 Computation of distances on the graph

Now, the following step has been to compute the distances between two consecutive points with the same ID, on the same arc during same day. The complexity here is related with the fact that we don't want to calculate the straight distance between the points but rather we want to calculate the distance following the geometry of the road. To do this, we had to develop a Python script (see Appendix A [4]), in which there are 2 main steps:

- Organizing points by Day and Device_Id on the same arc;
- Computation of distance between points as the difference between the distance between origin node of the arc and point x_{i+1} and the distance between the origin node of the arc and point x_i .

What we get is one Excel file, named 'distances.csv' structured as it is showed in Figure 26.

Giorno	Device_Id	Arco	Datetime1	Datetime2	Speed1	Speed2	Distance	Arc_Length	Distance_to_Start	Distance_to_End
14/01/2019	1061892	5725	14/01/2019 04:58:43	14/01/2019 04:59:28	79	78	1024.268742	3258.138759	2469.840062	1812.567439
14/01/2019	1061892	5725	14/01/2019 04:59:28	14/01/2019 05:00:10	78	78	1062.922492	3258.138759	1445.57132	2875.489931
14/01/2019	1061892	6414	14/01/2019 04:55:08	14/01/2019 04:55:56	56	70	1005.748971	5768.152942	4506.174147	256.2298247
14/01/2019	1147869	5953	14/01/2019 05:41:45	14/01/2019 05:42:46	79	80	1430.922195	2815.211214	231.0097204	1153.279299
14/01/2019	1147869	5953	14/01/2019 05:42:46	14/01/2019 05:43:31	80	69	1010.531343	2815.211214	1661.931915	142.7479565
14/01/2019	1360431	7237	14/01/2019 22:00:29	14/01/2019 22:01:51	50	62	1399.687728	1528.111827	12.21114299	116.2129562
14/01/2019	1835924	6421	14/01/2019 22:22:18	14/01/2019 22:26:04	22	28	2320.288508	2982.813159	158.4193487	504.1053026
14/01/2019	1835924	6616	14/01/2019 22:27:35	14/01/2019 22:28:43	60	60	1176.791532	1815.234751	535.8497065	102.5935127

Figure 26: Output file after execution of Python code to calculate distances among points

This is the meaning of the column displayed in Figure 26:

- 'Giorno' is the day of the acquisition;

- 'Device_Id' is the identifier of the vehicle;
- 'Arco' is the identifier of the arc;
- 'Datetime1' is the timestamp of the first point on the arc;
- 'Datetime2' is the timestamp of the second point on the arc;
- 'Speed1' is the instantaneous speed of the first point;
- 'Speed2' is the instantaneous speed of the second point;
- 'Distance' is the distance between the two points following the geometry of the graph;
- 'Arc_Length' is the total length of the arc;
- 'Distance_to_Start' is the distance between the origin node of the arc and the first point;
- 'Distance_to_End' is the distance between the second point and the end node of the arc;

All the distances are expressed in meters. Now, the next step will be related with the computation of the travel time of each arc by means of the calculated distances.

4.5 Filtering out HCD observations with service stops

Before proceeding with the main analysis, it is important to note the presence of certain points with a significantly large temporal gap. This likely occurs in situations where, for example, a vehicle remains on the same arc, parks in a nearby parking area, and then resumes its journey after a considerable time, such as 20 minutes. Since the vehicle remains associated with the same arc, the algorithm calculates the physical distance between these points without recognizing that they correspond to distinct temporal intervals.

Therefore, we decided to eliminate certain records based on a temporal criterion. Specifically, we calculated the difference between `Datetime2` and `Datetime1` in a new column 'delta.time' and plotted the distribution with a bin of 1 minute.

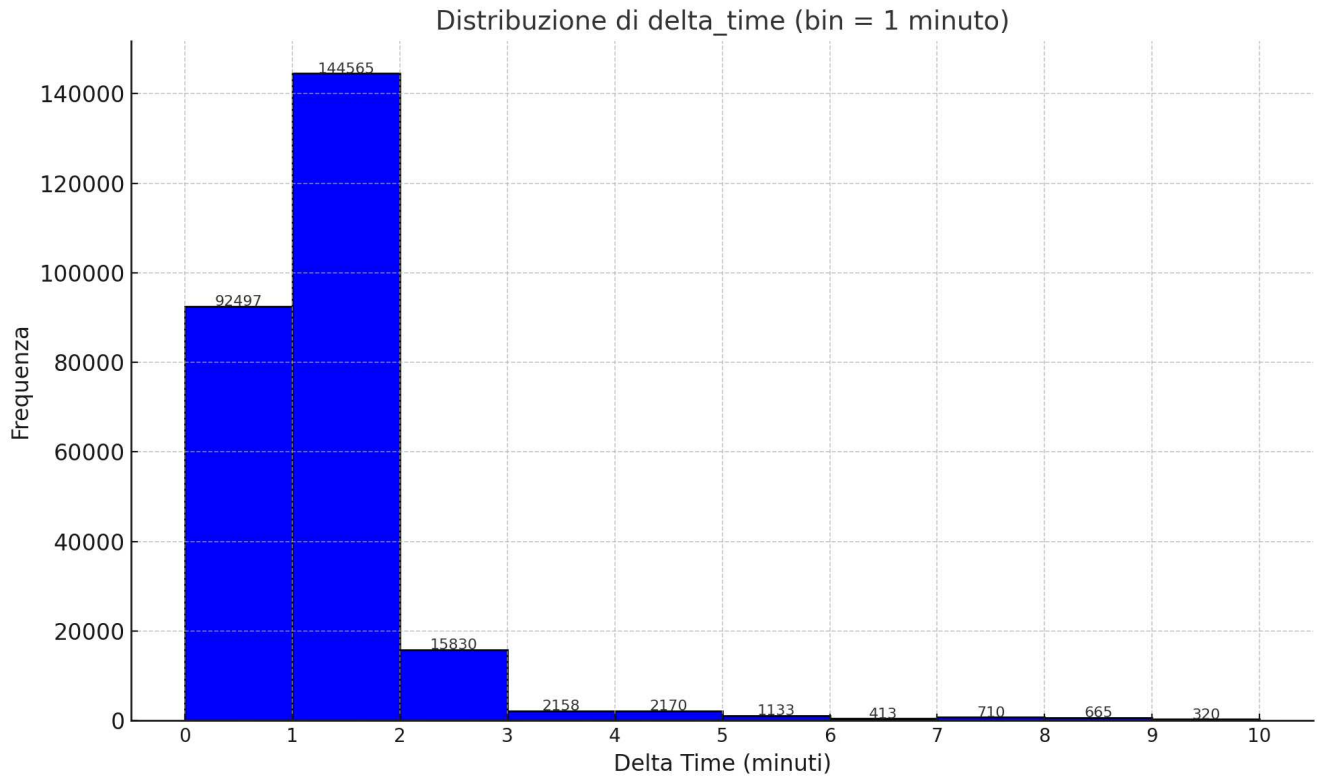


Figure 27: Distribution of delta_time from 'distances.csv'

As we can notice in Figure 27, the vast majority delta_time is concentrated between 0 and 4 minutes. To establish a statistical basis, we calculated the 95th percentile, which is 3.78 minutes, and excluded 19,682 records with a temporal distance greater than 4 minutes (rounded up to the nearest integer). Moreover, as it has been done in [15], our threshold is a bit higher than that established there, thus 2 minutes, which is also the maximum duration of traffic light cycle. As a result, these records were removed from the dataset.

Additionally, we deleted the points whose physical distance was zero, because useless for our purposes. These points have been further removed from the file "merged_matched_processed.csv" with 434,815 records left.

4.6 Statistics on the number of HCD observations on each arc

Before proceeding with the work, we wonder if the level of detail of the graph that we used was or not appropriate to the dataset we have. We want highlight we started with a dataset consisting on about six million points and now, after the processes described above, we have about four hundred thirty thousand points in the file "merged_matched_processed.csv" introduced in 4.3 *HCD data cleaning of matched points*. At this point we implemented two kind of analysis on R. We calculated, by means of a R script, displayed on Appendix [3], how many visits and how many Device_Id on each arc, getting the excel files 'arco_stats.xlsx' presented in the Figure 28.

	A	B	C	D
1	Arco	visits	visits_deviceId	average
2	2	2	1	2.00
3	3	12	5	2.40
4	9	38	5	7.60
5	11	2	1	2.00
6	13	2	1	2.00
7	15	6	3	2.00
8	23	4	2	2.00
9	24	2	1	2.00
10	29	21	10	2.10
11	30	2	1	2.00
12	31	245	96	2.55
13	35	9	3	3.00
14	38	8	4	2.00
15	41	37	15	2.47

Figure 28: Arcs statistics

We report the meaning of each column:

- Arco, it's the arc identifier;
- visits, it's the total number of vehicles recorded on the arc;
- visits_device_id, is the total number of deviceId on that arc;
- average, is calculated as the ratio between 'visits' and 'visits_deviceId'.

We have to mention we have data on 7093 arcs out of 7549, thus we can perform our analysis on 93% of the arcs.

4.6.1 Distribution of the average number of observations per device

We then reported the distribution of the column 'average', for each arc, displayed in Figure 29.

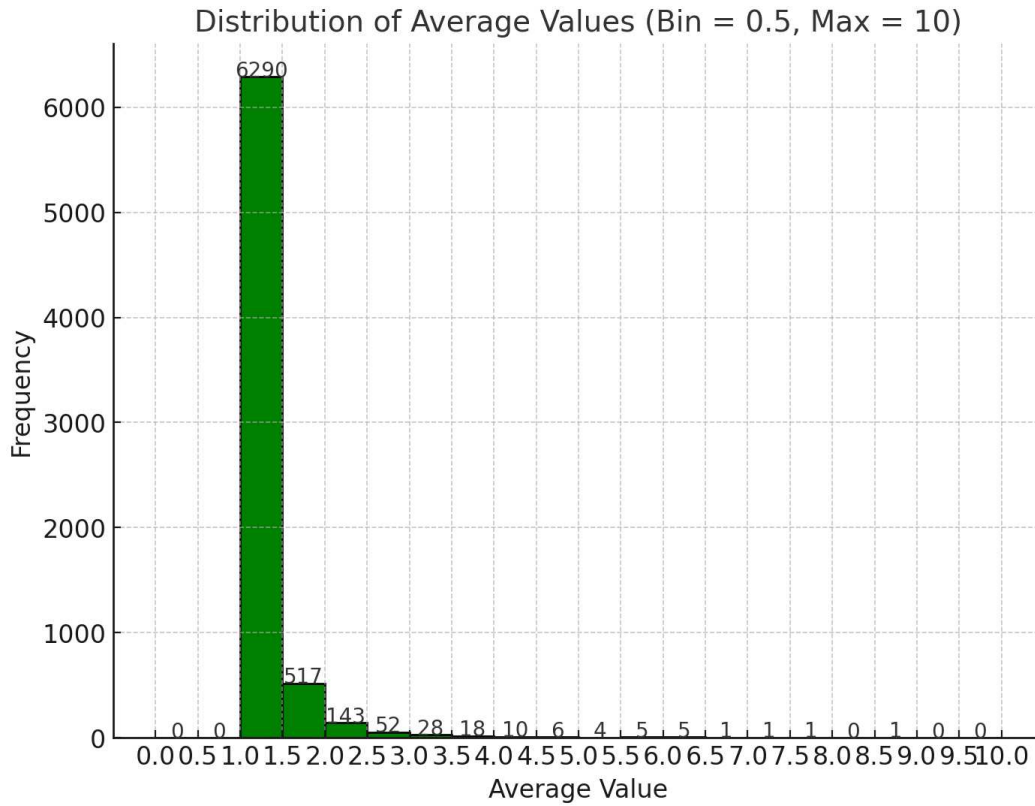


Figure 29: Distribution of 'average' column, from 'arco_stats.xlsx'

We can notice frequencies are high for lower values. It means most devices cross an arc only a few times on average. These arcs are used by a wide variety of vehicles, rather than being dominated by repeat visits from a few vehicles.

4.6.2 Distribution of the visits

We also plotted the distribution of the column 'visits.device.id' which represents the number of different vehicles visiting each arc in Figure 30.

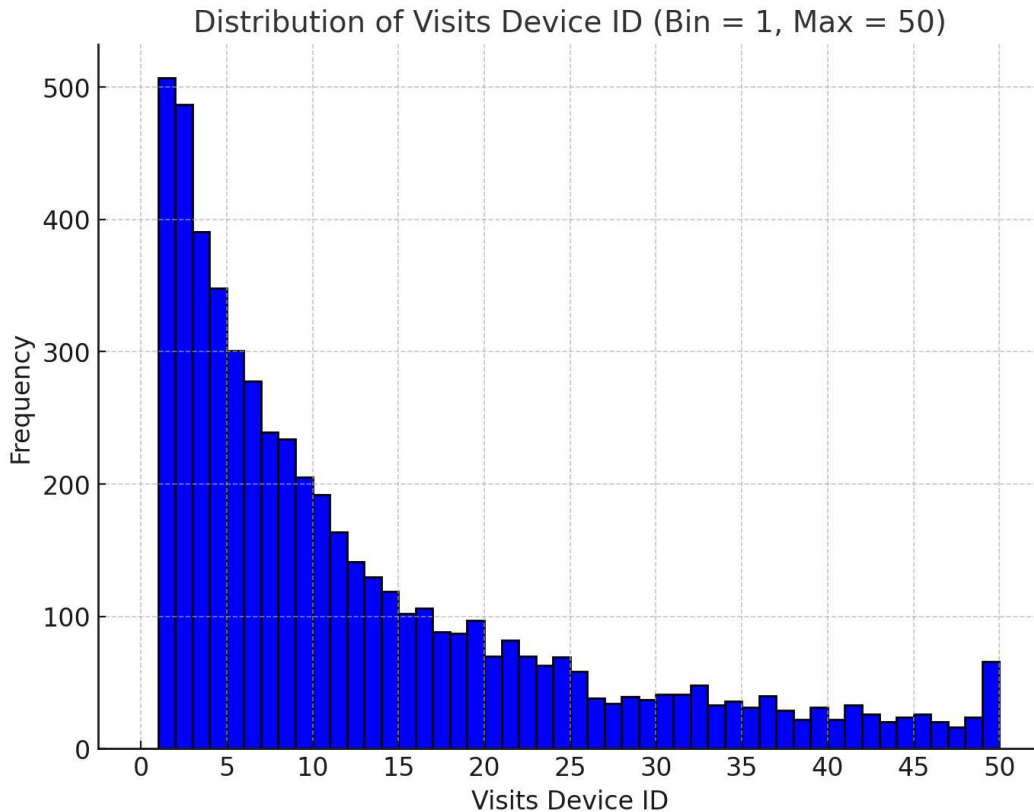


Figure 30: Number of deviceId visiting arcs distribution

Despite the high frequency of lower values, indicating that many arcs are visited by only a few devices, a substantial number of arcs are visited by 10 or more devices. Specifically, out of 7083 arcs:

- 4093 arcs are visited by more than 10 devices.
- 2990 arcs are visited by fewer than 10 devices, and among these, 1257 arcs are visited by 5 or more devices.

This suggests that the dataset is well-balanced, with a significant portion of arcs showing diverse usage. The distribution, which is right censored, confirms the quality of the data, providing a solid foundation for further analysis.

To sum up, we started with 6,491,019 points in the file showed in Figure 17 and through a series of structured steps involving selection of points inside the area covered by the graph in 4.2.1 *Data cleaning and selection*, data processing described in 4.2.4 *Association of arc's corresponding attributes*, and 4.3 *HCD data cleaning of matched points* we finally ended up with 464,337 points. Only the 8% of points maintained but still a good result because we have data on 93% of arcs.

5 Methodology, part 2: travel times derivation

In this chapter, we will outline the analysis process applied to the dataset, focusing on estimating the free-flow travel time and the observed travel time for each arc.

5.1 Free-flow travel time computation

In this paragraph we will focus on the estimation of the travel time under ideal conditions. We consider two alternatives to perform this task:

1. Exploit the data we already have on each arc of the graph dataset;
2. Develop a new method based on [19];

To ensure clarity, we will perform both processes and compare their results to choose the most appropriate method. It should be noted that the first one is based on an official data source, as it will be apparent in the following subhead, and therefore it could be considered the best solution. However, this data source is only available for the Piedmont region. Consequently, we will also explore the second method that is only making use of the above introduced HCD dataset, and it is therefore generalisable to any study area where such data are available.

5.1.1 Free flow travel time using previous information from the graph

In the dataset we already described in 3.4 *Graph of Turin* we have a column with the free flow speed on each arc 'Vf [kmh]'. The process to get this speed is accurately explained in [14] (paragraph 5.2). We report a brief explanation.

The estimation of free-flow speed, denoted as Vf, is derived from analyzing data collected on the road network¹¹ available on Geoportal of region Piemonte using an approach that integrates detailed information about segment lengths and instantaneous speeds recorded by moving vehicles. Speed data, in the dataset BDTRE, which are crucial for this calculation, are obtained through GPS devices or tracking systems that monitor the position and travel time along various segments of the network. Each arc in the network represents a road or a portion of it, consisting of multiple segments. For each segment, data on its length, travel time, and average speed are available.

To calculate Vf, in [14] data are first aggregated at the arc level by considering all its constituent segments. The weighted average speed of an arc, required for estimating Vf, is calculated using a ratio that accounts for the segment lengths and their respective travel times. This method assigns greater weight to longer segments, ensuring that the overall speed estimate represents the entire road accurately. However, to isolate the value of Vf, which represents the speed under free-flow conditions, only data collected during periods of low or absent traffic are considered. This filtering excludes the effects of congestion or delays, ensuring that the estimate reflects solely the road's performance under ideal conditions.

The formula for calculating Vf is implemented in QGIS, through an expression that combines segment lengths and instantaneous speeds (we remind that those instantaneous speed are taken from BDTRE dataset). This implementation automates the aggregation process, handling any missing data and ensuring that only valid segments contribute to the calculation. The result is a theoretical speed allowing smooth, uninterrupted movement, a critical parameter for comparing

¹¹https://www.geoportale.piemonte.it/geonetwork/srv/ita/catalog.searc/metadata/r_piemon : 2bb551d2 - bad8 - 488f - 9070 - 07f5a65b5f11

observed conditions with ideal ones and supporting more in-depth analyses of traffic and road network planning.

Once understood where Vf comes from, we created a .csv file named "points_processed.csv" (see Figure 31), after having applied to the already mentioned file "merged_matched_processed.csv" in the paragraph 4.3 *HCD data cleaning of matched points* a Python script (see Appendix [5]) to determine the list of all the points in the dataset, with their relative position on the arc, and their instantaneous speed, reporting:

- Device_Id;
- Arco;
- distance_to_start already mentioned in 4.4 *Computation of distances on the graph*;
- speed, thus the instantaneous speed of each vehicle recorded;
- Datetime;
- arc_length;
- Vf_kmh described here above;
- Main_Class, thus the class of the arc, whose meaning has been explained in Table 6;

Device_Id	Arco	distance_to_start	speed	Datetime	arc_length	Vf_kmh	Main_Class
1057014	6196	1684.275305	80	01/08/2019 03:33:43	3429.679	66	secondary
1057014	6196	2706.690441	54	01/08/2019 03:34:57	3429.679	66	secondary
1063872	6249	560.0930261	104	01/08/2019 04:41:04	2419.491	73	primary
1063872	6249	1601.426043	74	01/08/2019 04:41:48	2419.491	73	primary
1065138	2885	183.8207753	63	01/08/2019 04:36:26	1705.673	50	primary
1065138	2885	1199.839188	67	01/08/2019 04:37:22	1705.673	50	primary

Figure 31: List of points to be processed to determine free flow speed

Then, the travel time is simply determined as the ratio between the arc length and the aforementioned speed, under the column 'TT_ffs (min)' displayed in Figure 32 which has been added to the aforementioned file "points_processed.csv".

5.1.2 Free flow travel time based on HCD observations [19]

The alternative approach involves utilizing the HCD dataset at hand, following the method outlined in paragraph 2.3 *Travel time estimation under free flow conditions*. The concept focuses on performing a weighted average of all recorded speeds for each arc, where the weights are determined by the position of the vehicle on the arc. Specifically, the closer the vehicle is to the centre of the arc, the higher its speed's reliability, as it is presumed to be less affected by congestion on adjacent arcs. Conversely, the closer the vehicle is to the extremities of the arc, the lower its speed's reliability, as it is considered more likely to be influenced by congestion or delays originating from neighbouring arcs. This method is reasonable to apply, as we recall from the discussion in Section 4.1 *HCD data pre-processing* that all the data we have pertains to an hourly interval between 10:00

PM and 6:00 AM. In fact, we assume that there is no congestion during this interval that could have influenced the data. This is crucial because the objective is to determine the free flow speed.

The column 'distance_to_start' report the distance of the point with respect to the origin node of the arc, while the column 'arc_length' is reporting the length of the arc expressed in meters. As it is described in [19] we calculated the ratio between the two above cited columns, according to the Equation 26:

$$\theta = \frac{\text{distance_to_start}}{\text{arc_length}} \quad (26)$$

Then, based on the concept already explained above, the weight is calculated according to Equation 27:

$$w = 1 - |2 * \theta - 1| \quad (27)$$

So that, when the vehicle is approximately near to the centre of the arc ($\theta = 0.5$) the weight is maximum, while if θ is far from 0.5, the weight is low because the vehicle is considered to be influenced from the adjacent arcs. Now, we have the weight of the points based on their position and the instantaneous speed, so that we can proceed applying the formula showed in Equation 20. The result is a column, named 'TT_weighted (min)' displayed in Figure 32 in which we performed the ratio between the arc length and the speed calculated in Equation 20.

Device_Id	Arco	distance_to_start	speed	Datetime	arc_length	Vf_kmh	Main_Class	TT_ffs (min)	θ	ω	weighted_mean	TT_weighted (min)
2818606	2	12.4528378	37	28/06/2019 05:17:09	632.5272	75	motorway	0.51	0.019687	0.039375	61.06843888	0.62
2818606	2	477.3571956	63	28/06/2019 05:17:40	632.5272	75	motorway	0.51	0.754682	0.490635	61.06843888	0.62
5144954	9	40.37908337	37	01/08/2019 03:27:44	425.8024	73	primary	0.35	0.094831	0.189661	31.48787134	0.81
5144954	9	56.2216706	2	01/08/2019 03:32:26	425.8024	73	primary	0.35	0.132037	0.264074	31.48787134	0.81
5247074	9	325.3601963	61	12/07/2019 05:53:57	425.8024	73	primary	0.35	0.764111	0.471778	31.48787134	0.81
5247074	9	48.06697548	39	12/07/2019 05:57:44	425.8024	73	primary	0.35	0.112886	0.225771	31.48787134	0.81
5126636	9	24.20506474	51	13/07/2019 04:37:25	425.8024	73	primary	0.35	0.056846	0.113692	31.48787134	0.81
5126636	9	23.29619023	2	13/07/2019 04:37:38	425.8024	73	primary	0.35	0.054711	0.109423	31.48787134	0.81

Figure 32: Output of elaboration of free flow speed with the two different methodologies, from the file "points_processed.csv"

5.1.3 Selection of the most appropriate free flow travel time and comparison between the two different methodologies

Now, we need to construct a dataset in which each arc is associated with the travel time under free flow conditions, calculated using the two methodologies described above. Before proceeding, it is essential to highlight three important aspects:

1. **Missing Data:** we do not have data for all arcs in the dataset. Therefore, in cases where no data is available, we will apply the methodology described in 5.1.1 *Free flow travel time using previous information from the graph*;
2. **Analysing Results:** results obtained from the methodology described in 5.1.2 *Free flow travel time based on HCD observations [19]* must be analysed to determine the conditions under which these results can be considered valid;
3. **Preference for the Second Methodology:** When both methodologies are applicable according to the above two points and reliable based on the subsequent considerations, we will always prefer the second one. This preference is due to its development based on our available data, ensuring reproducibility in diverse contexts. In fact, this approach remains

applicable even in scenarios where local datasets, such as BDTRE, are unavailable, as long as we have access to similar HCD data;

To understand which methodology is better to be choose, we followed a sequence of structured steps:

- Creation of an Excel file, in which we associate to each arch, the corresponding travel time developed with the two above described methodologies, named "archi_fftt.xlsx";
- Determination of speed from the above calculated travel times;
- Determination of speed limits based on the road class, under the columns 'Main_Class';
- Difference between speed determined with both methodologies and speed limits, expressed in percentage with respect to the limit;
- Selection of the methodology with the lower difference;

Concerning the limits, we associated the limit based on the functional class of the road, taken from OpenStreetMap '<https://wiki.openstreetmap.org/w/index.php?title=IT:Key:highway&oldid=2222720>'.

As it is mentioned in the website the class is only assigned based on functional characteristics and not considering the administrative classification or the owner. Here we report the limits:

- motorway¹²:120 km/h;
- trunk¹³:100 km/h;
- primary¹⁴: 90 km/h;
- secondary¹⁵:90 km/h;
- tertiary¹⁶:50 km/h;
- unclassified¹⁷:50 km/h;
- residential¹⁸:30 km/h;

The limits for each road category, as determined by those present, are specified in DM 6792/2001 [8].

The result is a file excel 'archi_fftt.xlsx' structured as it is showed in Figure 33:

¹²<https://wiki.openstreetmap.org/wiki/IT:Tag:highway%3Dmotorway>

¹³<https://wiki.openstreetmap.org/wiki/IT:Tag:highway%3Dtrunk>

¹⁴<https://wiki.openstreetmap.org/wiki/IT:Tag:highway%3Dprimary>

¹⁵<https://wiki.openstreetmap.org/wiki/IT:Tag:highway%3Dsecondary>

¹⁶<https://wiki.openstreetmap.org/wiki/IT:Tag:highway%3Dtertiary>

¹⁷<https://wiki.openstreetmap.org/wiki/Tag:highway%3Dunclassified>

¹⁸<https://wiki.openstreetmap.org/wiki/IT:Tag:highway%3Dresidential>

Arco	Main_Class	Limits	arc_length	TT_ffs (min)	TT_weighted (min)	Vf	V_weighted	deltaVf	deltaV_weighted	FFTT (min)
1	tertiary	50	101.03	0.080824	#N/D	75	NA	50.0%	NA	0.08
2	motorway	120	632.66	0.506128	0.621460662	75	61.08126	37.5%	49.1%	0.51
3	motorway	120	680.78	0.510585	#N/D	80	NA	33.3%	NA	0.51
4	tertiary	50	48.49	0.042785294	#N/D	68	NA	36.0%	NA	0.04
5	secondary	90	34.44	0.031309091	#N/D	66	NA	26.7%	NA	0.03
6	secondary	90	33.96	0.029530435	#N/D	69	NA	23.3%	NA	0.03
7	secondary	90	33.54	0.029165217	#N/D	69	NA	23.3%	NA	0.03
8	secondary	90	31.55	0.028681818	#N/D	66	NA	26.7%	NA	0.03
9	primary	90	425.91	0.350063014	0.811364577	73	31.495829	18.9%	65.0%	0.35

Figure 33: Free flow travel time of each arc, from 'archi_fftt.xlsx'

Where 'V_f' (already introduced in Section 5.1.1 *Free flow travel time using previous information from the graph*) and 'V_{weighted}' have been calculated respectively:

$$V_f = \frac{arc_length}{TT_{ffs}} \quad (28)$$

$$V_{weighted} = \frac{arc_length}{TT_{weighted}} \quad (29)$$

Then, the column 'deltaVf' and 'deltaV_{weighted}' represent the percentage variation of the speed with respect to the limit. Calculated as follows:

$$deltaVf = \frac{V_f - Limits}{Limits} \quad (30)$$

$$deltaV_{weighted} = \frac{V_{weighted} - Limits}{Limits} \quad (31)$$

The final column, 'FFTT' is the free flow travel time associated to each arch, chosen as the travel time corresponding to the minimum between 'deltaVf' and 'deltaV_{weighted}'.

Moreover, we wanted to analyse the difference between the two methodologies with respect to the arc length. In particular we plotted the difference between the two columns 'Vf' and 'V_weighted' showed in Figure 34 getting the column 'delta_speed' and represented the scatter plot.

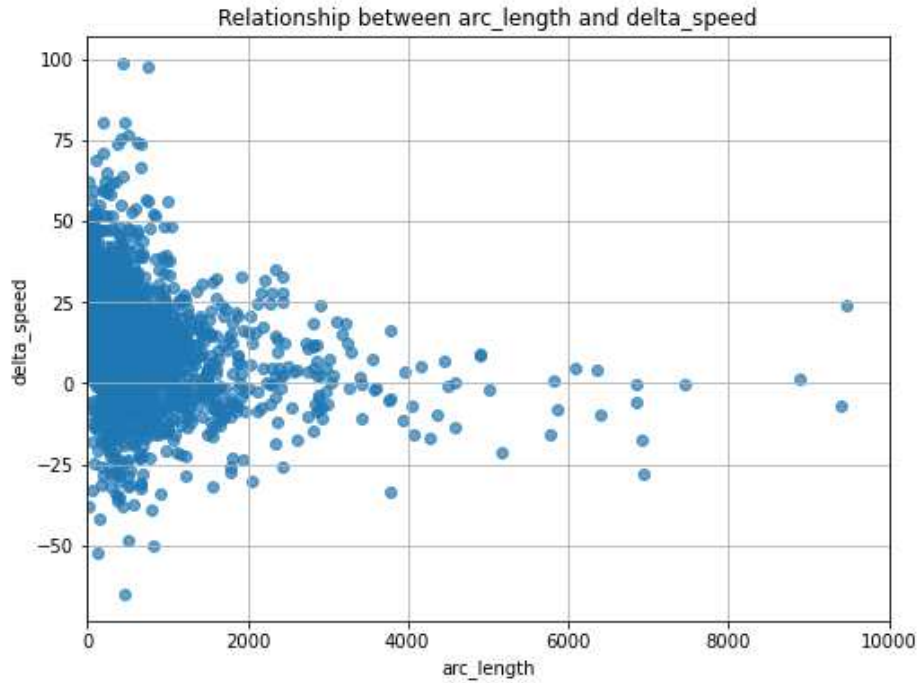


Figure 34: Relationship between 'delta_speed' and 'arc_length', from 'archi_fftt.xlsx'

We can see there is a big variance when the arc_length is low. This is likely due to the significant influence of small changes in travel times, which have a more pronounced effect on smaller arcs. Whereas, as the length of the arcs increases, delta_speed tends to decrease and become more stable. This behaviour aligns with the fact that longer arcs incorporate a greater amount of data, reducing the influence of local anomalies.

By narrowing the focus to arcs with a maximum length of 200 meters, as shown in Figure 35, we can delve deeper into the analysis to uncover more specific patterns and relationships within this subset of data.

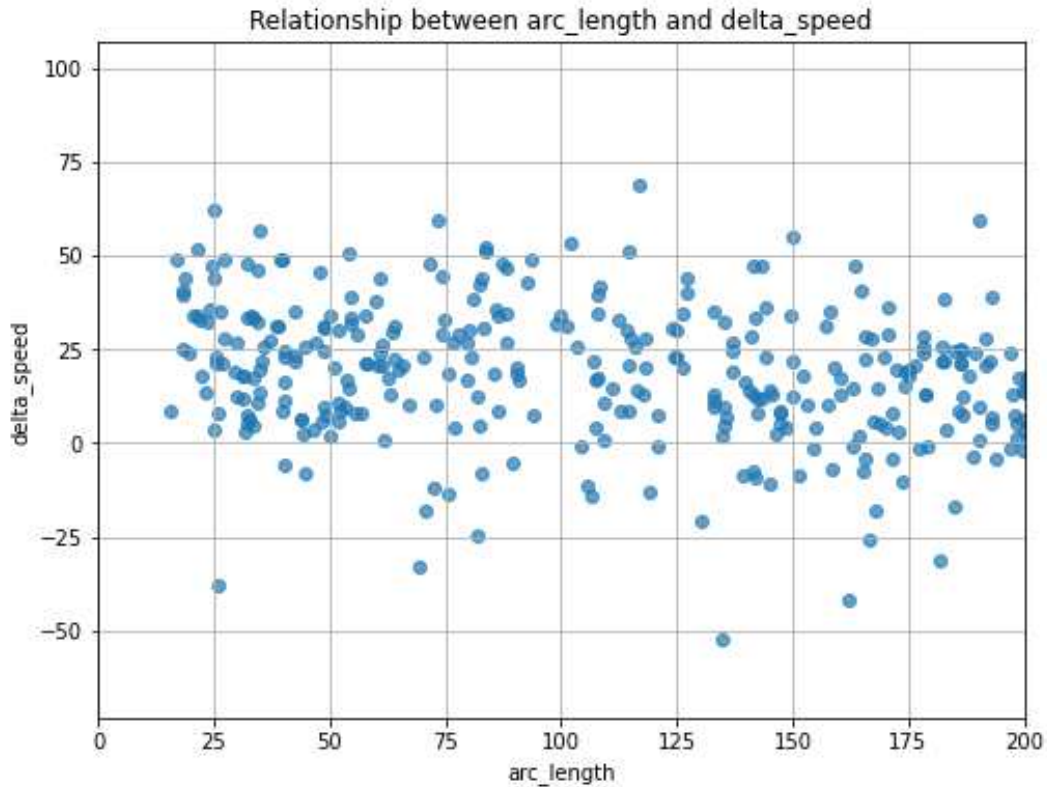


Figure 35: Relationship between 'delta_speed' and 'arc_length' (max=200m), from 'archi_fftt.xlsx'

The majority of values are positive, since delta_speed is the difference between the speed derived from methodology presented in 5.1.1 *Free flow travel time using previous information from the graph* and the speed derived methodology presented in 5.1.2 *Free flow travel time based on HCD observations [19]*, we understand the former, when the arc_length is low, is often higher. Since we used, as it is mentioned in 4.1 *HCD data pre-processing* data collected only during nighttime, when traffic is supposed to be absent, this can be due to several reasons, like:

- Nighttime road conditions, e.g., insufficient lighting, reduced visibility, precautionary slowing by drivers for safety reasons;
- Infrastructure or geometric reasons: short arcs (<200m) often include features that require speed reduction, such as curves, intersections, or speed bumps, which become even more relevant at night.
- Nighttime environmental conditions: if the data was collected during periods with adverse weather conditions (e.g., rain, fog), drivers may have slowed down compared to the reference speed.

5.2 Observed travel time computation from the FCD dataset

5.2.1 FCD dataset manipulation and selection of observed days

Once we established the free flow travel time of each arch, we are ready to meet the real purpose of this thesis, already introduced in 1.1 *Thesis objectives*, thus to calculate the increment of travel time in Floating Car Data (introduced in 3.3.3 *Floating Car Data (FCD)*) due to congestion, by

comparing these latter with the free flow time calculated by means of HCD. Here as well, we developed two approaches to carry out the analysis:

1. Calculate the travel time by means of a simple proportional calculus, in a high disaggregate fashion;
2. Calculate the travel time for each arc by computing the weighted average of the travel times recorded by all vehicles that traversed that arc (more an aggregate fashion);

Here, we have to apply the same processes already introduced in advance in 4 *Methodology, part 1: HCD data processing and map matching*, to the Floating Car Data, to get a dataset which can be processed. Based on the contingency table shown in Figure 16, we identified 120 potential days for analysis. From these, we selected 4 representative days to ensure computational feasibility. Importantly, all selected days fall within a standard working period during school time, which is when the majority of movements have taken place. The selected days are represented in Table 8:

Table 8: Selected day for FCD data manipulation

Date	Day
24/09/2019	Tuesday
18/10/2019	Friday
20/11/2019	Wednesday
12/12/2019	Thursday

We provide a brief description of the input files and the final processed files.

We started with 'merged_FCD_hh-hh_ddmm.csv' where hh-hh stands for the hourly interval and dd and mm are respectively the day and the month. Five hourly intervals have been selected for each day: 00-06,06-10,10-16,16-20,20-24; since we have four days we have a total of 20 files. The structure of each file is like the one presented in Table 5. Moreover, based on the deviceId and dateTime, we were able to associate each record to one trip, whose details are presented in Table 3, and therefore we added the trip characteristics to each record whenever possible. Unfortunately, we could not make this for each deviceId, because there are some missing trips and moreover, the trip characteristics that we initially omitted will be useful especially when performing analysis of results in 6 *Results*. To make this association, we had to apply a python script (see Appendix 8), getting 20 additional files named 'merged_FCD_hh-hh_ddmm+.csv'.

We had initially 8,614,394 records. After the data manipulation described in detail in 4.2.1 *Data cleaning and selection*, there are 5,814,050 records remaining. After processes described in 4.2.2 *Online map matching with ORS*, 4.2.3 *Snapping on QGIS*, 4.2.4 *Association of arc's corresponding attributes* and 4.3 *HCD data cleaning of matched points*, we have obtained the final files 'matched_processed_ddmm.csv', for each day, containing a total of 1,138,855 records. To these, we applied the python script reported in Appendix [4] and get the files 'distances_processed_ddmm.csv', for each day, that have been further merged into a new file 'distances_processed.csv'. The structure is therefore different with respect to that presented in 4.4 *Computation of distances on the graph*.

This is the list of columns of 'distances_processed.csv':

- 'Giorno' is the day of the acquisition;
- 'deviceId' is the identifier of the vehicle;

- 'Arco' is the identifier of the arc;
- 'Datetime1' is the timestamp of the first point on the arc;
- 'Datetime2' is the timestamp of the second point on the arc;
- 'Speed1' is the instantaneous speed of the first point;
- 'Speed2' is the instantaneous speed of the second point;
- 'Distance' is the distance between the two points following the geometry of the graph;
- 'Arc_Length' is the total length of the arc;
- 'Distance_to_Start1' is the distance between the origin node of the arc and the first point;
- 'Distance_to_Start2' is the distance between the origin node of the arc and the second point;
- 'Distance_to_End1' is the distance between the first point and the end node of the arc;
- 'Distance_to_End2' is the distance between the second point and the end node of the arc;
- Arc length;
- Main_Class;
- Trip_id;
- Datetime_p, thus the starting timestamp of the trip;
- Datetime_a, thus the arrival timestamp of the trip;
- Lat_parten;
- Lon_parten;
- Lat_arrivo;
- Lon_arrivo;
- Type;
- km_percors;
- speedKmh_y;
- Sesso;
- Età_intest;

The presence of those 2 additional columns 'Distance_to_End1' and 'Distance_to_End2', is needed for the proportional calculus presented in 5.2.2 *Vehicle level travel time derivation*.

5.2.2 Vehicle level travel time derivation

The subsequent analysis process is relying on a proportional calculation. Here, we relate the distance travelled within the given time interval to the portion of arc travelled so far. We have to distinct two cases:

1. Only two points on the same arc;
2. More than two points on the same arc;

For clearness, we could say this is a general formula in the first case:

$$TT_{i,j} = \frac{Arc_Length_j}{Distance_{t,t+1}} * delta_time \quad (32)$$

Where i represent the deviceId, j the arc, t and $t+1$ the initial and successive timestamps, respectively, and delta_time denote the difference between these two consecutive timestamps. In the second case, we adopted a more sophisticated approach. Consider the simplest case in which we have 3 points on the same arc, as it is showed in Figure 36:

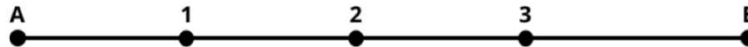


Figure 36: Three points on the same arc

Where A and B are the origin and ending node of the arc j , while 1,2,3 are the points of same deviceId successively recorded on the same arc at the instant t , $t+1$ and $t+2$ respectively. In this case, considering the structure of the Excel file 'distances_processed.csv', we will have two distinct rows. The first one representing timestamps and distance between points 1 and 2, the second one representing distance and timestamps between points 2 and 3.

Considering only the row relative to points 1 and 2:

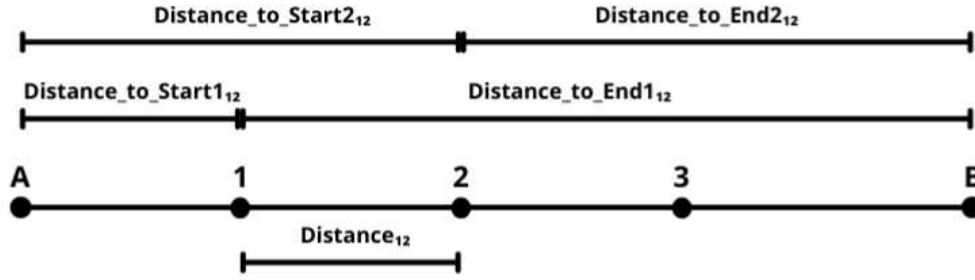


Figure 37: Notation on the arc considering points 1 and 2

Looking at Figure 37 we can now introduce the formula to calculate the travel time between node A and point 2:

$$T_{A2} = \frac{Distance_to_Start2_{12}}{Distance_{12}} * delta_time_{12} \quad (33)$$

Now considering the record relative to points 2 and 3:

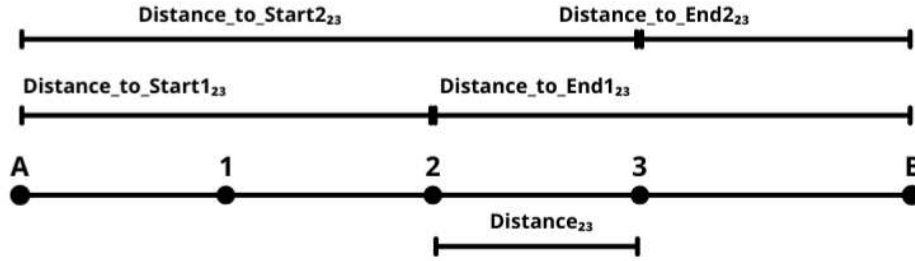


Figure 38: Notation on the arc considering points 2 and 3

Looking at Figure 38 we can now introduce the formula to calculate the travel time between point 3 and node B:

$$T_{3B} = \frac{Distance_to_End1_{23}}{Distance_{23}} * delta_time_{23} \quad (34)$$

So that, at the end, the total travel time needed to travel the arc AB is:

$$TT_{i,AB} = T_{A2} + T_{3B} \quad (35)$$

This has been done for all the deviceId on the same arc. The problem was, how to generalize this method for each arc, since we can have more than three points on the same arc. So we implemented an approach exploiting Excel to automatize the calculus. To make understand this, we have to proceed step by step.

Firstly, we grouped and sorted the excel file 'distances-processed.csv' by 'Giorno', 'deviceId' and 'Arco'.

Then we calculated in a new column named 'diff_arco' the difference of the arc identifier of two consecutive rows. From this, we can understand when there are zeroes, it means there are more than two points on the same arc.

Then, in a new column named 'test' we implemented a formula which returns 'FALSO' if the record is relative to only two points on the same arc and 'VERO' if there are consecutive rows representing more than two points.

Therefore, next to the last column, we created the column 'tt_duepunti' in which, if the previous column 'test' was 'FALSO' we calculated travel time with the simple proportional calculus presented in the first part of this section, otherwise it returns zero value and we have to continue because it means there are at least three points.

So that, we created a column 'progr' where we plotted 1 if, considering column 'test', the value of the previous record was 'FALSO' and the value of the current one was 'VERO'; 2 if the value of the current is 'VERO' and the value of the successive is 'FALSO'; "NA" in the other cases. This, because, if we remind the meaning of column 'test', when we have 'FALSO' and then 'VERO' it means the arc has changed so the current record is representing the first two points on the same arc. When, instead, the current is 'VERO' and the next one is 'FALSO', it means the current record is representing the last two points recorded on that arc.

In the middle if there are more than 3 points we have always "NA". In this way, the structure is always like: 1, NA,NA,NA...,2; to detect same deviceId on consecutive arcs and implement the last presented proportional calculus. To make better understand, we report the structure in Figure 39 :

deviceId	Arco	Datetime1	Datetime2	Distance	Arc_Length	Distance_to_Start1	Distance_to_Start2	Distance_to_End1	Distance_to_End2	delta_time	diff_arco	test	tt_duepun	progr
2597048	6962	15/01/2020 16:15:43	15/01/2020 16:15:54	114.9421264	1189.464823	351.1243395	466.0664658	838.3404832	723.3983568	11	0	VERO	0	NA
2597048	6962	15/01/2020 16:15:54	15/01/2020 16:16:11	145.4652362	1189.464823	466.0664658	611.531702	723.3983568	577.9331206	17	0	VERO	0	NA
2597048	6962	15/01/2020 16:16:11	15/01/2020 16:17:01	258.5086067	1189.464823	611.531702	870.0403087	577.9331206	319.4245139	50	0	VERO	0	NA
2597048	6962	15/01/2020 16:17:01	15/01/2020 16:17:12	176.0687485	1189.464823	870.0403087	1046.109057	319.4245139	143.3557654	11	416	VERO	0	2
2597048	7378	15/01/2020 16:11:53	15/01/2020 16:12:39	203.0781737	610.351431	220.1733976	423.2515713	390.1780334	187.0998597	46	-5719	FALSO	138.253	NA
2597089	1659	15/01/2020 08:51:27	15/01/2020 08:52:06	231.5862032	630.7763212	526.555951	294.9697478	104.2203702	335.8065734	39	4470	FALSO	106.2251	NA
2597089	6129	15/01/2020 08:58:01	15/01/2020 08:58:37	331.0346004	1421.269788	1375.722935	1044.688335	45.5468532	376.5814536	36	0	VERO	0	1
2597089	6129	15/01/2020 08:58:37	15/01/2020 08:58:57	143.8574672	1421.269788	1044.688335	900.8308677	376.5814536	520.4389208	20	0	VERO	0	NA
2597089	6129	15/01/2020 08:58:57	15/01/2020 08:59:15	196.7486689	1421.269788	900.8308677	704.0821787	520.4389208	717.1876097	18	0	VERO	0	NA
2597089	6129	15/01/2020 08:59:15	15/01/2020 08:59:17	22.62745524	1421.269788	704.0821787	681.4547235	717.1876097	739.8150649	2	0	VERO	0	NA
2597089	6129	15/01/2020 08:59:17	15/01/2020 09:00:35	459.0052409	1421.269788	681.4547235	222.4494826	739.8150649	1198.820306	78	0	VERO	0	NA
2597089	6129	15/01/2020 09:00:35	15/01/2020 09:00:55	148.4421725	1421.269788	222.4494826	74.00731012	1198.820306	1347.262478	20	512	VERO	0	2

Figure 39: Result of described process above, from 'distances_FCD.xlsx'

Successively, we created a new column 'Ti' where we implemented the formula presented in Equation 33 if 'progr' was 1, or that presented in Equation 34 if 'progr' was 2. If the value was "NA", we simply used the observed delta_time between the two consecutive points of that row.

Finally, in the column 'tt_disaggregate' we grouped and summed up the value of the column 'Ti' corresponding to the same deviceId on the same arc in the same day.

The values in this last column, will be then compared with the free flow travel times in the next section to get results and make final considerations.

5.3 Statistics on the number of FCD observations on each arc

As we did in 4.6 *Statistics on the number of HCD observations on each arc*, we aim to determine the number of observations available for each arc to assess whether the resulting data can be considered representative of real traffic conditions. As mentioned in 1.1 *Thesis objectives*, FCD will be processed to derive observed travel times, helping to identify the most critical arcs. However, since our analysis is based on only four days of data due to computational constraints, we need to evaluate the reliability of these results. A higher number of observations per arc indicates greater reliability in the findings. We created a file, starting from 'distances_processed.csv' already introduced in 5.2.1 *FCD dataset manipulation and selection of observed days*, named 'arco_stats_FCD.xlsx', in which we have the structure shown in Figure 40:

Arco	visits	visits_deviceid	average
1	4	3	1
2	22	18	1
3	25	19	1
4	2	2	1
5	1	1	1
7	4	4	1
8	1	1	1
9	38	31	1

Figure 40: Statistics on arcs, from 'arco_stats_FCD.xlsx'

We report the meaning of each column:

- Arco, it's the arc identifier;
- visits, it's the total number of vehicles recorded on the arc;
- visits_device.id, is the total number of different deviceId on that arc;
- average, is calculated as the ratio between 'visits' and 'visits_deviceId'.

We have data on 4675 arcs out of 7549, thus we can perform our analysis on 62% of the arcs.

5.3.1 Distribution of the visits

We plotted the distribution of the column 'visits_device_id', under the name of 'idcount' which represents the number of vehicles visiting each arc in Figure 41.

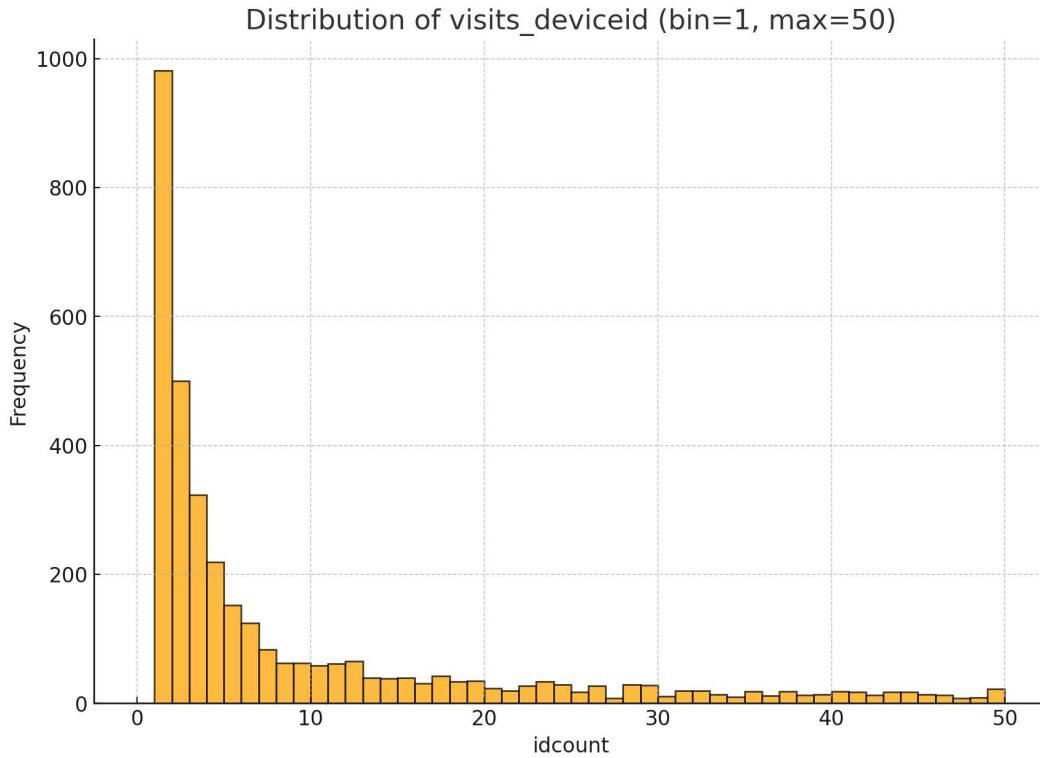


Figure 41: Number of deviceId visiting arcs distribution

Despite the high frequency of lower values, indicating that many arcs are visited by only a few devices, a pretty high number of arcs are visited by 5 or more devices. Specifically, out of 4675 arcs:

- 2561 arcs are visited by more than 5 devices, of which 2169 have been visited more than 10 devices.
- 2114 arcs are visited by fewer than 5 devices.

The distribution, which is right censored, confirms the quality of the data, providing a solid foundation for further analysis.

6 Results

This chapter presents the computation of the results derived from the application of the methodology presented in the preceding chapters. The aim is to synthesize and quantify time lost in congestion by processing the data we got so far, with different levels of aggregation according to the thesis goals mentioned at the beginning. Specifically, this work focuses on calculating both disaggregate and aggregate data to reveal the most congested arcs and zones.

6.1 Increase in travel times by vehicle and by arc

By means of the files cited above 'archi_fftt.xlsx', 'distances_processed.xlsx' introduced respectively in 5.1.2 *Free flow travel time based on HCD observations [19]* and in 5.2 *Observed travel time computation from the FCD dataset*, we created a new excel file summarizing the most important features of each arc, named 'results_vehicle_level.xlsx' consisting of 342,817 records. It uses travel times calculated in 5.2.2 *Vehicle level travel time derivation* and compare them with free flow travel time of each arch. In Figure 42 we can see the structure.

deviceld	Arco	arc_length	tt_disaggregate	FFTT (s)
5222326	1	101.03	7.63	4.85
5202512	2	632.66	33.49	30.37
3232221	3	680.78	31.73	30.64
5119488	3	680.78	20.61	30.64
5214733	3	680.78	53.89	30.64
5225881	3	680.78	6.00	30.64
5239711	3	680.78	30.42	30.64
2809133	9	425.91	34.33	21.00

Figure 42: Structure of 'results_vehicle_level.xlsx'

It can be noted that the arc identifier may sometimes be repeated because the analysis involves multiple vehicles traversing the same arc. Thus the analysis is disaggregated by vehicle and by arc.

To quantify the impact of traffic on travel times, we created a new column 'Wasted_time_vehicle' in which we calculated the difference between 'tt_disaggregate' and 'FFTT (s)' to get the time wasted during rush hours with respect to the reference time (calculated, as we remind from 5.1.3 *Selection of the most appropriate free flow travel time and comparison between the two different methodologies*, during off peak hours).

$$Wasted_time_vehicle = tt_disaggregate - FFTT(s) \quad (36)$$

Moreover, we express the time wasted in terms of percentage rather than absolute numbers, so we calculated the percentage as:

$$Wasted_percentage = \frac{Wasted_time_vehicle}{FFTT(s)} \quad (37)$$

And added the new column named 'Wasted_percentage'.

deviceId	Arco	Arc_Length	tt_disaggregate	FFTT (s)	Wasted_time_vehicle	Wasted_percentage
5222326	1	101.01	7.63	4.85	2.78	57%
5202512	2	632.53	33.49	30.37	3.12	10%
4377936	2	632.53	28.46	30.37	-1.91	0%
3232221	3	680.63	31.73	30.64	1.10	4%
5119488	3	680.63	20.61	30.64	-10.03	0%
5214733	3	680.63	53.89	30.64	23.26	76%
5225881	3	680.63	6.00	30.64	-24.64	0%

Figure 43: Structure of 'results_vehicle level.xlsx', with percentage of time wasted for each vehicle

Figure 43 reports the final structure of the file. We can notice the presence of negative values, meaning that the vehicle is not wasting time due to congestion. In fact, when a negative value is detected, the relative percentage of time wasted plotted is zero.

In Figure 44 we plot the distribution of the column 'Wasted_time_vehicle'.

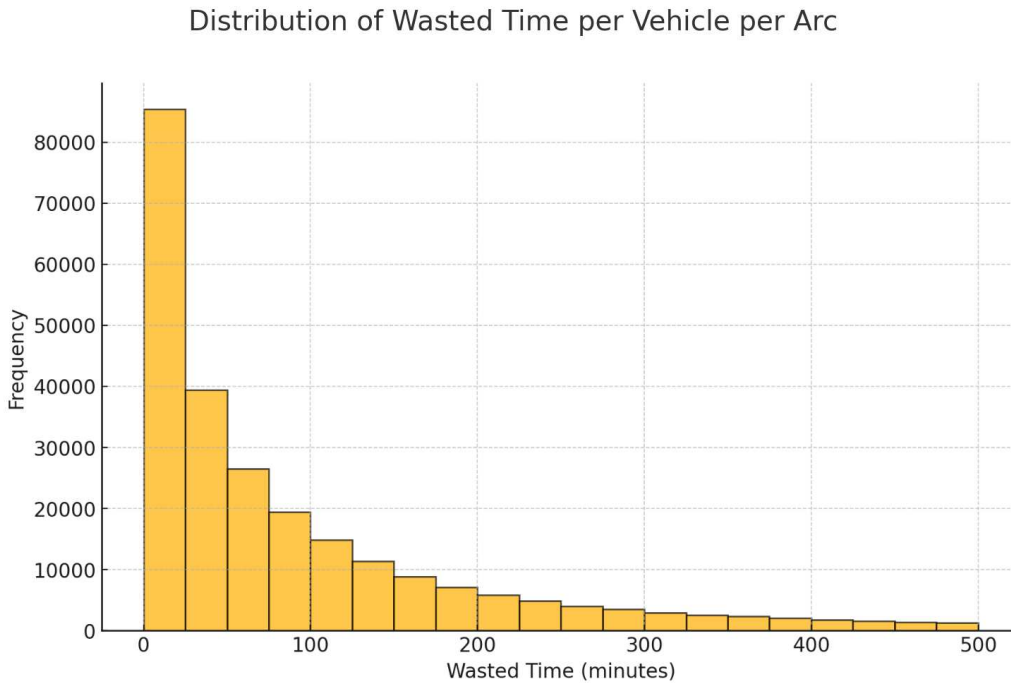


Figure 44: Distribution of column 'Wasted_time_vehicle', from 'results_vehicle level.xlsx'

The distribution shows that most of vehicles experience relatively low congestion times, indicating that the network generally operates efficiently for the majority of users. However, there are notable instances of significantly higher delays, which point to potential bottlenecks or localized issues in the system.

In Figure 45 we plot the distribution of the column 'Wasted_percentage'.

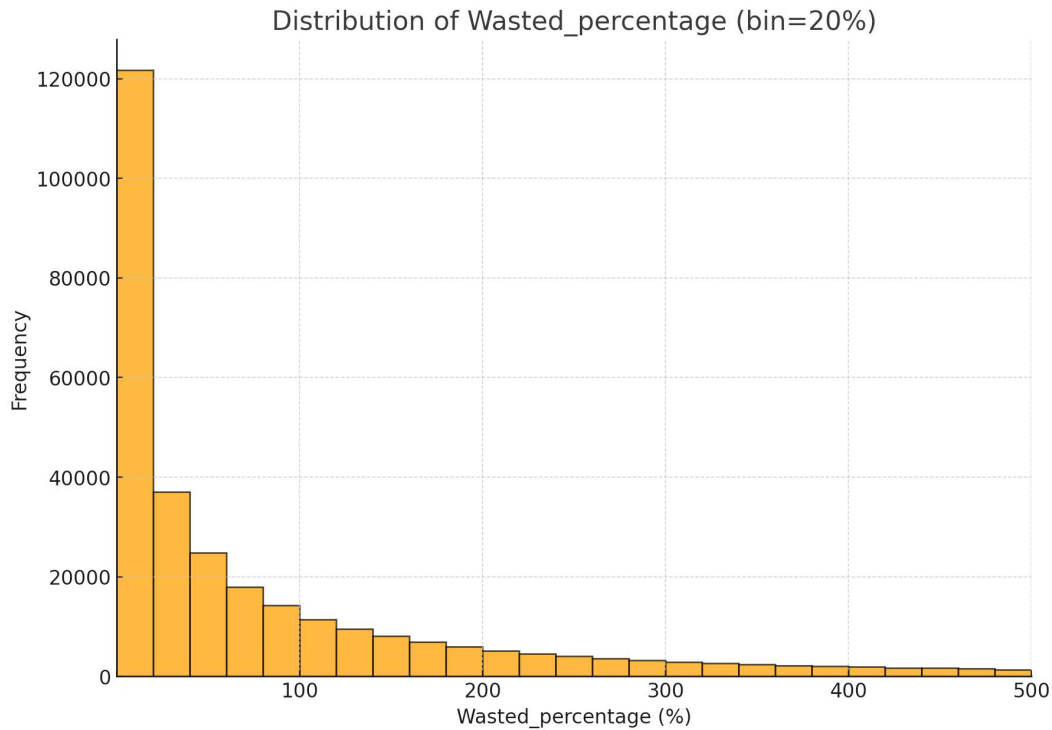


Figure 45: Distribution of column 'Wasted_percentage', from 'results_vehicle_level.xlsx'

The data suggest a skewed distribution where most vehicles are not seriously affected by congestion, but a smaller subset (less than 50% of the total number of vehicles) suffers from significantly higher percentages of wasted time. These extreme cases could point to specific bottlenecks, localized congestion, or unique conditions such as traffic incidents or infrastructure limitations.

6.2 Comparison of data availability during peak hours vs entire-day data

Moving forward, our analysis will distinguish between peak hours and full-day data. We have defined a relatively broad peak hour window, spanning from 6–10 AM and 4–8 PM, totalling eight hours (33% of total daily interval), otherwise, considering only a few hours would have posed a data availability issue. In the 'results_vehicle_level.xlsx' file, which serves as the foundation for our subsequent analysis, there are a total of 342,817 records, with 164,939 of them falling within the peak hour interval, accounting for 48% of the total data. This is why, when analysing the number of available data points, the full-day analysis will not have significantly more data compared to the peak-hour analysis.

6.3 Increase in travel times at arc level

In the previous paragraph 6.1 *Increase in travel times by vehicle and by arc* we calculated travel times of each vehicle on each arc in a highly disaggregated fashion. Here we try to slightly aggregate more the results, by summing up the travel time of all different vehicles on the same arc. We will carry out one analysis only considering peak hours and one considering the whole day.

6.3.1 Increase in travel times at arc level during peak hours

We first filtered out the data corresponding to 'Datetime_a' and 'Datetime_p' (whose meaning has already introduced in 5.2.1 *FCD dataset manipulation and selection of observed days*) between 6-10 AM or 4-8 PM which are considered peak hour intervals. Then, we had to group all the values in the column 'Wasted_time_vehicle' from the file 'results_vehicle_level.xlsx' for the same arc and make the sum (not considering zero or negative values because they don't represented time wasted), creating a new file 'results_arc_level_peak.xlsx'. The latter, contains one row for each arc, whose structure is presented in Figure 46.

Arco	Arc_Length	FFTT (s)	Wasted_time_arc (m)	idcount	daycount
1	101.0081262	4.84944	0.023955387	1	2
2	632.5272075	30.36768	7.835351307	6	4
3	680.6313024	30.6351	7.40071079	13	4
4	48.47823392	2.567117647	0.018465405	1	1
5	34.42743358	1.878545455	0.066521868	1	1
7	33.52791756	1.749913043	0.070861592	4	3

Figure 46: Structure of 'results_arc_level_peak.xlsx'

To account for the varying number of vehicles recorded on each arc, we added a column called 'idcount' to represent the total number of vehicles per arc. We have observations from 164,939 vehicles during peak hours on all the 3900 arcs that we consider in this analysis. Moreover, it is also relevant to check the number of days in which we have data related to each arc, therefore we added a further column 'daycount' to this effect. Then, we calculated the average wasted time by dividing the total wasted time by the number of vehicles for each arc. In Figure 47 we can see the new column 'Average_wasted_time_arc' on the basis of what we just said, that is pointing out the average time wasted in congestion per vehicle, on that specific arc.

Arco	Arc_Length	FFTT (s)	Wasted_time_arc (m)	idcount	daycount	Average_wasted_time_arc (min/veh)
1	101.0081262	4.84944	0.02	1	1	0.02
2	632.5272075	30.36768	7.84	6	4	1.31
3	680.6313024	30.6351	7.39	12	3	0.62
4	48.47823392	2.567117647	0.02	1	1	0.02
5	34.42743358	1.878545455	0.07	1	1	0.07
7	33.52791756	1.749913043	0.07	4	3	0.02
8	31.54073028	1.720909091	0.02	1	1	0.02
9	425.8023904	21.00378082	166.11	17	4	9.77

Figure 47: Structure of 'results_arc_level_peak.xlsx' with new column 'Average_wasted_time_arc'

Moreover, there could be a certain relationship between the wasted time and the length of the arc, so we show the scatter plot with this relation in Figure 48.

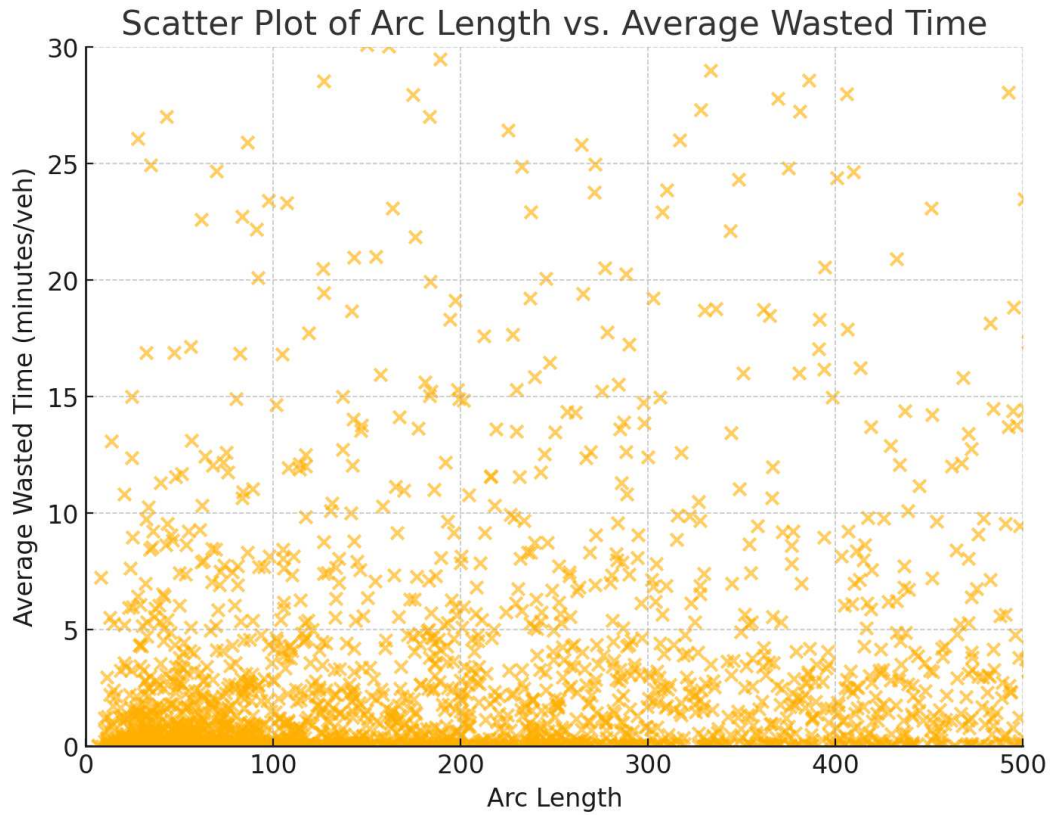


Figure 48: Relationship between arc length and average time wasted on each arc

The points appear widely dispersed with no evident pattern or trend, which aligns with the near-zero correlation coefficient (0.043). This indicates that the arc length does not significantly influence the average time wasted. Factors other than the arc length, such as traffic conditions, road type, or congestion levels, may have a more substantial impact on the wasted time. If a linear relationship was present, we could have divided by the length, but considered the results, we don't need to further process them.

Now we can visualize the results on Qgis to graphically understand what are the arcs where most time is wasted due to congestion. This is shown in Figure 49.

results_arc_level
graph_arc_level_peak
0 - 3 min/veh
3 - 6 min/veh
6 - 9 min/veh
9 - 12 min/veh
12 - 15 min/veh
15 - 30 min/veh
>30 min/veh
Average_wasted_time_arc (min/veh)

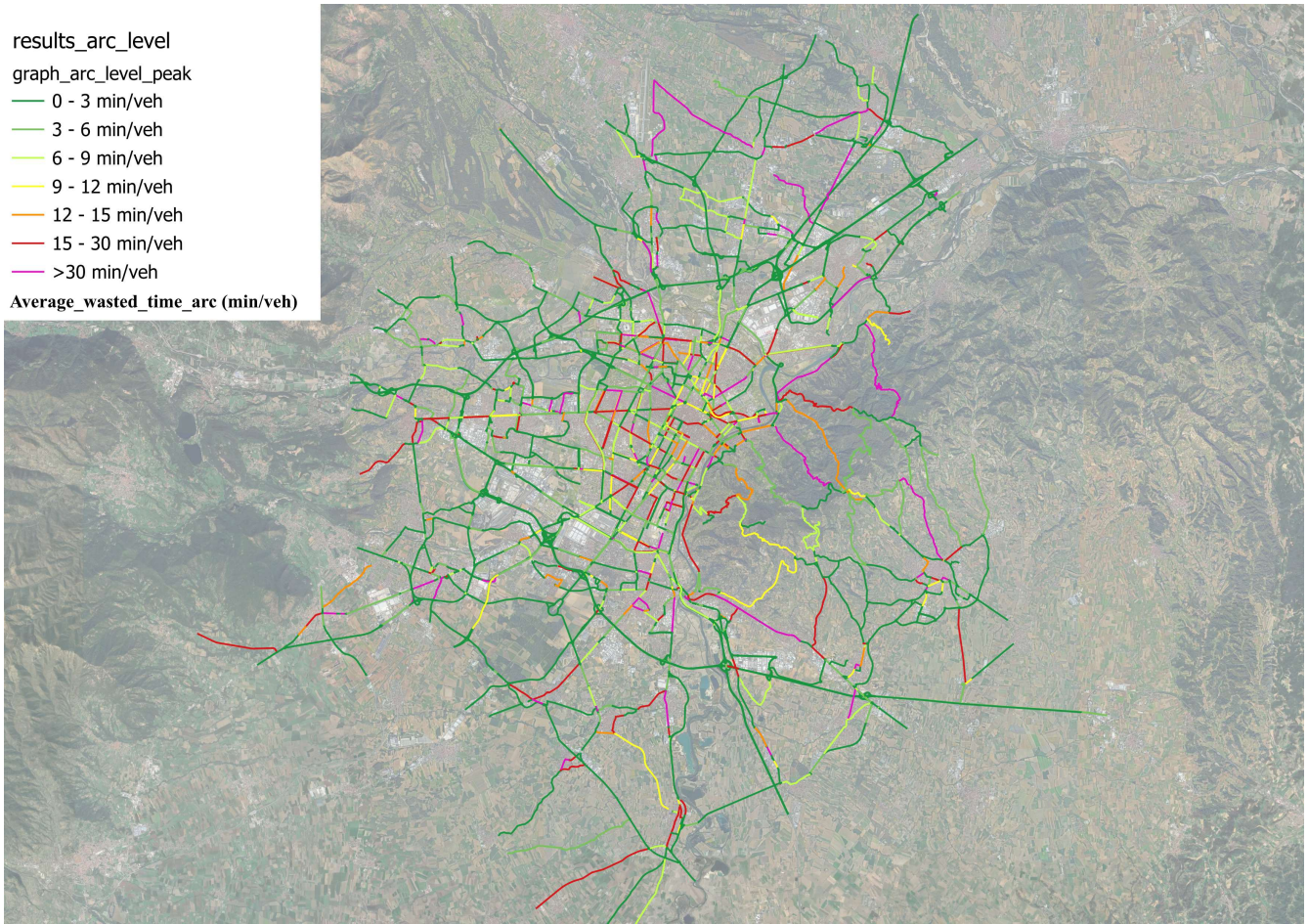


Figure 49: Visualization of arc level congestion, from Qgis layout

In Figure 50 we provide a zoom of congestion in the central urban area of Turin.

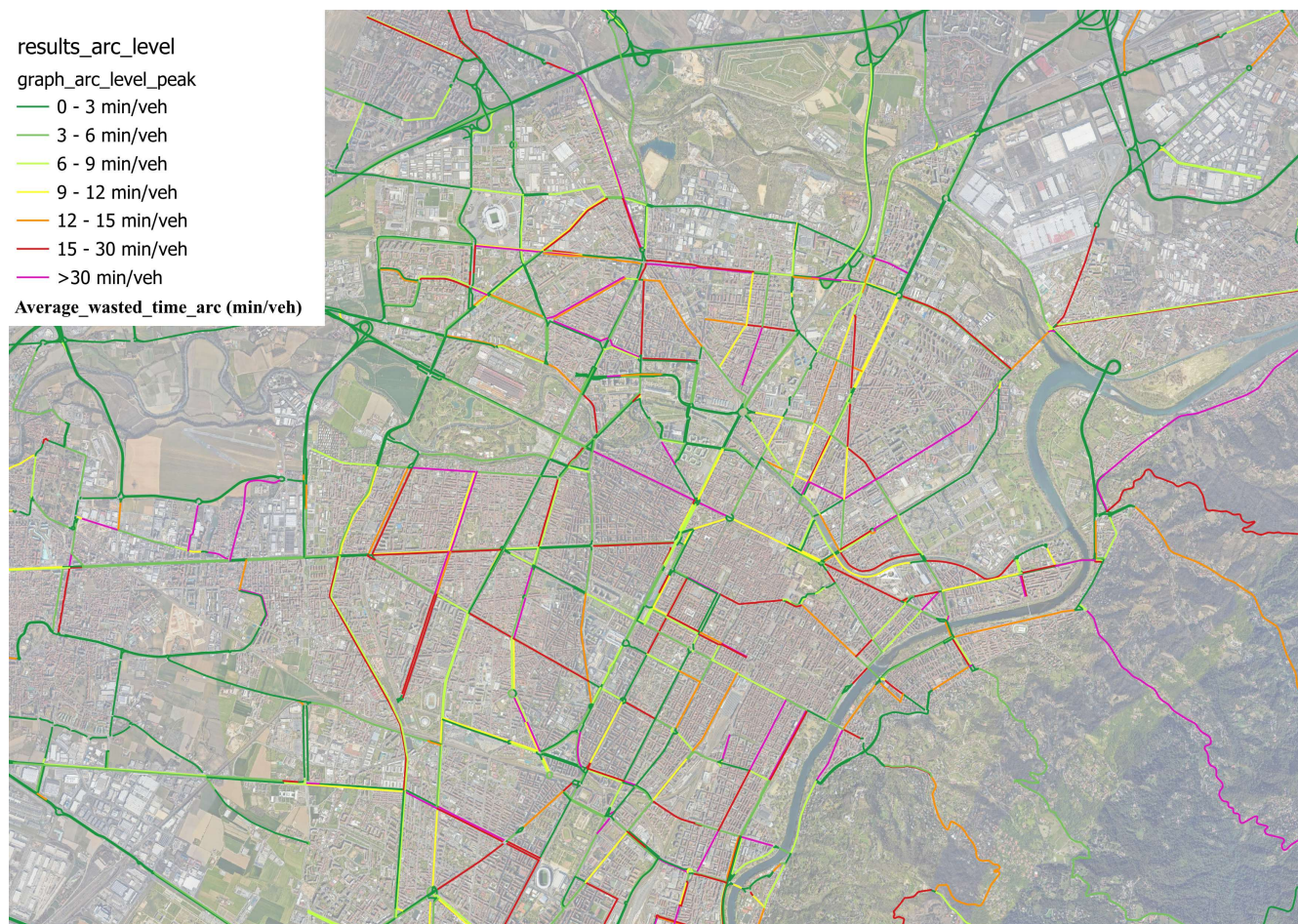


Figure 50: Visualization of arc level congestion in the central area of Turin, from Qgis layout

Moreover, we aim to provide a table showcasing the ten most congested arcs in Turin based on the recently conducted analysis. To achieve this in QGIS, we first selected the arcs within the municipal boundaries of Turin. Next, we sorted them in descending order according to the variable 'Average_wasted_time (min/veh)'. To ensure a reliable estimation, we filtered the data using the 'idcount' column, retaining only values greater than 30. We then selected the top 30 values and exported them as a new shapefile, 'mostcongested_arcs_peak.shp'. Subsequently, we applied the Reverse Geocoding function available in QGIS, which, based on the arcs' positions, provides their corresponding addresses. The selection was limited to the top 30 values due to the computational intensity of the process. Moreover, we considered the average daily traffic for each of these arcs, taken from the file 'OT_DUE.xlsx' created in the thesis work [14]. We are in fact interested in an estimation of the traffic flow during rush hours: therefore, we multiplied the above mentioned average daily traffic in each arc times the fraction of traffic in the peak hour intervals defined in 6.2 *Comparison of data availability during peak hours vs entire-day data*. To achieve this, we consider the hourly distribution of traffic in Turin as reported in [14] (see Figure 51).

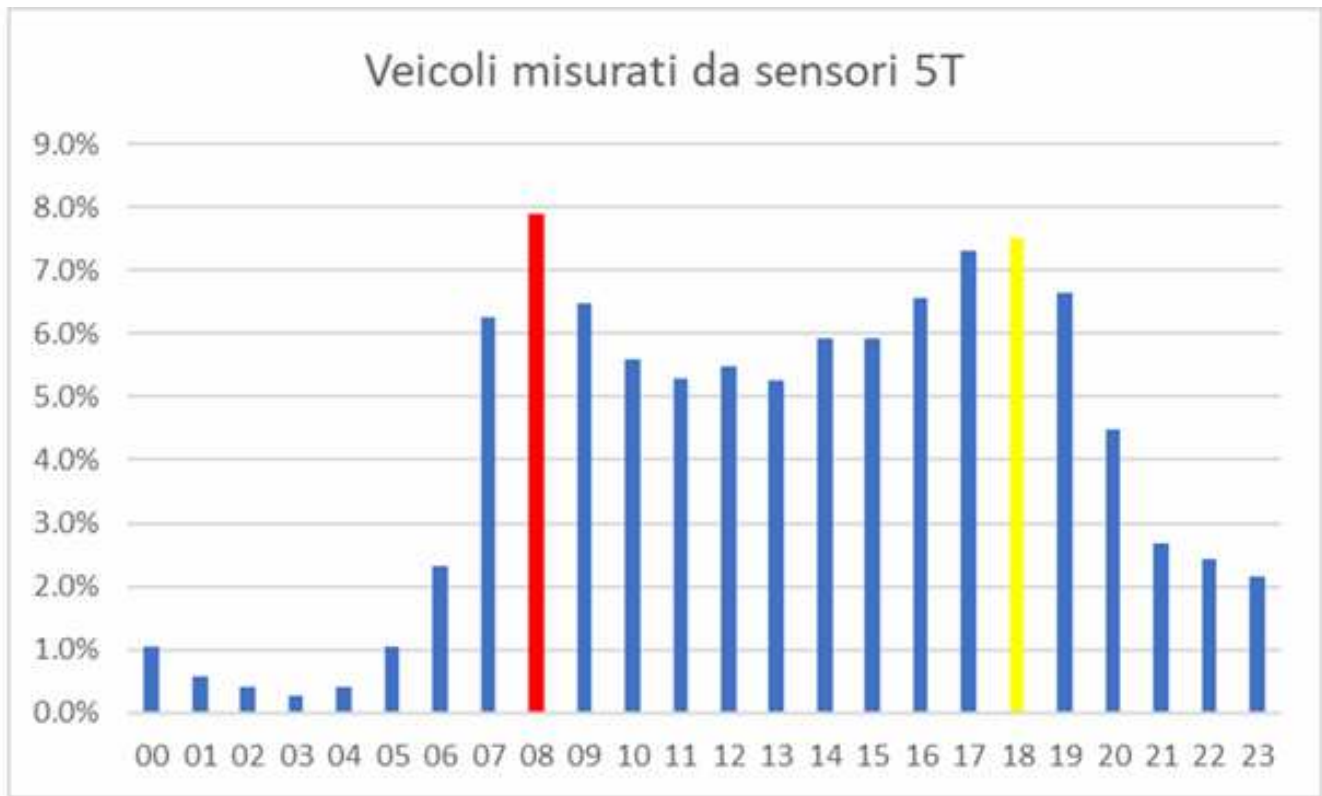


Figure 51: Hourly flow distribution measured on February, 13th, 2019, from [14]

The fraction of traffic in the previously cited interval (6-10 AM and 4-8 PM) is of 50,9%. At the end, by multiplying the 'Average_time_wasted (min/veh)' times the flow during peak hours, we get the total time wasted by all vehicles travelling during rush hours in a typical working day along that specific arc, under the variable named 'Total_time_wasted (min)'. It's important to notice, since the above mentioned average daily traffic doesn't take into account the direction, when an arc is bidirectional, we first divided by two and we also show the direction to which the value it is referred. In Table 9 we can see the ten most congested arcs in Turin according to what we just said.

Table 9: 10 most congested arcs in Turin during peak hours

Arco	L [m]	Name	idcount	Bidir	Direction	Av_w_t (min/veh)	Tot_w_t (h)
6423	407.92	Corso Toscana, Madonna di Campagna	36	YES	-	639	8260
5980	477.89	Corso Racconigi, Cenisia	56	YES	-	411	20547
6081	1691.8	Corso Regina Margherita, Basso San Donato	488	NO	Nord-ovest/Sud-est	288	26066
6830	388.52	Via Cernaia, Quadrilatero Romano	46	YES	-	179	18735
1654	283.84	Corso Umbria, Basso San Donato	37	YES	-	172	6009
6144	604.02	Via Lanzo, Barriera di Lanzo	70	YES	-	135	14381
1929	700.76	Via Chiesa della Salute, Borgo Vittoria	90	YES	-	115	5684
1755	660.31	Corso Moncalieri, Crimea	319	YES	-	107	8980
1372	355.65	Corso Dante Alighieri, Pilonetto	85	YES	-	85	2640
6592	422.28	Via Nino Oxilia, Rebaudengo	125	YES	-	81	5805

In Table 10 we represented 'Average_time_wasted (min/veh)' and 'Total_time_wasted (min)' respectively with 'Av_w_t (min/veh)' and 'Tot_w_t (h)'. An interesting observation, is that the arc where each vehicle wastes the most time is not always the most congested. In fact, we must consider the actual number of vehicles passing through each arc. For example, while Corso Toscana is the arc where each vehicle experiences the highest unit time waste, the arc with the greatest overall time wasted is actually Corso Regina Margherita.

6.3.2 Increase in travel times at arc level during all day

Here, we had to group all the values in the column 'Wasted_time_vehicle' from the file 'results_vehicle_level.xlsx' for the same arc and make the sum (not considering zero or negative values because they don't represent time wasted), creating a new file 'results_arc_level.xlsx'. The latter, contains one row for each arc, whose structure is presented in Figure 52.

Arco	Arc_Length	tt_disaggregate	FFTT (s)	Wasted_time_vehicle	Wasted_time_arc	idcount	daycount
1	101.0081262	4.396643242	4.84944	0	159.1558905	3	2
2	632.5272075	25.63952317	30.36768	0	564.2673854	18	4
3	680.6313024	28.07459307	30.6351	0	523.9382388	19	4
4	48.47823392	3.725232358	2.567117647	1.158114711	2.266039041	2	1
5	34.42743358	5.869857514	1.878545455	3.991312059	3.991312059	1	1
7	33.52791756	2.99993467	1.749913043	1.250021627	4.251695494	4	3

Figure 52: Structure of 'results_arc level.xlsx'

Here as well as before, to account for the varying number of vehicles recorded on each arc, we added a column called 'idcount' to represent the total number of vehicles per arc and the column 'daycount' in which we count the number of days. We have observations from 342,817 vehicles in each of the 4675 arcs that we consider in this analysis. Then, we calculated the average wasted time by dividing the total wasted time by the number of vehicles for each arc. In Figure 53 we can see the column 'Average_wasted_time_arc' on the basis of what we just said that is pointing out the average time wasted in congestion per vehicle per day, on that specific arc.

Arco	Arc_Length	FFTT (s)	Wasted_time_arc (m)	idcount	daycount	Average_wasted_time_arc (min/veh)
1	101.0081262	4.84944	2.65	3	2	0.88
2	632.5272075	30.36768	9.40	18	4	0.52
3	680.6313024	30.6351	8.73	19	4	0.46
4	48.47823392	2.567117647	0.04	2	1	0.02
5	34.42743358	1.878545455	0.07	1	1	0.07
7	33.52791756	1.749913043	0.07	4	3	0.02
8	31.54073028	1.720909091	0.02	1	1	0.02
9	425.8023904	21.00378082	491.39	31	4	15.85

Figure 53: Structure of 'results_arc level.xlsx' with new column 'Average_wasted_time_arc'

Now we can visualize the results on Qgis to graphically understand which are the arcs where most time is wasted due to congestion. This is shown in Figure 54.

results_arc_level

graph_arc_level

0 - 3 min/veh

3 - 6 min/veh

6 - 9 min/veh

9 - 12 min/veh

12 - 15 min/veh

15 - 30 min/veh

>30 min/veh

Average_wasted_time_arc (min/veh)

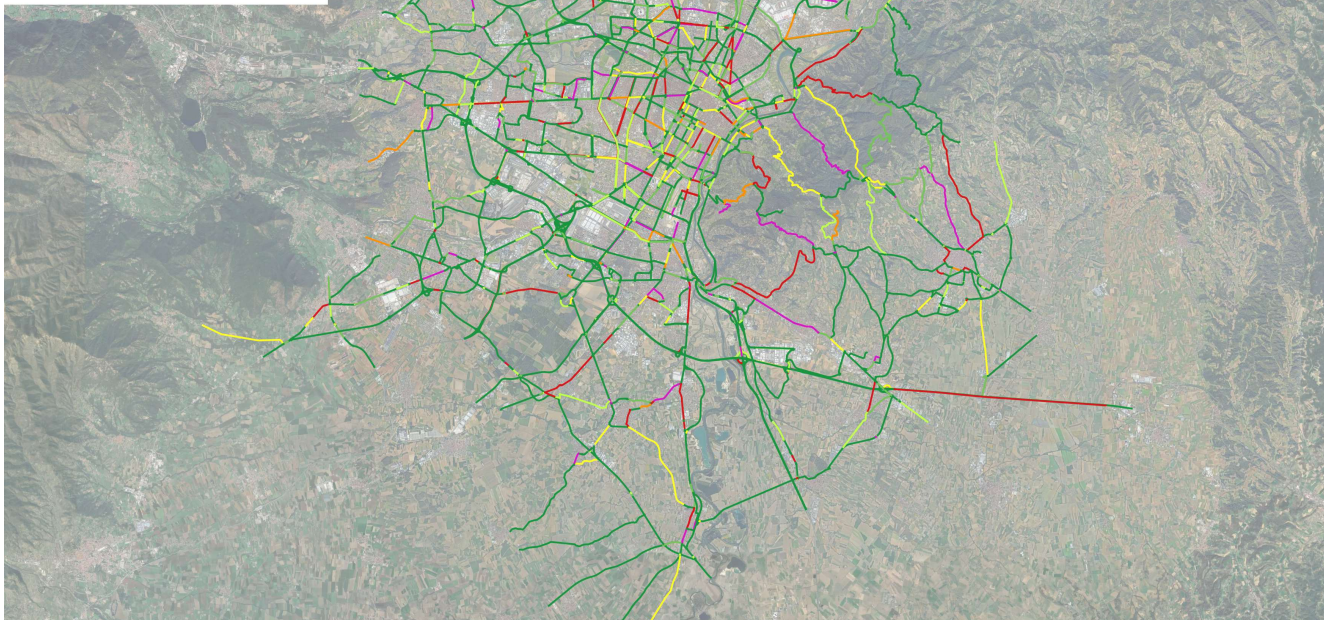


Figure 54: Visualization of arc level congestion, from Qgis layout

In Figure 55 we provide a zoom of congestion in the central urban area of Turin.

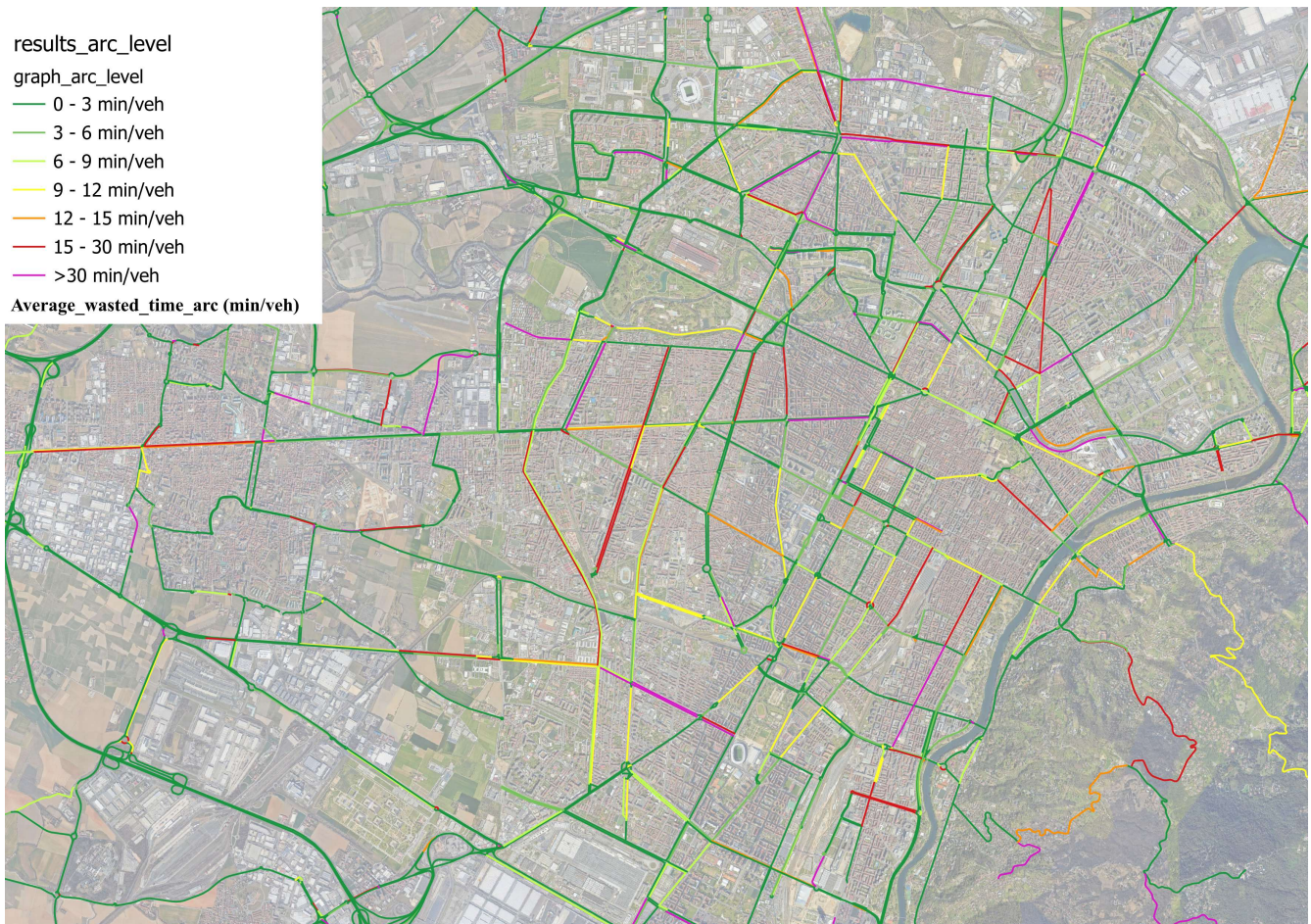


Figure 55: Visualization of arc level congestion in the central area of Turin, from Qgis layout

Moreover, we aim to provide a table showcasing the ten most congested arcs in Turin based on the recently conducted analysis with the same structure as Table 9, with the only difference that we didn't multiplied for any fraction of traffic, because here the whole day is object of analysis and that we filtered 'idcount'>50 because we have a larger time interval in this case.

Table 10: 10 most congested arcs in Turin

Arco	L [m]	Name	idcount	Bidir	Direction	Av_w_t (min/veh)	Tot_w_t (h)
6747	325.22	Corso Potenza, Lucento	60	YES	-	1346	100879
6423	407.92	Corso Toscana, Madonna di Campagna	65	YES	-	359	9109
6438	450.48	Corso Giovanni Agnelli, Circo-scrizione 2	130	YES	-	289	29187
4711	30.33	Strada del Francese, Circo-scrizione 6	116	YES	-	205	33068
4950	72.35	Strada Provinciale della Reggia di Venaria, Circo-scrizione 6	1013	YES	Both	151	38463
7130	405.8	Corso Sebastopoli, Santa Rita	63	YES	-	131	18616
6633	348.18	Corso Giuseppe Gabetti, Borgo Po	71	YES	-	131	10472
4607	266.54	Corso Regina Margherita, Lucento	138	YES	-	123	41382
7063	1652.38	Via Guglielmo Reiss Romoli, Madonna di Campagna	1074	YES	Both	109	35337
6434	1234.01	Corso Grosseto, Madonna di Campagna	195	YES	-	109	21450

We can conclude by emphasizing the distinction between analysing only the average time wasted per vehicle versus considering the total time wasted. The choice of analysis should depend on the specific objective. For instance, if we are interested in the economic impact of congestion on society, the total time wasted is more relevant as it provides a measure of overall social waste. On the other hand, if the focus is on user perception and optimizing traffic management to enhance their experience, the average time wasted per vehicle is a more reliable metric to consider. A small road with low flow but high average time wasted, it's a nightmare for those who travel on it, despite it has a limited impact on the city. A road with low minutes lost per vehicle but a high traffic volume results in a significant collective waste of time and could be a priority for intervention.

Also other kinds of analyses can be carried out to have a clearer picture of the situation, such as analysis on speed, as it will be done in the subsequent paragraph 6.4 *Speed variation at arc level*.

6.4 Speed variation at arc level

As we did previously, we conduct the analysis both during peak hours and across the entire day. We study in which arcs, on average, the real speed is mostly different from the free flow speed. By means of the arc length and the travel time we can calculate both the free flow speed and the real speed.

To calculate the free flow speed, we simply divide the arc's length by the free flow travel time. However, to determine the real speed, we consider the average speed of all vehicles travelling along the same arc.

6.4.1 Speed variation at arc level during peak hours

In more detail, the file 'results_vehicle_level.xlsx' contains a list of the travel times for each vehicle on the arc. For each vehicle, we computed its speed by dividing the arc's length by its travel time. Once we have these individual speeds, we calculate the average of all vehicle speeds on that arc to determine the real speed, and add a new column in the file 'results_arc_level_peak.xlsx' called Average_speed_arc.

This approach ensures that the real speed reflects the collective performance of all vehicles travelling on the arc, rather than being based solely on individual measurements. Then, the difference between the two speeds has been calculated in the column 'delta_speed':

$$\text{delta_speed} = \text{Average_speed_arc} - \text{FFS} \quad (38)$$

And by dividing this last column per the FFS we got the 'Relative difference' expressed in percentage (with values whose range is from -100%, when vehicles are completely stopped, up to 377% if the vehicle travels considerably faster than the FFS of the arc):

$$\text{Relative_difference} = \frac{\text{delta_speed}}{\text{FFS}} \quad (39)$$

The structure is showed in Figure 56:

Arco	Arc_Length	FFTT (s)	FFS (km/h)	Average_speed_arc (km/h)	delta_speed	Relative_difference
1	101.0081262	4.84944	74.98	57.84	-17.14	-23%
2	632.5272075	30.36768	74.98	90.13	15.15	20%
3	680.6313024	30.6351	79.98	88.54	8.55	11%
4	48.47823392	2.567117647	67.98	47.49	-20.50	-30%
5	34.42743358	1.878545455	65.98	21.11	-44.86	-68%
7	33.52791756	1.749913043	68.98	43.98	-24.99	-36%
8	31.54073028	1.720909091	65.98	37.31	-28.68	-43%

Figure 56: 'results_arc_level.xlsx', with calculus of speeds and their difference

On average, the deviation with respect to the free flow speed is of -36%, calculated as the average of the last column showed in Figure 56. Here as well, we can visualize the relative difference on Qgis. (Figure 57)

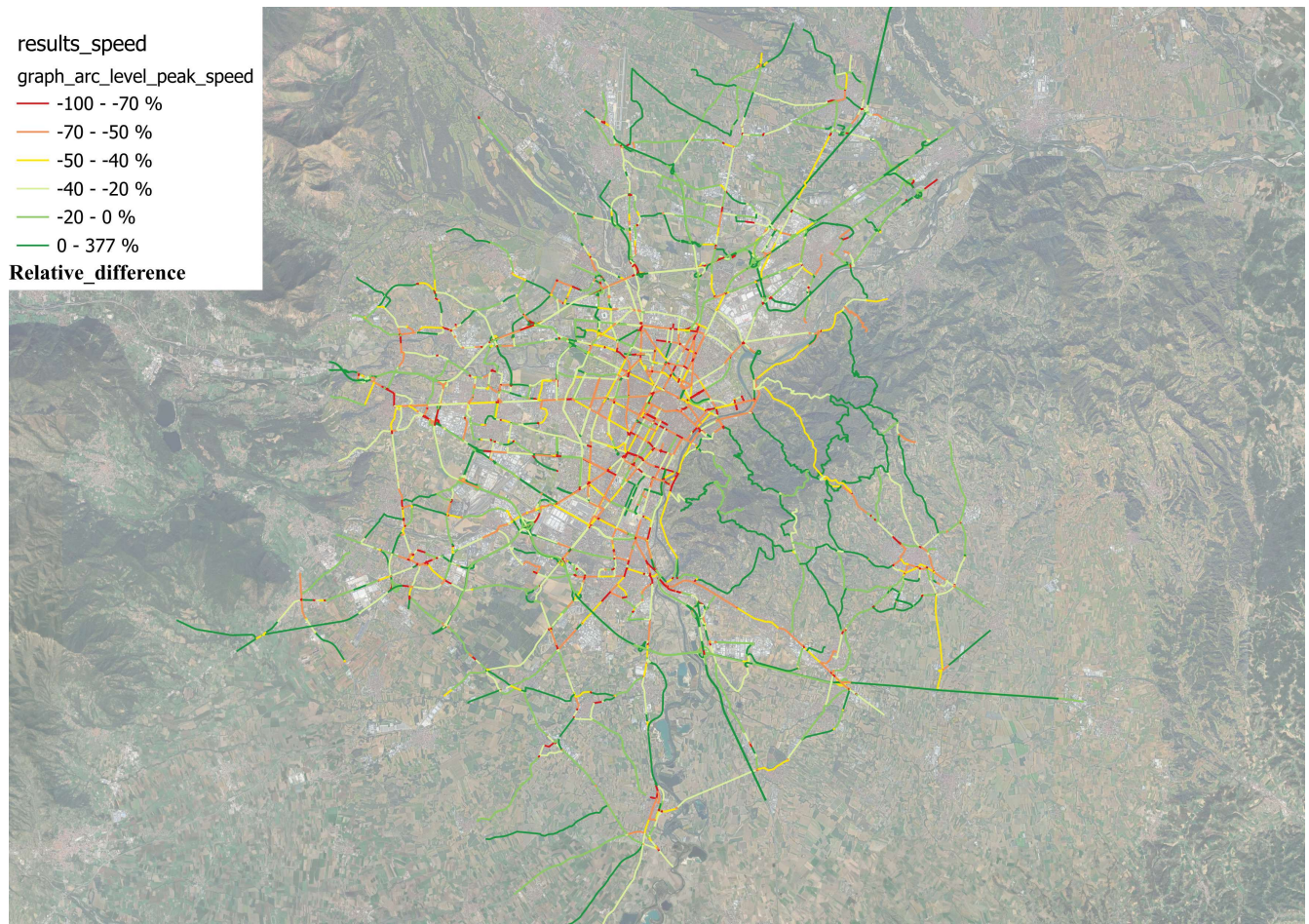


Figure 57: Real speed deviation with respect to free flow speed during peak hours, expressed in percentage

Zooming out in the urban centre of Turin (Figure 58):

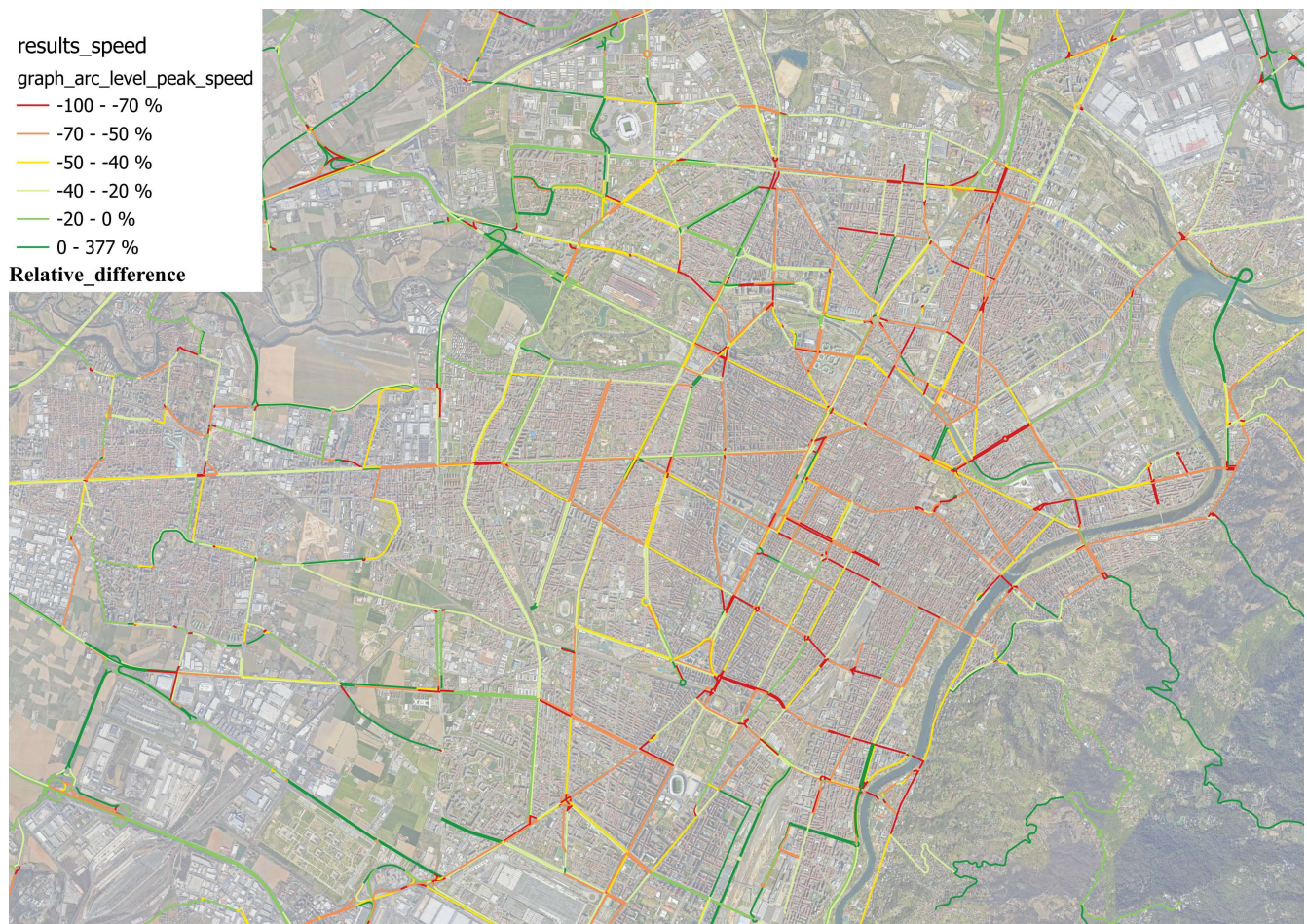


Figure 58: Real speed deviation with respect to free flow speed during peak hours, expressed in percentage, zoom on Turin urban centre

Moreover we provide, in Table 12, a list of 10 arcs where there is the highest difference with respect to the free flow speed, now considering the column 'delta_speed':

Table 11: 10 highest delta_speed arcs in Turin during peak hours

Arco	L [m]	Name	Bidir	Direction	idcount	delta_s	FFS	s_limit
4820	629.17	Tangenziale Nord, Circo- scrizione 6	NO	-	114	-100	125	130
5027	508.24	Tangenziale Nord, Villaretto, Circoscrizione 6	NO	-	59	-85	105	110
6719	670.97	Corso Grosseto, Borgo Vittoria	NO	-	84	-67	100	50
7217	504.64	Corso Giovanni Agnelli, Borgo Cina	NO	-	40	-58	90	50
1989	447.74	Raccordo Au- tostradale Torino-Caselle, Borgo Vittoria	NO	-	46	-47	59	110
3298	409.29	Corso Unità d'Italia, Italia '61	NO	-	152	-46	62	70
6000	361.4	Corso Cairoli, Borgo Nuovo	YES	Both	112	-46	70	50
2517	394.18	Corso Giacomo Matteotti, Cen- tro	NO	-	65	-45	60	50
4467	441.2	Corso Giovanni Agnelli, Borgo Cina	NO	-	79	-45	73	50
6911	793.39	Sottopasso Statuto, San Donato	NO	-	118	-45	77	50

As observed in Table 12, the arcs with the highest difference during peak hours are represented by motorway links. This outcome is expected, as the Turin motorway is typically congested during peak hours.

6.4.2 Speed variation at arc level during all day

On average, the deviation with respect to the free flow speed is of -38%, calculated as the average of the last column showed in Figure 56. Here as well, we can visualize the relative difference on Qgis (Figure 59).

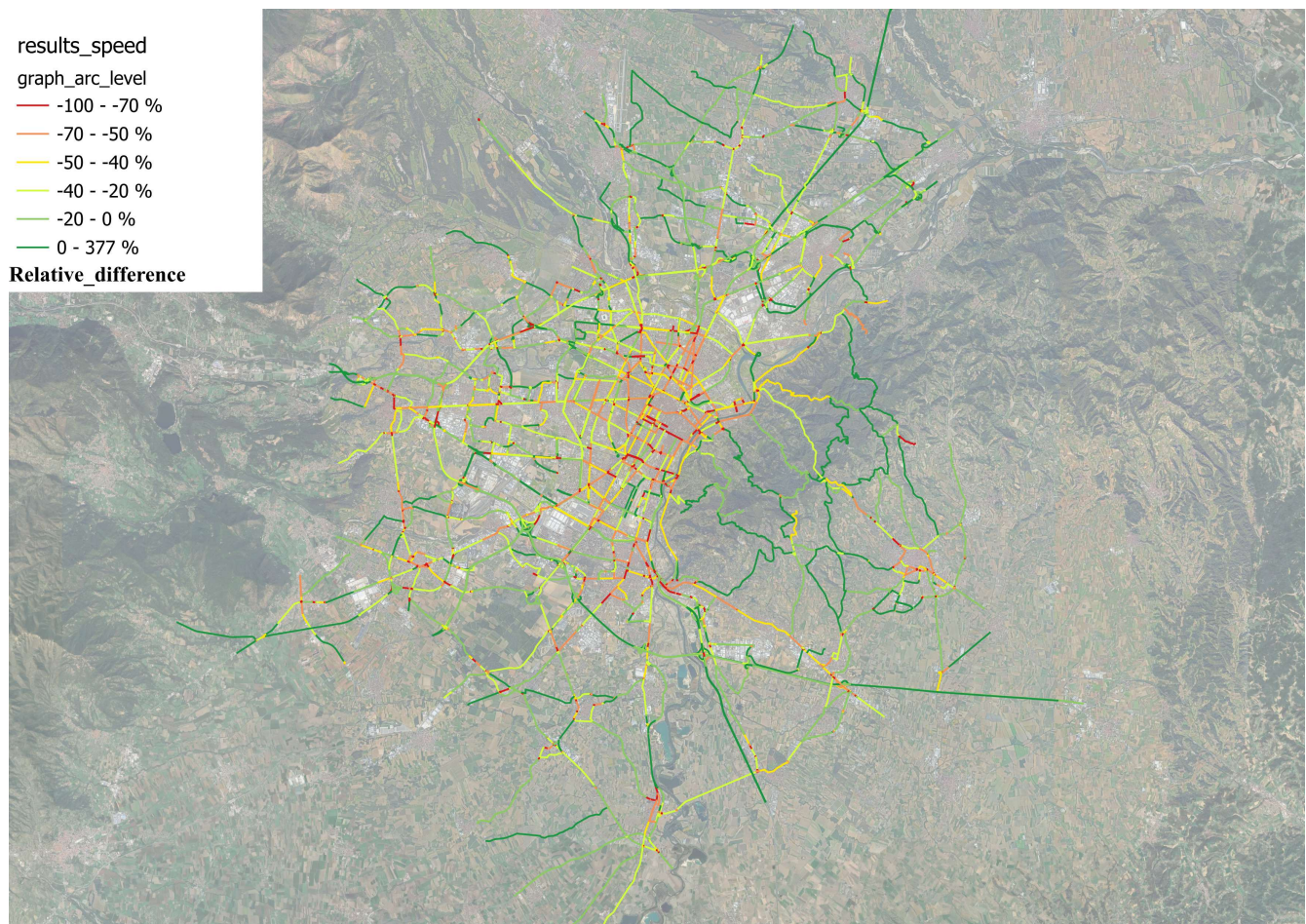


Figure 59: Real speed deviation with respect to free flow speed, expressed in percentage

Zooming out in the urban centre of Turin (Figure 60):

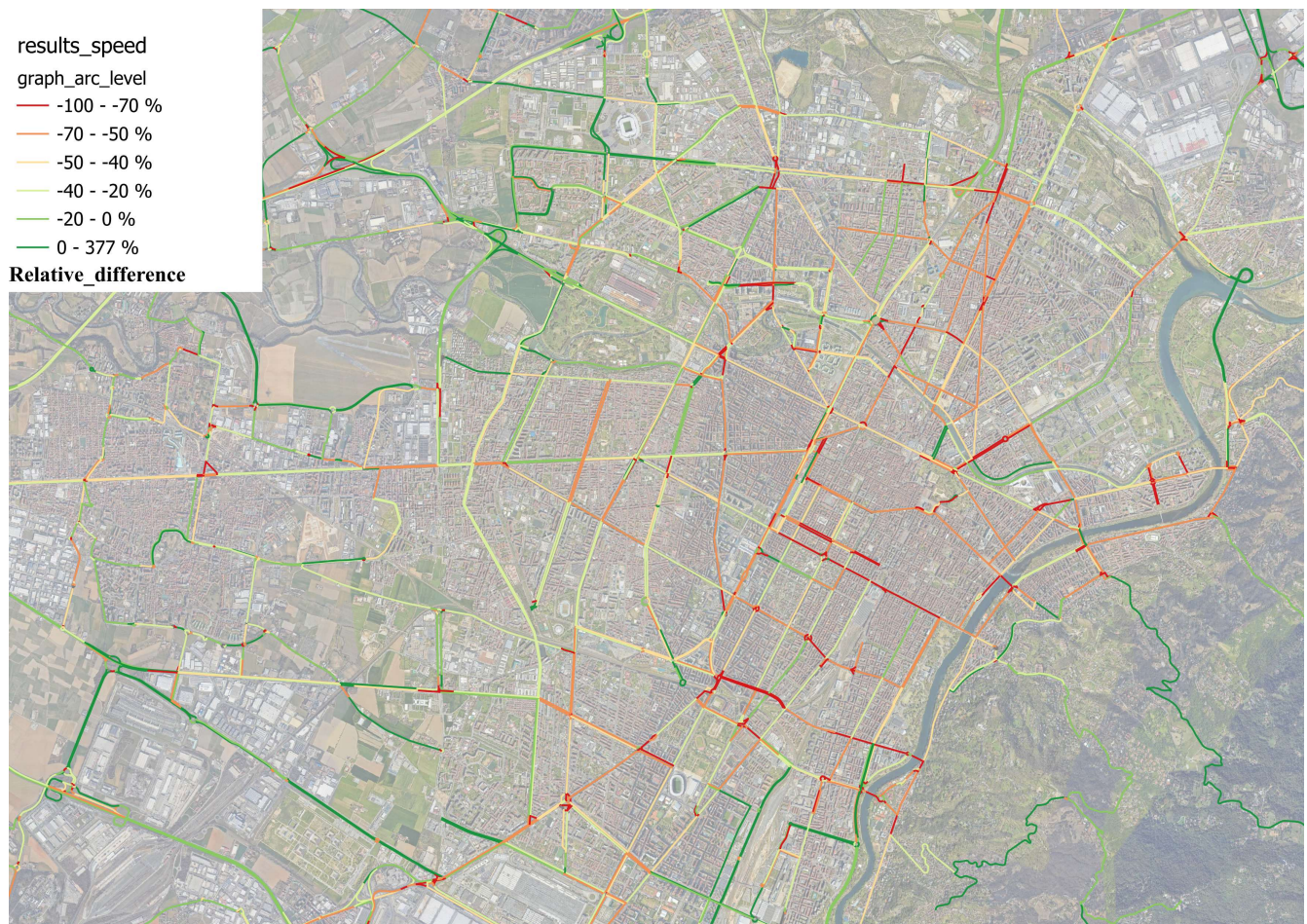


Figure 60: Real speed deviation with respect to free flow speed, expressed in percentage, zoom on Turin urban centre

Moreover we provide, in Table 12, a list of 10 arcs where there is the highest difference with respect to the free flow speed, now considering the column 'delta_speed':

Table 12: 10 highest delta_speed arcs in Turin

Arco	L [m]	Name	Bidir	Direction	idcount	delta_s	FFS	s_limit
6719	670.97	Corso Grosseto, Borgo Vittoria	NO	-	178	-70	100	50
2437	388.99	Strada dell'Aeroporto, Circostrizione 6	NO	-	65	-67	85	50
7217	504.64	Corso Giovanni Agnelli, Borgo Cina	NO	-	91	-60	90	50
931	61.1	Strada Comunale del Portone, Mirafiori Nord	NO	-	81	-58	60	50
1655	40.48	Lungo Stura Lazio, Pietra Alta	NO	-	87	-55	66	50
4564	49.81	Piazzale Costantino Il Grande, Santa Rita	NO	-	273	-54	55	50
4623	39.6	Corso Belgio, Vanchiglietta, Circostrizione 7	NO	-	334	-49	50	50
2116	19.34	Corso Alessandro Tassoni, Martinetto	NO	-	154	-49	52	50
2518	53.16	Corso Germano Sommeiller, San Salvario	NO	-	82	-48	55	50
1521	54	Strada Val San Martino, Madonna del Pilone	NO	-	190	-46	54	50

6.4.3 Case study: Turin streets where the average speed is larger than 30 km/h

Furthermore, speed is often a topic of intense debate, particularly concerning travel safety and efficiency. In this context, the Italian city of Bologna, located in northwestern Italy, has introduced a new initiative called "Città 30" (City 30) in January 16th, 2024 in the context of the approval of the detailed urban traffic plan (PPTU, in italian Piano Particolareggiato del Traffico Urbano) [7]. While the standard urban speed limit in Italy is 50 km/h, Bologna has become the first major Italian city to implement a comprehensive 30 km/h speed limit across its entire municipal area

in a systematic and widespread manner. This move aims to enhance road safety, improve urban living conditions, and promote more sustainable mobility.

Opponents of the 30 km/h speed limit often present several arguments¹⁹, primarily focusing on the perceived drawbacks of such policies in urban settings. These criticisms include concerns about practicality, efficiency, economic impact, and individual freedoms. One of the most common arguments is that lower speed limits lead to increased travel times, particularly for commuters who rely on private vehicles. Critics argue that forcing cars to move at 30 km/h instead of 50 km/h could slow down traffic, making daily commutes longer and less efficient, especially in larger cities where congestion is already an issue. They contend that this could have a ripple effect on productivity, as more time spent in traffic means less time available for work, family, or leisure activities.

They also state that there has been an increase of travel time even with the public transport system. Moreover, they sustain that travelling slower, using low gears, can increase environmental pollution. The municipality of Bologna recently has published data which support the effectiveness of this measure²⁰, concerning road accident data recorded by the Local Police on roads within the municipal territory of Bologna (excluding highways and the ring road) from January 15th, 2024, to January 12th, 2025 (a 52-week period), compared to the average for the corresponding periods of the two preceding years (January 17th, 2022 - January 15th, 2023, and January 16th, 2023 - January 14th, 2024), showing the following trends in particular:

- -13.10% total accidents;
- -48.72% fatalities;
- -11.08% injured persons;
- -9.78% accidents with injuries;
- -20.71% accidents without injuries;
- +36.00% persons in critical condition;

The environmental context is marked by a significant reduction in NO₂ (nitrogen dioxide) levels recorded in 2024 at the ARPAE monitoring station in Porta San Felice. The average hourly value of 29 µg/m³ recorded in 2024 (as of November 30th, 2024, the latest available data) represents a 29.3% decrease compared to the annual average for 2022–2023 (41 µg/m³). In absolute terms, this is the lowest level recorded in the past 10 years.

¹⁹<https://www.ilrestodelcarlino.it/bologna/cronaca/bologna-citta-30-ultime-notizie-sw2lqzoa>

²⁰<https://www.comune.bologna.it/informazioni/citta-30-dati-6-mesi>

Basing on what we just said, it could be interesting to visualize in Turin, what are the arcs where the mean speed on the arcs is greater than 30 km/h during peak hours (Figure 61). We used the column 'Average_speed_arc' of the file 'results_arc_level_peak.xlsx' introduced in 6.3.2 *Increase in travel times at arc level during all day*.

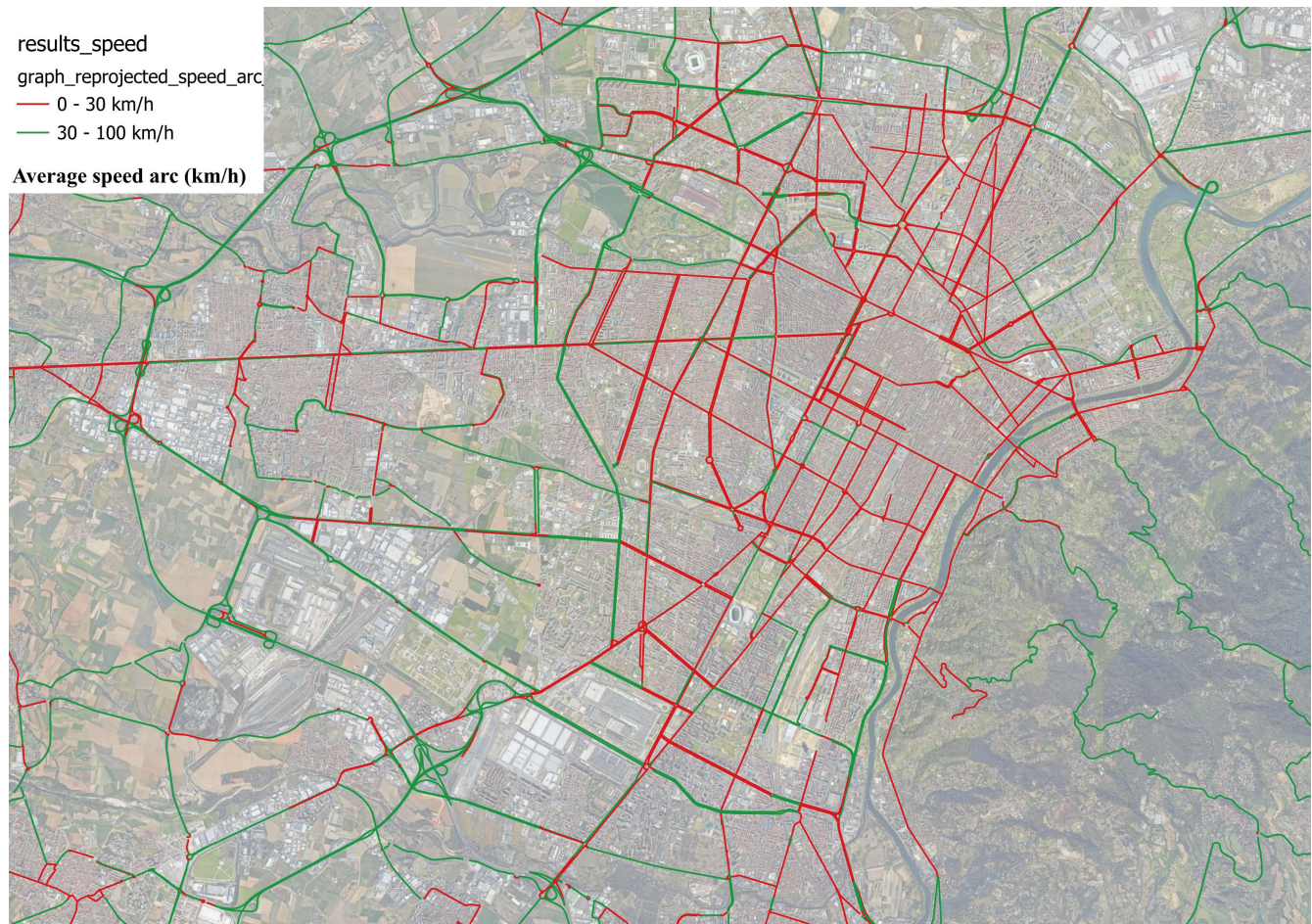


Figure 61: Average speed on each arc, zoom on Turin

As observed, the arcs where the average speed is below 30 km/h are primarily concentrated in Turin's urban centre, with a few exceptions. This is not a result of any political decisions by the municipality but rather a consequence of traffic congestion.

6.5 Increase of arc travel times at zonal level

To better reflect the zoning within Turin, from Appendix 9.5 of [1], as illustrated in Figure 62, and the administrative boundaries of the municipalities²¹ surrounding Turin shown in Figure 8, we created a new shapefile, "zonetoealtricomuni.shp" presented in Figure 63. This shapefile defines zones used to aggregate the travel times analysed in 6.3 *Increase in travel times at arc level*.

²¹<https://geoportale.igr.piemonte.it/cms/>

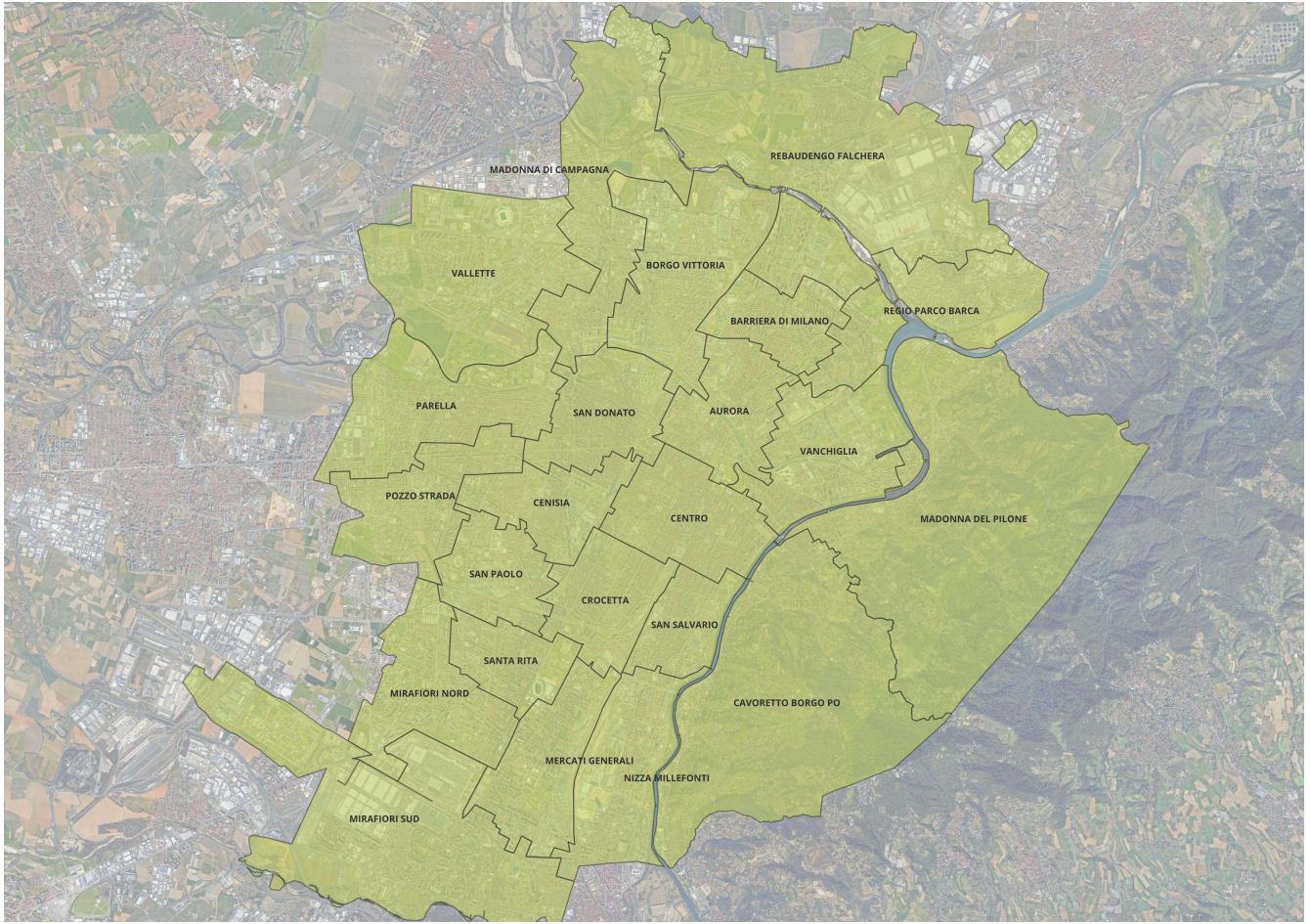


Figure 62: Zoning inside Turin

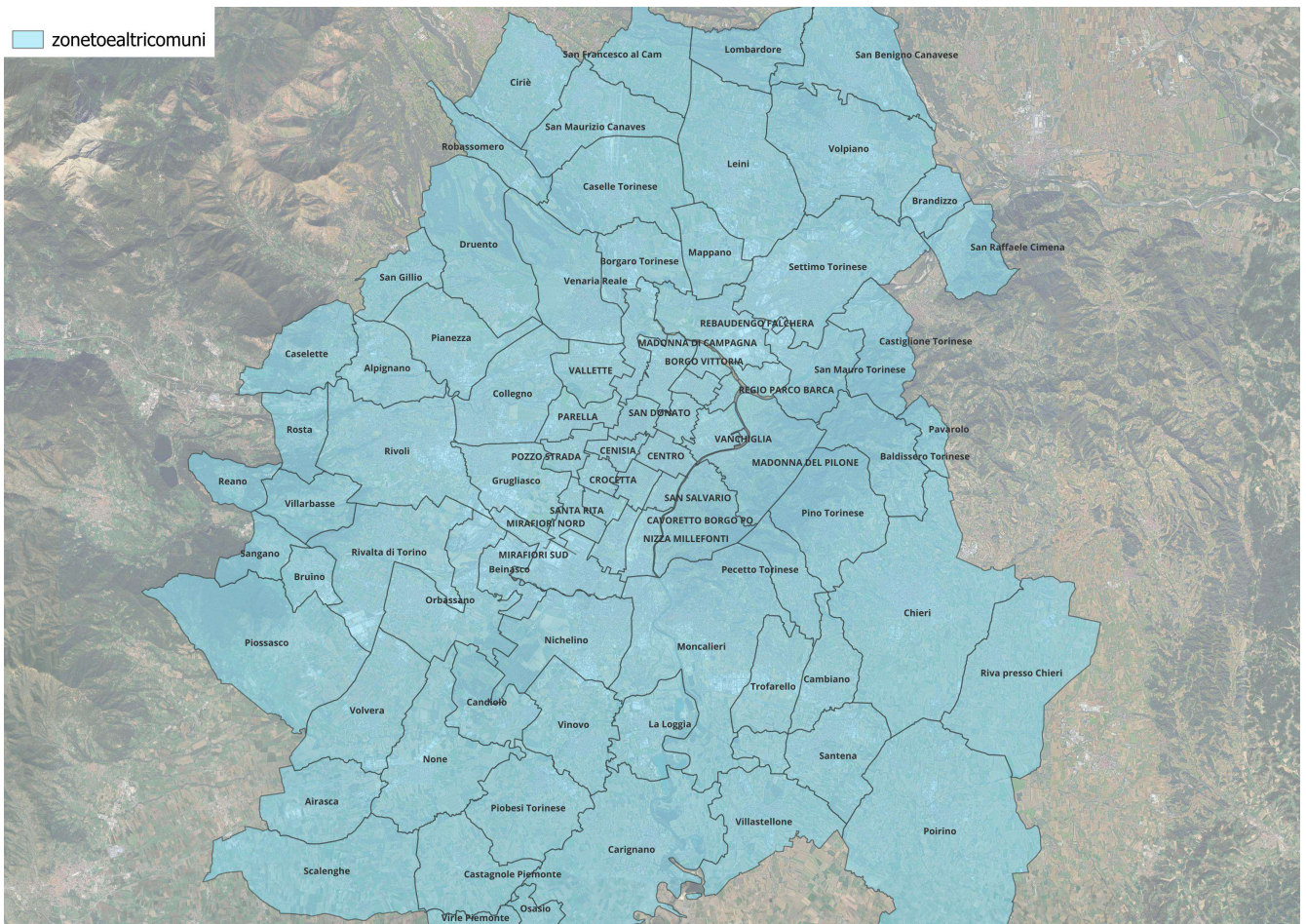


Figure 63: Zoning of the study area

This process resulted in a total of 84 zones. Using QGIS, we assigned each arc to its corresponding zone and calculated the sum of the 'Average_time_wasted_arc' column within each zone to determine the wasted time at the zonal level. This has been done both considering peak hours and the entire day.

6.5.1 Increase of arc travel times at zonal level during peak hours

In this case, we aggregated at zonal level the 'Average_time_wasted_arc' from the file 'results_arc_level_peak.xls'. In reality, the time wasted could be strongly influenced by the extension of each zone, therefore by the total length of arcs falling into each zone, so, we decided to 'normalize' the time wasted by dividing it per the total length of arcs inside each zone, getting a new variable 'unit_time_wasted_zone' expressed in minutes/veh/kilometres, thus the amount of time lost by each vehicle for each km of road. We can visualize the new result in Figure 64:

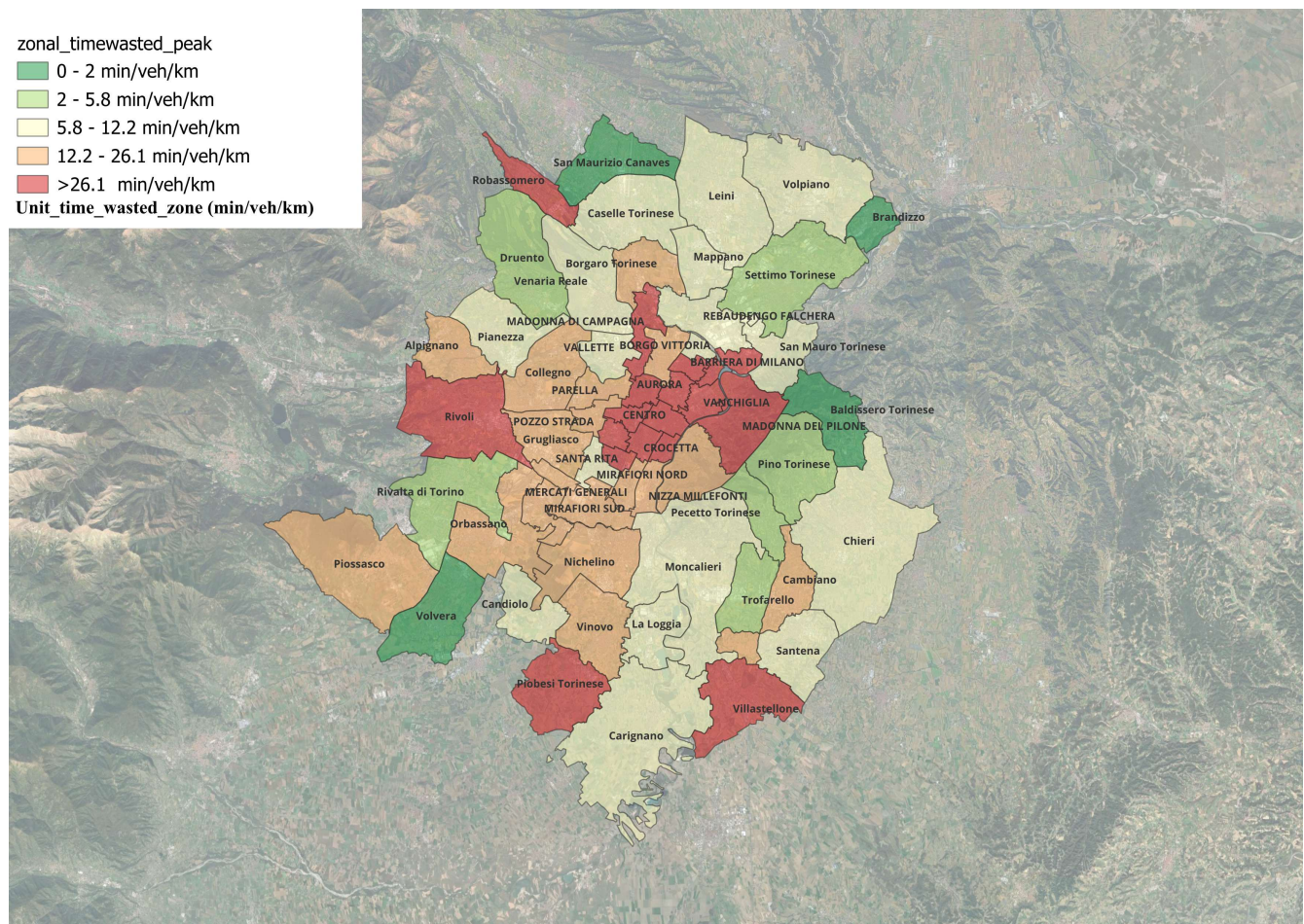


Figure 64: Unit average time wasted for each zone during peak hours

We see that the zones in which most of the time is wasted due to congestion are those located inside the central urban area of Turin (CENISIA,CENTRO, MADONNA DI CAMPAGNA, SAN SALVARIO, VANCHIGLIA, REGIO PARCO BARCA, SAN PAOLO, CROCETTA, SANTA RITA, BARRIERA DI MILANO, AURORA, MADONNA DEL PILONE, SAN DONATO) but also some located outside Turin like Rivoli,Robassomero,Villastellone, Piobesi Torinese.

6.5.2 Increase of arc travel times at zonal level during entire day

In this case, we aggregated at zonal level the 'Average_time_wasted_arc' from the file results_arc_level.xlsx. (see Figure 65) We directly expressed the result in the column 'unit_time_wasted_zone' in min/veh/km of road.

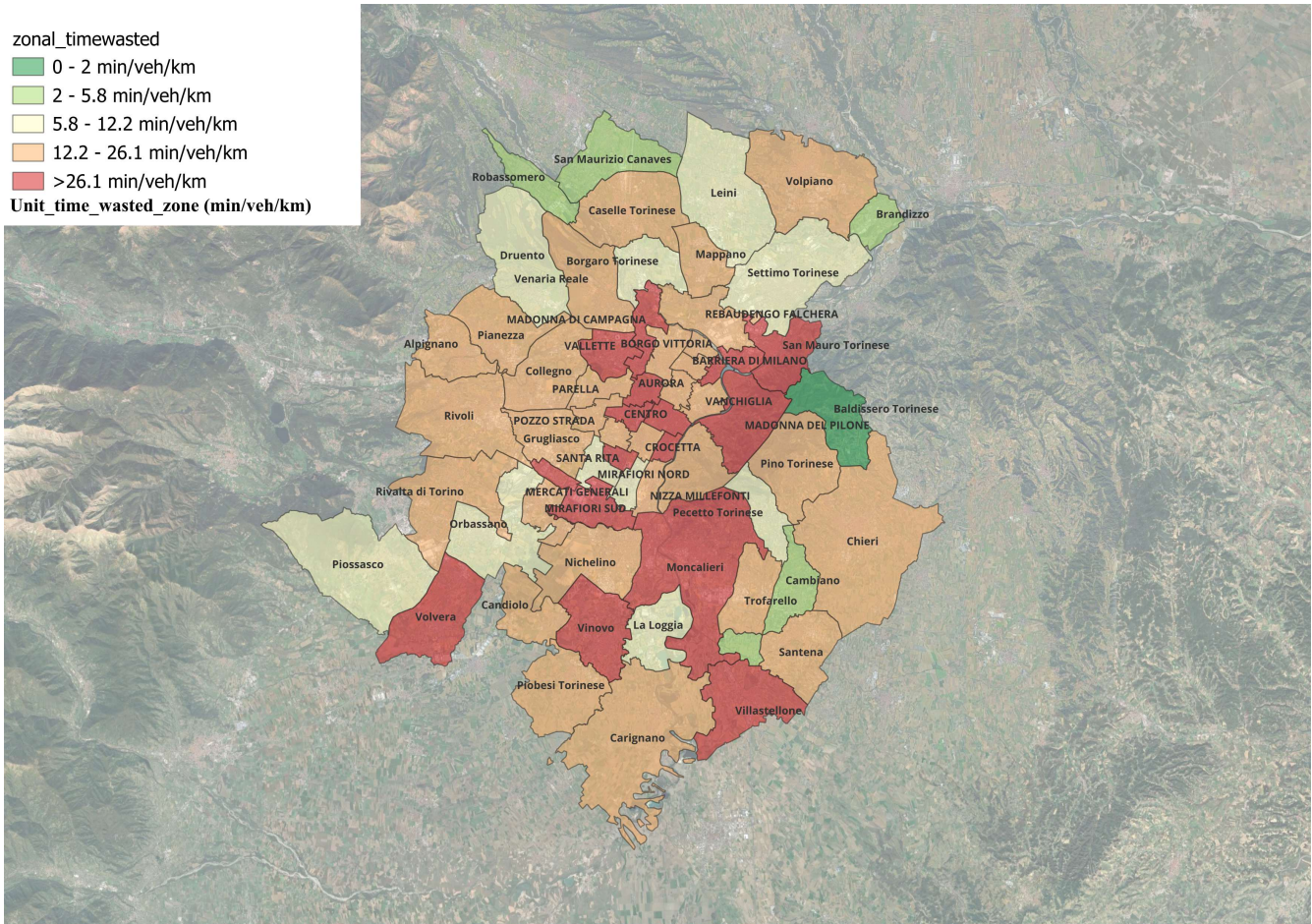


Figure 65: Unit average time wasted for each zone considering all day

As shown in Figure 65, the results align with previous findings, confirming that the most congested areas are concentrated in the urban centre of Turin. However, congestion appears to be slightly lower in the very centre and more pronounced in the surrounding areas. This may be due to the fact that, during peak hours, a larger number of people travel toward the city centre for work or study, whereas throughout the rest of the day, traffic conditions tend to be more balanced.

6.6 Increase of vehicular travel times at zonal level

As well as we did in 6.5 *Increase of arc travel times at zonal level* here we aggregate previous results at the zonal level, but focusing on the wasted time by each vehicle inside each zone irrespective of the arc that was travelled. Like before, we performed the analysis both for peak hours and the entire day. Therefore, we summed up the above mentioned wasted times and divided by the number of vehicles recorded inside each zone, getting a new variable 'wasted_time_zonal_disaggregated' expressed in minutes per vehicle.

6.6.1 Increase of vehicular travel times at zonal level during peak hours

We can visualize the results in Figure 66:

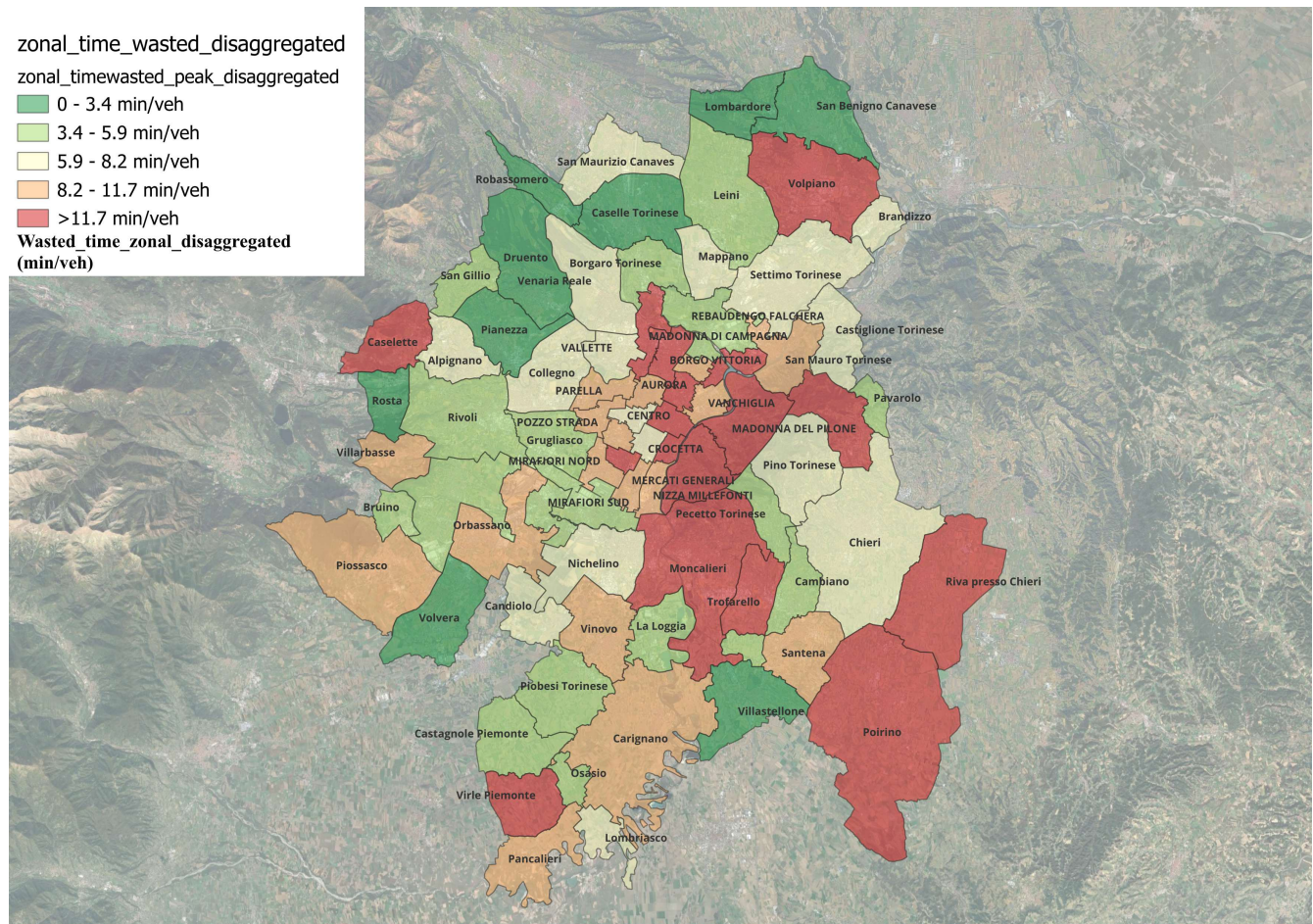


Figure 66: Unit average time wasted for each zone during peak hours, expressed in minute per vehicle

The results are coherent with the previous analysis carried out in the previous section, where the worst zones are located mostly inside the central urban area of Turin, with the addition of some external zones that are larger. One might consider normalising the time wasted inside each zone based on the area within each zone, however such results would not have a very intuitive meaning. It could also be interesting to consider the total time wasted inside each zone, based on considerations done at the end of 6.3.2 *Increase in travel times at arc level during all day*. In Figure 67 the reader can appreciate the result.

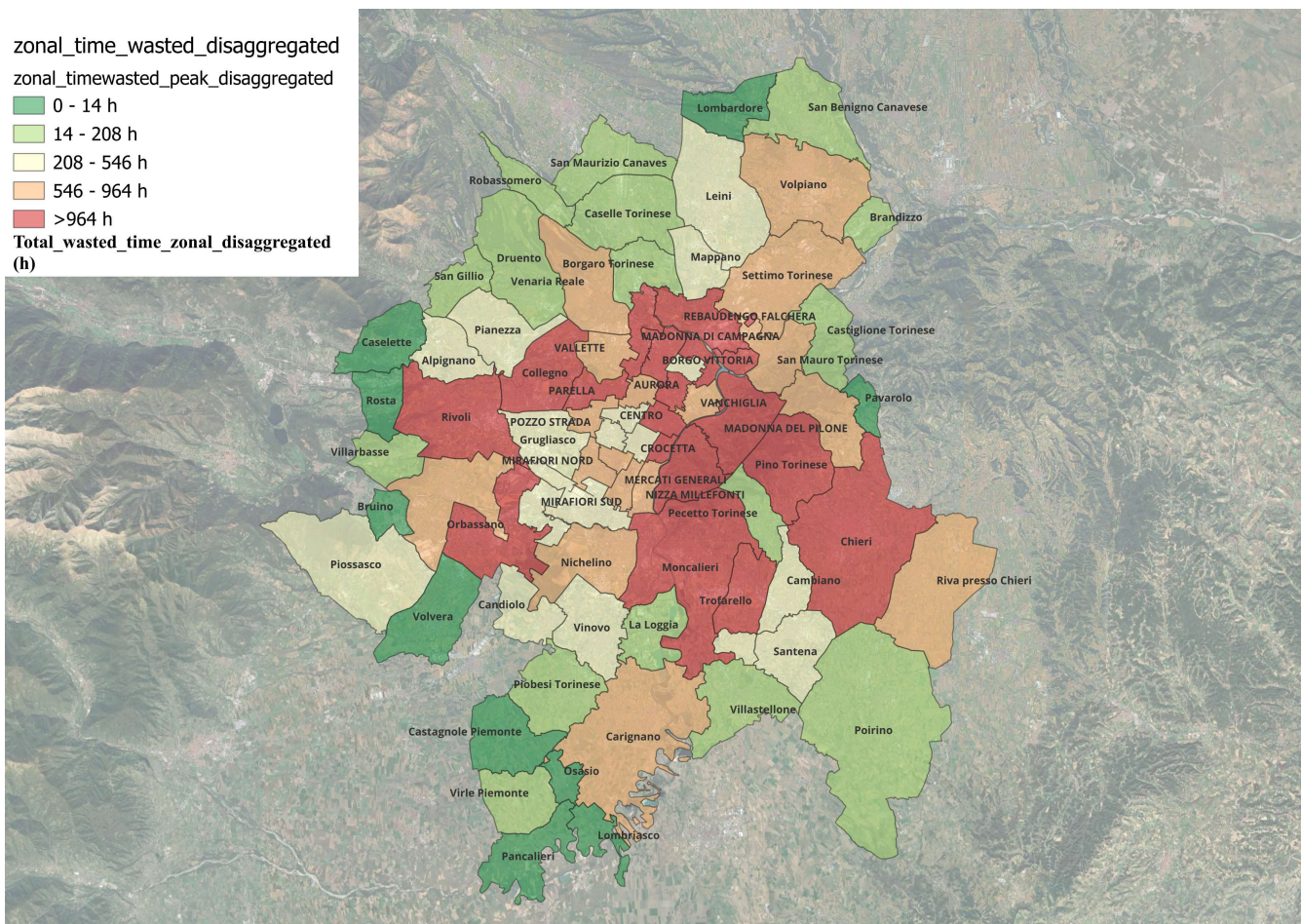


Figure 67: Total time wasted for each zone during peak hours, expressed in hours

It is important to mention that we multiplied by the number of monitored vehicles. However, we do not know the exact number of vehicles present in each area, so the result in this case should be considered as an indicator.

6.6.2 Increase of vehicular travel times at zonal level during the whole day

We can visualize the results in Figure 68.

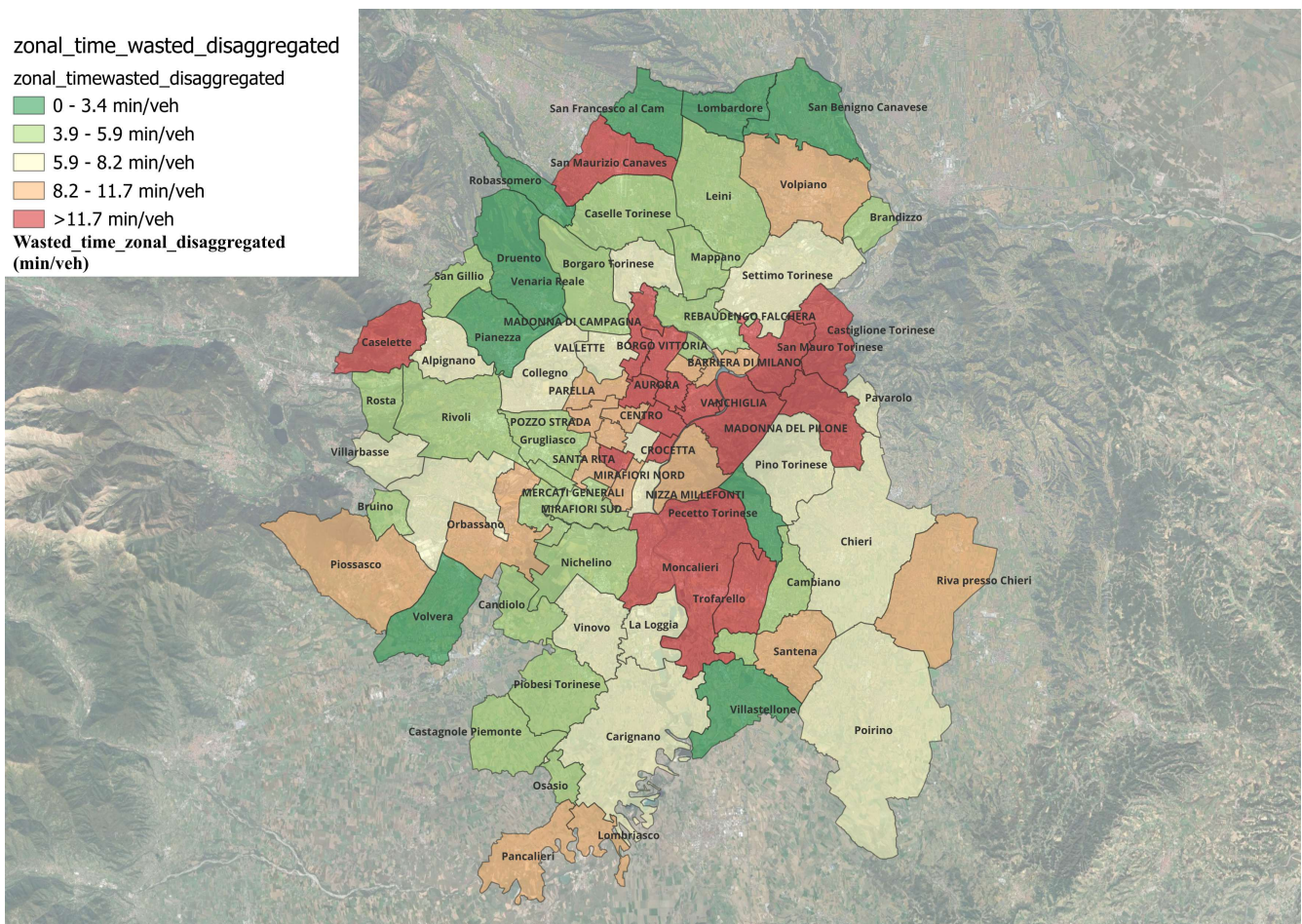


Figure 68: Unit average time wasted for each zone, expressed in minute per vehicle per day

Instead, if we focus on total time wasted for each zone, not depending on number of vehicles (Figure 69).

zonal_time_wasted_disaggregated
zonal_timewasted_disaggregated
0 - 14 h
14 - 208 h
208 - 546 h
546 - 964 h
>964 h
Total_wasted_time_zonal_disaggregated
(h)

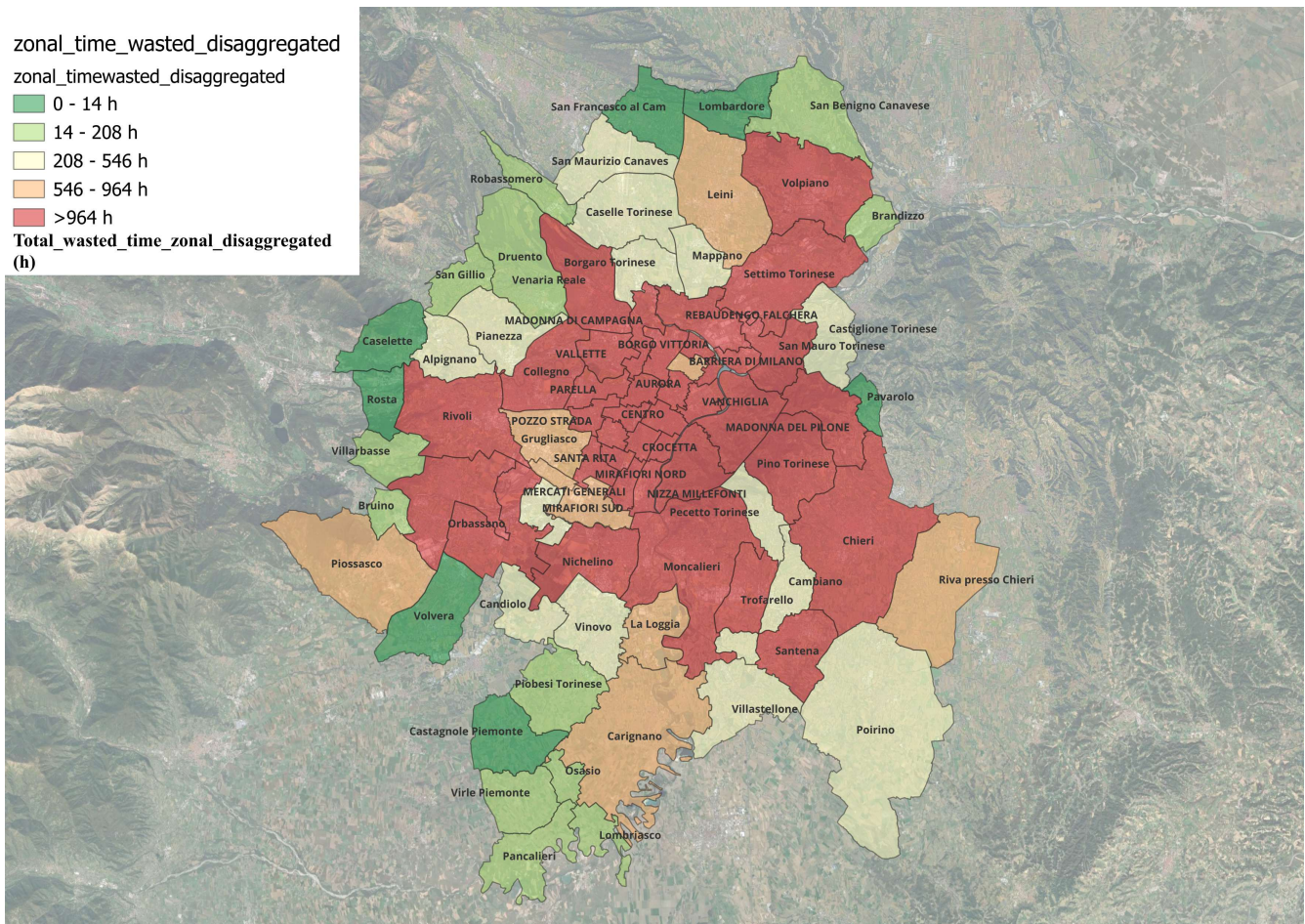


Figure 69: Total time wasted for each zone during all day, expressed in hours

To better compare the results, since it's difficult to interpret them only looking at Figures 66 and 68, we created an Excel file 'confronto_zonaldisaggregated.xlsx' where we had initially three columns: 'ZONE', 'wasted_time_zonal_disaggregated (min/veh)' and 'wasted_time_zonal_disaggregated_peak(min/veh)'. The structure is presented in Figure 70.

ZONE	wasted_time_zonal_disaggregated (min/veh)	wasted_time_zonal_disaggregated_peak (min/veh)
Alpignano	6.684463698	6.406101145
AURORA	13.99677092	12.35811027
Baldissero Torinese	24.09871119	20.71056643
BARRIERA DI MILANO	9.070988315	9.051367969
Beinasco	4.408614569	5.754462522
Borgaro Torinese	6.725194222	4.624498286
BORGHO VITTORIA	16.4214364	22.18777305
Brandizzo	4.764655606	8.051916273
Bruino	5.650409303	4.118928188
Cambiano	4.652093223	4.090679733
Candiolo	5.010799803	5.923964441
Carignano	7.795538543	10.03348035
Caselette	16.68003378	33.03053579

Figure 70: Structure of 'confronto_zonaldisaggregated.xlsx'

We calculated the ratio between 'wasted_time_zonal_disaggregated_peak' and 'wasted_time_zonal_disaggregated' to understand to which extent traffic conditions worsen during peak-hour traffic. A high ratio in-

dicates that most traffic congestion is concentrated during peak hours, while a ratio close to one suggests that traffic remains consistent throughout the day. Our analysis revealed that in the central urban area of Turin, the ratio is, on average, close to one. This confirms the expected scenario where traffic congestion is spread throughout the day, highlighting the need to manage mobility demand, particularly during peak hours. Conversely, suburban areas exhibit the highest ratios, indicating that traffic is primarily concentrated in peak hours. This pattern is likely driven by commuter movements (pendolarism). For instance, Virle Piemonte has the highest ratio (29.3), meaning that virtually all wasted time due to congestion occurs during peak hours. Vinovo, Trofarello, and San Salvario show ratios between 2 and 3, indicating that congestion in these areas doubles during peak hours. This suggests that these zones are heavily influenced by commuter flows or specific traffic patterns during peak times. On the other hand, zones located inside centre urban area of Turin, represent a ratio next to one, meaning the problem of traffic is equally distributed along the day, so need to be taken important long term decisions. We provide in Figure 71 the graphic visualization of those comments.

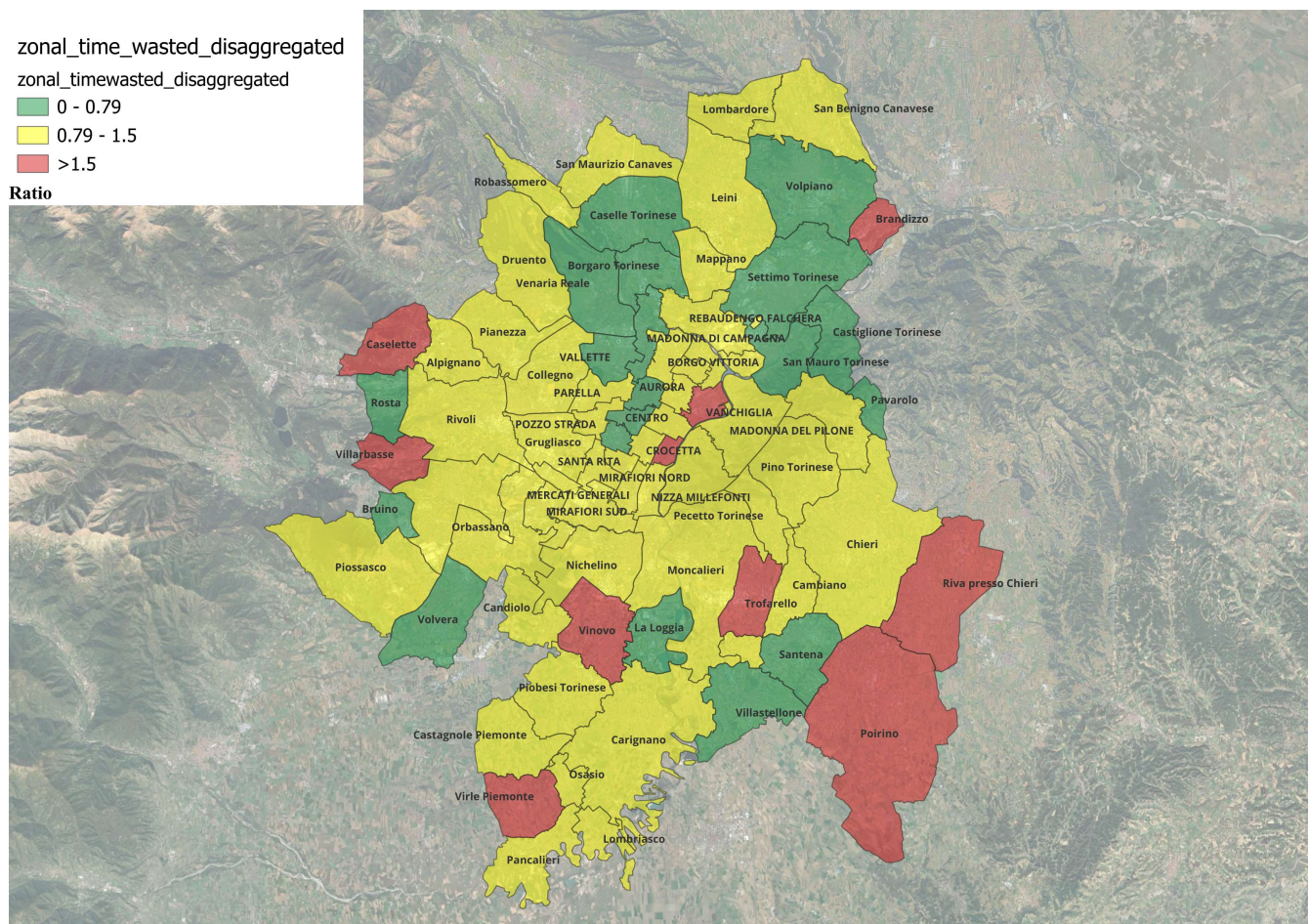


Figure 71: Visualization of ratio between 'wasted_time_zonal_disaggregated_peak' and 'wasted_time_zonal_disaggregated'

6.7 Speed measure for O/D relations between the zones introduced in 6.5 Increase of arc travel times at zonal level

Accessibility of an area is the easiness of reaching interest destinations.

Over the years, different models have been developed to give a measure of accessibility, in order to represent the interaction between the transport system and land use.

In Figure 72, we present a table with a commonly accepted categorization of the possible alternatives [4].

		Behavioral	Not-behavioral
Attractiveness/ Cost-based	Disaggregate	Activity-based random utility models	-
	Aggregate	Trip-based random utility models	Gravity models
Opportunity-based	Disaggregate	Activity-based perceived opportunities models	-
	Aggregate	Zonal perceived opportunities models	Isochrones-based models

Figure 72: Classification of the accessibility measures reviewed from the literature, from [4].

In the context of opportunity-based models, we will measure accessibility by considering the speed. Starting from the file 'results_vehicle_level.xlsx' introduced in 6.1 Increase in travel times by vehicle and by arc we used the columns 'Datetime_partenza' and 'Datetime_arrivo' to associate origin and destination to each trip, based on the zoning proposed in 6.5 Increase of arc travel times at zonal level. Four new columns have been created in a new file named 'origin_destination.xlsx' named respectively 'zone_origin' (containing the id of the zone), 'zone_O' (containing the name of the zone), 'zone_destination' and 'zone_D'.

By means of those columns, we created the origin/destination matrix, that we cannot report here entirely, because it is an 84x84 matrix. We only report an extract in Figure 73.

O/D Matrix		zone_destination	ZONE_D					
zone_origin	ZONE_O	esterna	Airasca	Alpignano	Baldissero Torinese	Beinasco	Borgaro Torinese	
0	esterna	38502	116	960	41	1474	483	
1	Airasca	30	36					
2	Alpignano	206		937		12	65	
3	Baldissero	198			804			
4	Beinasco	226		2		1598	13	
5	Borgaro To	144		23		5	1043	
6	Brandizzo	183					22	
7	Bruino	129	2					
8	Cambiano	121				15		
9	Candiolo	344					24	
10	Carignano	197						

Figure 73: Origin/Destination matrix, from 'origin_destination.xlsx'

Then, by means of the column 'speedKmh' present among the characteristics of each trip, as we can recall from 3 *HCD Metadata*, we created another 84x84 matrix, where each cell contains the average speed of all the trip sharing the same origin destination pattern. The partial structure of this new matrix can be seen in Figure 74, where the first row and the first column are related to trips which have either the origin or the destination (or both, in case of the top left cell) outside the study area.

Average speed (km/h)		zone_destination	ZONE_D					
zone_origin	ZONE_O	esterna	0	1	2	3	4	5
			Airasca	Alpignano	Baldissero Torinese	Beinasco	Borgaro Torinese	
0	esterna		45	69	49	32	57	50
1	Airasca		71	15				
2	Alpignano		42		13		32	39
3	Baldissero		53			20		
4	Beinasco		54		52		14	43
5	Borgaro To		57		34		69	19
6	Brandizzo		41					46
7	Bruino		49	47				
8	Cambiano		48				33	
9	Candiolo		48					67

Figure 74: Average speed matrix, from 'origin_destination.xlsx'

At the end of each row, we have the average weighted speed (weighted with the amount of movements for each cell) of all the trip starting from that specific zone. For example if we consider the zone 1 'Airasca', all the trip starting from there, have an average weighted speed of 46 km/h. Conversely, at the end of each column, we have the average weighted speed of all trip arriving to that specific zone.

To represent those results, we connect all pairs of centroids with lines in the shapefile 'desire_lines.shp', whose colour is representing a specific range of the above average speeds. Since we have $84 \times 84 = 7056$ lines, we cannot represent them all, so we started by representing only the desire lines relative to Turin urban area, filtered with a number of recorded trips of at least 250 to have meaningful results. The result can be appreciated in Figure 75.

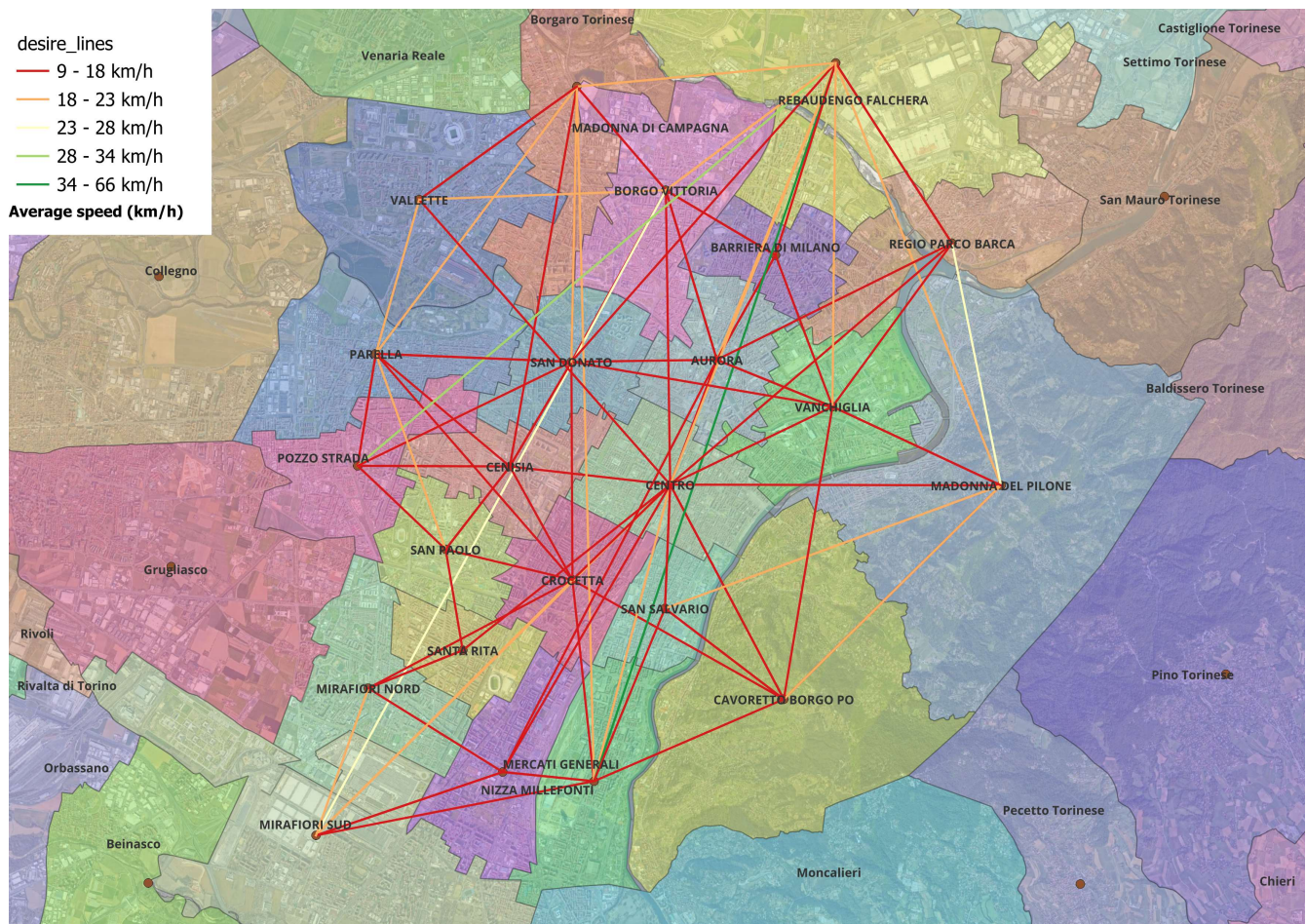


Figure 75: Desire lines of Turin

As observed, most movements within Turin occur at speeds ranging from 9 to 23 km/h, indicating a significant impact of congestion on travel. Then, we considered Turin as a whole big zone, and plotted desire lines of movements towards Turin, with at least 250 trips, in Figure 76:

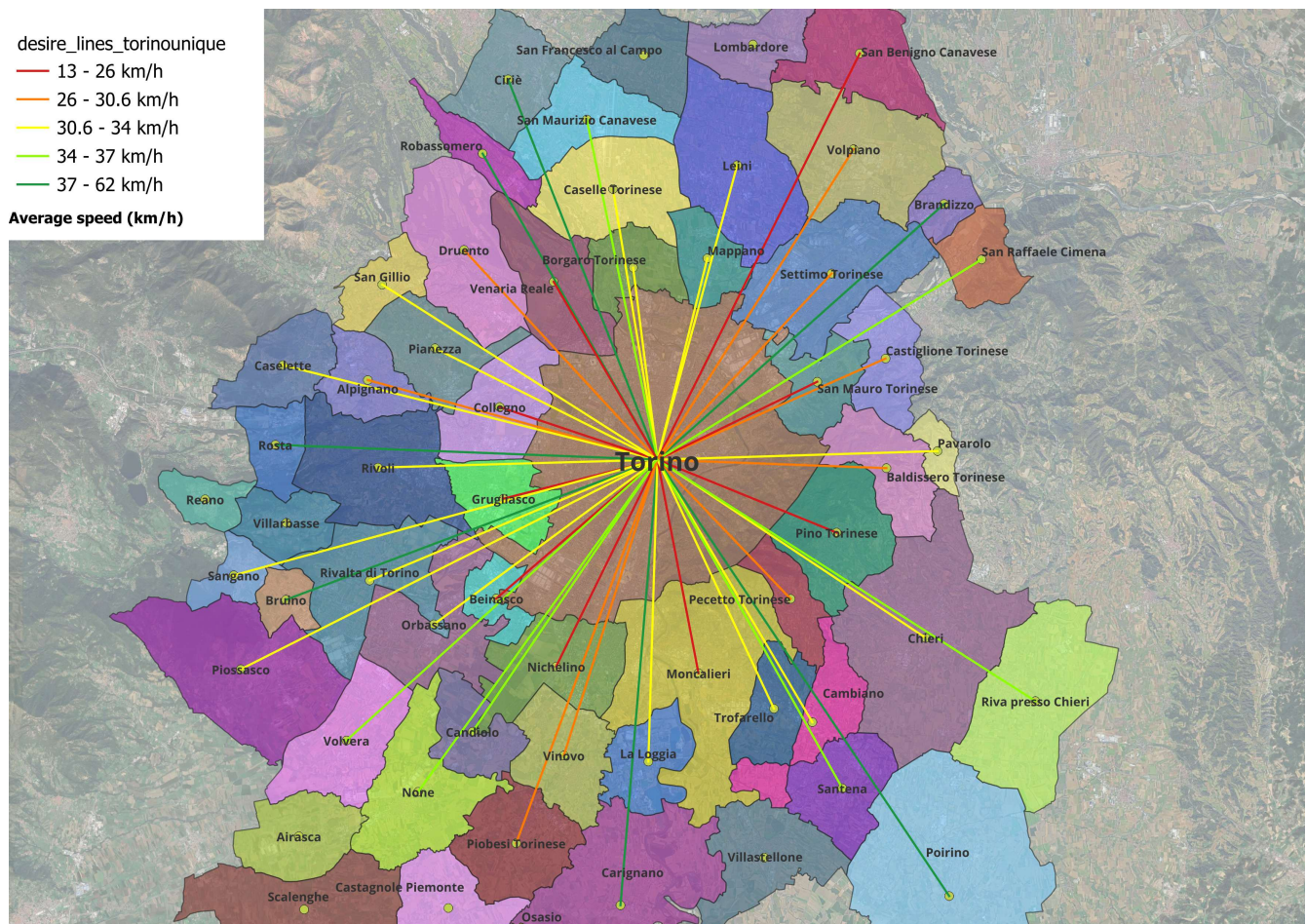


Figure 76: Desire lines towards Turin

Overall, speeds are higher when trips originate outside Turin and end within the city. This is likely due to the greater availability of roads with freer-flowing traffic compared to the more congested urban streets in the city center.

6.8 Increase in travel times for different travellers' categories

We are interested also in understanding the characteristics of the trip maker, therefore, we selected 3 independent classes:

- Vehicle typology (private car or fleet);
- Holder (male, female, legal entity) ;
- Gender and age of holder (if not legal entity): male or female crossed with 18-25 / 25-35 / 35-45 / 45-55 / 55-65 / >65;

We used the excel file introduced in 6.1 *Increase in travel times by vehicle and by arc*, 'results_vehicle_level.xlsx' and created pivot tables. Starting by the vehicle typology, we summed up the variables 'wasted_time_vehicle' for each vehicle type to get the total time wasted and 'tt_disaggregate' to get the total travel time. Most of time wasted is attributable to private cars (85%). However, the highest percentage can be due to the higher number of private cars, so

effectively, what we have to look at, is not the absolute number but we have to divide by the count of elements inside each class, getting the average wasted time per type of vehicle; moreover we also plotted the average speed as the ratio between the total distance traveled and the total travel time, the result can be appreciated in Figure 77:

Type	Type_count	km_traveled	Total travel time (h)	Total_Wasted_time_vehicle (h)	Average_wasted_time (min/type)	av_speed (km/h)
car	271966	390628	53376	47587	10.5	8.2
fleet	34626	59405	2413	1766	3.1	33.6
unknown	36225	53041	7144	6367		
Total	342817	503074	62932	55719		

Figure 77: Pivot table with vehicle type characteristics (private or commercial), from 'results_vehicle_level.xlsx'

Of course private cars are those which travel more, they waste more time in congestion and have a lower average speed, this could be due to the fact that some commercial vehicles could have reserved lanes for example. Concerning the gender, we did the same, and we report the pivot table in Figure 78:

Gender	Gender_count	km_traveled	Total travel time (h)	Total_Wasted_time_vehicle (h)	Average_wasted_time (min/gender)	av_speed (km/h)
F	97471	137373	15730	13630	8.4	8.7
M	148568	212956	29974	26869	10.9	7.1
unknown	96778	152744	17229	15220		
Total	246039	503074	62932	55719		

Figure 78: Pivot table with gender characteristics, from 'results_vehicle_level.xlsx'

On average, males waste more time in congestion rather than females and have a lower average speed.

Now, at the end, we have to cross the gender with the age and to have a better visualization and to make comparison between each class, we constructed a three entry table, where the variables plotted in the third dimension of the table are the following five:

- km traveled;
- Total travel time (h);
- Total wasted time (h);
- Average wasted time (min/class);
- Average speed (km/h);

The results are plotted in Figure 79.

Five entry table		Age							
Gender		na	18-25	25-35	35-45	45-55	55-65	>65	Total
F	km_traveled	18132	2667	19079	21440	33637	27952	14466	137373
	Total_travel_time (h)	2016	296	2442	2184	3901	2993	1898	15730
	Total_wasted_time (h)	1735	255	2154	1863	3391	2566	1666	13630
	Average_wasted_time (min/class)	8	8	10	8	9	8	10	
	Average speed (km/h)	9.0	9.0	7.8	9.8	8.6	9.3	7.6	
M	km_traveled	26018	4992	22710	30404	40150	52641	36041	212956
	Total_travel_time (h)	2067	603	2074	6115	7795	6841	4480	29974
	Total_wasted_time (h)	1675	528	1746	5682	7210	6102	3926	26869
	Average_wasted_time (min/class)	5	9	7	16	15	10	9	
	Average speed (km/h)	12.6	8.3	11.0	5.0	5.2	7.7	8.0	

Figure 79: Three entry pivot table crossing gender and age ranges, from 'results_vehicle_level.xlsx'

Focusing on females, the 45-55 age group travels the most (33,637 km) and experiences the highest total wasted time (3,391 hours). However, on average, women over 65 spend the most time in congestion, with an average delay of 10 minutes per trip.

For males, those aged 55-65 travel the most, while the 45-55 age group accumulates the highest total wasted time (7,210 hours), likely due to travelling more during peak hours. On average, however, the 35-45 age group experiences the longest delays, wasting 16 minutes per trip.

Comparing genders, men travel more and experience higher total wasted time overall. This could be mostly due to different habits, in fact according to [11], women use more public transport rather than men.

7 Conclusions

In an increasingly connected world where big data has become a vital resource, floating car data could provide us a reliable support in managing traffic and planning the transport system.

Starting with the fundamental and straightforward question: "Which arcs in Turin experience the highest time loss due to congestion?" we followed a structured methodology consisting on three fundamental steps: raw data preprocessing, map matching and extrapolation of results. With raw data preprocessing, we started by treating a dataset of GPS traces of vehicles travelling in Turin that was provided by TIM. Two different kinds of data were available, namely Historical Car Data (HCD) and Floating Car Data (FCD). The former, that are assembled the day after the measurement campaign at 9 AM, are more compact and are divided in heading (characteristics of the trip along with trip maker characteristics) and details (spatial and temporal information on the successive position of the vehicle during the trip). Their processing enabled us to determine the travel times required to traverse each arc of the graph under ideal conditions, which we assessed exclusively during nighttime. With one year of data available, we based our calculations on 102 out of 366 days. While including all sampled days would have yielded even more precise results, we believe the current dataset provides a solid foundation and remains sufficiently reliable. The latter data, namely the FCD, were utilized to assess congestion levels and derive insights. This analysis was conducted on only four days out of the entire year due to computational constraints. To ensure a more homogeneous analysis, we followed a structured sampling plan when selecting these days. Naturally, using the complete dataset would have provided a more representative picture of the overall situation. However, this type of analysis can be performed on a single day or over an entire year, depending on the specific objectives. All those data have been imported on Qgis, where each vehicle was represented with a GPS trace, that naturally had to be matched with the arcs.

Here comes the map matching phase, in which, through a series of steps, we were able to associate each point to the correct arc. Here one limitation was due to the fact that it is not convenient to consider a graph where all the arcs have been represented, rather than focusing on a graph representing only the main roads, so that we had to discard a good portion of data; expanding the dataset to include more roads would further enhance the accuracy and applicability of the analysis. During this phase we exploited the functionalities of Qgis and the API of Openrouteservice (ORS) to snap the points to the graph and then correctly associate the correct arc with its main geometric and functional characteristics to each point. Once did this, we were ready with the last part, thus the extrapolation of results. This last part, we can say it's divided in two phases, the first one, little bit more addressing from a computational view point, where we exploited the matched points, to determine speed and travel times of each arc through a series of algorithms properly explained in each paragraph; the second one, where we used those travel times and speed to extract results and make our consideration, which is the most interesting part of this work, where we practically meet the objective of the thesis.

We came up with a set of tables and images that identified the most critical arcs and zones within the study area in terms of time losses caused by congestion.

We carried analysis at three main levels, vehicle, arcs and zones.

At vehicle level, we analysed the distribution of wasted travel time for each vehicle on each arc, highlighting a situation in which the network is generally efficient with some exceptions where the wasted times were really high. At arc level, we analysed the excess in travel times to travel a certain arc with respect to the free flow travel time, both during only peak hours or considering the entire day. Overall each analysis showed what are the most critical arcs from the two viewpoints, the average time wasted by each vehicle, which give us a measure of how much time is wasted on

average from one vehicle passing from there, and the total time wasted, which is, we can say, a more complete information, because it takes into account also of the importance of the single arc. In fact, if on a certain arc the time lost is for example 100 minutes for each vehicle but only few vehicle passes from there, probably the arc has some problem but this is not affecting the overall social benefit too much. On the other hand, there could be some arcs where the average time wasted is 10 minutes per vehicle, but a lot of vehicles passes from there, in this case, the problem is related not to the arc itself but to the traffic demand, and this is affecting a lot the congestion on the network. We faced results confirming our statement, for instance Corso Toscana is the road with the highest average time wasted by each vehicle, but effectively, Corso Regina Margherita, is the road where most time is wasted, so from a transport planner view point, if we want to solve problem of congestion, we have to act on the second mentioned road.

Moreover, the analysis carried out during peak hours compared with results of analysis carried out during the entire day, allowed us to understand if the problem of traffic is more related with peak hours or if it is constant during all day, to understand with which kind of policy we have to act.

At zonal level, inside Turin we used the zoning according to neighbourhood while out from Turin we simply considered the administrative boundaries of each municipality. For each zone, we carried out two analyses with a different level of aggregation. Firstly, we aggregated the total time wasted on each arc at zonal level dividing it by the total length of arcs because the result otherwise could be strongly affected from this aspect, and we determined the zones where most time is wasted due to congestion. Secondly, we aggregated the wasted time at vehicular level, according to the zone where they travelled, and divided by the number of vehicles inside each zone, determining the most congested zones. Both analyses gave the same results, with the zones of the central urban area of Turin which are the most congested. It is however important to consider that, by means of the analysis carried out during peak hours and during the whole day, we noticed there are some zones which are green (meaning not too much time is wasted) if consider the whole day, that become red (meaning a lot of time is wasted) if only peak hours are considered. This means that the traffic is there concentrated during peak hours, maybe because there is a lot of pendolarism. For example, Virle piemonte, Vinovo and Trofarello, are municipalities with the just mentioned characteristic. This information could be helpful to manage traffic for administrators. So, despite computational limitations, our findings were both reasonable and insightful. Concerning policy implications, municipalities could use HCD to conduct an analysis similar to ours and then leverage real-time FCD for short-term decision-making. Alternatively, they could analyse an entire year of FCD to inform long-term strategies and identify persistent network issues.

Finally, there is still much work to be done in this field, but this methodology proves to be a highly effective tool for managing traffic and infrastructure, both in the short and long term.

Bibliography

- [1] Agenzia della mobilità piemontese. *Indagine sulla Mobilità delle Persone e sulla Qualità dei Trasporti in Piemonte - IMQ 2022*. Tech. rep. Report completo sulla mobilità e qualità del trasporto in Piemonte basato su indagini campionarie telefoniche. Piemonte, Italia: Agenzia della mobilità piemontese, 2022. URL: <https://mtm.torino.it>.
- [2] Oruc Altintasi, Hediye Tuydes-Yaman, and Kagan Tuncay. “Detection of urban traffic patterns from Floating Car Data (FCD)”. In: *Transportation research procedia* 22 (2017), pp. 382–391.
- [3] Transportation Research Board. *1. HCM User’s Guide*. 2022. URL: <https://app.knovel.com/hotlink/khtml/id:kt01310I51/highway-capacity-manual/hcm-users-guide>.
- [4] Ennio Cascetta, Armando Carteni, and Marcello Montanino. “A new measure of accessibility based on perceived opportunities”. In: *Procedia-Social and Behavioral Sciences* 87 (2013), pp. 117–132.
- [5] Ioannis Chatziioannou et al. “A Structural Analysis for the Categorization of the Negative Externalities of Transport and the Hierarchical Organization of Sustainable Mobility’s Strategies”. In: *Sustainability* 12.15 (July 2020), p. 6011. URL: <https://www.mdpi.com/2071-1050/12/15/6011>.
- [6] Città Metropolitana di Torino. *Piano Urbano della Mobilità Sostenibile (PUMS)*. 2022. URL: <http://www.cittametropolitana.torino.it/cms/trasporti-mobilita-sostenibile/pums/pums-piano-approvato-2022>.
- [7] Comune di Bologna. *Approvazione del Piano Particolareggiato del Traffico Urbano (PPTU): Bologna Città 30,410039 / 2023*. 2024. URL: https://atti9.comune.bologna.it/atti/wpub_delibere.nsf/cercaDC.xsp.
- [8] *Decreto Ministeriale 6792 del 2001*. Decreto Ministeriale. Pubblicato nel 2001. Italia.
- [9] Jan Fabian Ehmke, Stephan Meisel, and Dirk Christian Mattfeld. “Floating car based travel times for city logistics”. In: *Transportation research part C: emerging technologies* 21.1 (2012), pp. 338–352.
- [10] Tomislav Erdelić et al. “Estimating congestion zones and travel time indexes based on the floating car data”. In: *Computers, Environment and Urban Systems* 87 (2021), p. 101604.
- [11] Istituto Superiore di Formazione e Ricerca per i Trasporti (ISFORT). *21° Rapporto sulla mobilità degli italiani*. 2024. URL: <https://www.isfort.it/progetti/21-rapporto-sulla-mobilita-degli-italiani-audimob/>.
- [12] Ajay Kumar Gupta and Udai Shanker. “A comprehensive review of map-matching techniques: Empirical analysis, taxonomy, and emerging research trends”. In: *International Journal of Web Services Research (IJWSR)* 19.1 (2022), pp. 1–32.

- [13] Erik Jenelius and Haris N Koutsopoulos. “Travel time estimation for urban road networks using low frequency probe vehicle data”. In: *Transportation Research Part B: Methodological* 53 (2013), pp. 64–81.
- [14] Flavio Pallavicino. “Stima dei flussi di traffico attraverso dati di localizzazione da celle telefoniche”. Supervisors: Prof. Ing. Marco Diana, Ing. Andrea Chicco. Tesi di Laurea Magistrale. Turin, Italy: Politecnico di Torino, Dec. 2022.
- [15] Miriam Pirra and Marco Diana. “Integrating mobility data sources to define and quantify a vehicle-level congestion indicator: an application for the city of Turin”. In: *European transport research review* 11 (2019), pp. 1–11.
- [16] Mohammed A Quddus. “High integrity map matching algorithms for advanced transport telematics applications”. PhD thesis. Imperial College London London, 2006.
- [17] Mahmood Rahmani, Erik Jenelius, and Haris N Koutsopoulos. “Route travel time estimation using low-frequency floating car data”. In: *16th international ieee conference on intelligent transportation systems (itsc 2013)*. IEEE. 2013, pp. 2292–2297.
- [18] Irum Sanaullah, Mohammed Quddus, and Marcus Enoch. “Developing travel time estimation methods using sparse GPS data”. In: *Journal of Intelligent Transportation Systems* 20.6 (2016), pp. 532–544.
- [19] Chaoyang Shi, Bi Yu Chen, and Qingquan Li. “Estimation of travel time distributions in urban road networks using low-frequency floating car data”. In: *ISPRS International Journal of Geo-Information* 6.8 (2017), p. 253.
- [20] H Van der Loop et al. “Validation and usability of floating car data for transportation policy research”. In: *World Conference on Transport Research-WCTR*. 2019.
- [21] Christopher E White, David Bernstein, and Alain L Kornhauser. “Some map matching algorithms for personal navigation assistants”. In: *Transportation research part c: emerging technologies* 8.1-6 (2000), pp. 91–108.
- [22] Zhaosheng Yang, Bowen Gong, and Ciyun Lin. “Travel time estimate based on floating car”. In: *2009 Second International Conference on Intelligent Computation Technology and Automation*. Vol. 3. IEEE. 2009, pp. 868–871.

APPENDIX

1 Map Matching code

Here is the Python code we developed with the support of ChatGPT to carry out the first step of map matching:

```
import csv
import json
import requests

# Imposta la tua chiave API di OpenRouteService
api_key = '5b3ce3597851110001cf624889a69173e7334d9892d083113bfcf97d'

# Percorso al file CSV
csv_file_path = 'C:/Users/Asus/Desktop/tesi/SampleData190115-7-8/
SampleData190115-7-8/merged.csv'

# Endpoint dell'API di OpenRouteService per il map snapping
ors_url = 'https://api.openrouteservice.org/v2/snap/driving-car'

# Dimensione massima del batch per rispettare i limiti dell'API
batch_size = 2000

# Lista per memorizzare le righe originali dal CSV
rows = []

# Leggere il file CSV ed estrarre tutte le colonne
with open(csv_file_path, mode='r') as csvfile:
    reader = csv.DictReader(csvfile)
    all_locations = []

    # Estrarre tutte le coordinate dal file CSV
    for row in reader:
        # Salva la riga originale
        rows.append(row)
        # Aggiungi solo le coordinate per l'invio all'API
        all_locations.append([float(row['longitude']), float(row['
latitude'])])

# Suddividi le coordinate in batch pi piccoli
batches = [all_locations[i:i + batch_size] for i in range(0, len(
all_locations), batch_size)]

# Inizializziamo una lista per memorizzare tutte le risposte dell'API
all_snapped_points = []
```

```

# Invia ogni batch separatamente
for batch_num, batch in enumerate(batches):
    json_data = {
        "locations": batch
    }

    # Convertire i dati in formato JSON
    json_output = json.dumps(json_data, indent=4)
    print(f"Inviando batch {batch_num + 1} su {len(batches)}...")

    # Inviare la richiesta POST all'API di OpenRouteService
    response = requests.post(
        ors_url,
        headers={
            'Authorization': api_key,
            'Content-Type': 'application/json'
        },
        data=json_output
    )

    # Verifica se la richiesta andata a buon fine
    if response.status_code == 200:
        print(f"Batch {batch_num + 1} eseguito con successo!")
        response_data = response.json()

        # Estrai le coordinate abbinate sotto la chiave 'location', se
        # esiste
        for snapped in response_data['locations']:
            if snapped and 'location' in snapped: # Verifica che '
                location' esista e non sia None
                all_snapped_points.append(snapped['location']) #
                Aggiungi solo le coordinate abbinate
            else:
                # Se 'location' non esiste o None, inserisci valori di default
                all_snapped_points.append([None, None])
        # Puoi scegliere altri valori di default
        else:
            print(f"Errore nella richiesta per il batch {batch_num + 1}: {
                response.status_code}")
            print("Messaggio di errore:", response.text)
            break # Interrompi in caso di errore

# Aggiungere le nuove coordinate abbinate a ciascuna riga del CSV
for i, snapped in enumerate(all_snapped_points):
    rows[i]['snapped_longitude'] = snapped[0] # Prima coordinata (
        longitude)

```

```

    rows[i]['snapped_latitude'] = snapped[1] # Seconda coordinata (
        latitude)

# Scrivere il nuovo file CSV con tutte le colonne originali e le
coordinate abbinate
output_csv_path = 'C:/Users/Asus/Desktop/tesi/SampleData190115-7-8/
SampleData190115-7-8/snapped_output.csv'
with open(output_csv_path, mode='w', newline='') as csvfile:
    fieldnames = list(rows[0].keys()) # Ottiene tutte le colonne
        originali e le nuove coordinate
    writer = csv.DictWriter(csvfile, fieldnames=fieldnames)

    writer.writeheader()
    writer.writerows(rows)

print(f" Risultato finale salvato in '{output_csv_path}'")

```

2 Counting deviceId-Arco Occurrences in R

Here is the R code developed with the support of ChatGPT to counts how many times one vehicle occur on a certain arc during same day:

```

# Carica i pacchetti necessari
library(readxl)
library(dplyr)
library(openxlsx)

# Specifica il percorso del file Excel
file_path <- 'C:/Users/Asus/Desktop/tesi/shapefile_wholedataset/
matched_jan-apr.xlsx'

# Leggi il dataset dal file Excel
df <- read_excel(file_path)

# Assumiamo che il dataset contenga una colonna 'Datetime' da cui
estrarre il giorno
# Crea una nuova colonna "Giorno" basata sul giorno del timestamp
df <- df %>%
    mutate(Giorno = as.Date(Datetime))

# Conta quante volte un Device_Id associato a un Arco in uno stesso
Giorno
count_df <- df %>%
    group_by(Device_Id, Arco, Giorno) %>%
    summarise(counts = n()) %>%
    ungroup()

```

```

# Aggiungi la colonna del conteggio al dataset originale
df_with_counts <- df %>%
  left_join(count_df, by = c("Device_Id", "Arco", "Giorno"))

# Sovrascrivi il file Excel originale con i dati filtrati
write.xlsx(df_with_counts, file_path, overwrite = TRUE)

cat("File Excel aggiornato con i punti filtrati: ogni Arco ha almeno 2
punti nello stesso giorno.\n")

```

3 Computation of average number of visits on each arc

Here is the R script, developed with the support of ChatGPT, to count the average number of visit on each arc:

```

# Carica i pacchetti necessari
library(readxl)
library(dplyr)

# Percorso del file Excel
file_path <- "C:/Users/Asus/Desktop/tesi/recap_sperimental_shp_excel/
matched_processed.xlsx"

# Carica il file Excel
df <- read_excel(file_path, sheet="FALSO")

# Step 1: Raggruppa per 'Arco' e conta quante volte stato visitato
e da quanti deviceId diversi
arco_stats <- df %>%
  group_by(Arco) %>%
  summarise(
    numero_visite = n(), # Conta quante volte l'arco stato
    visitato (visite totali)
    numero_device_id = n_distinct(deviceId) # Conta quanti deviceId
    unici hanno visitato l'arco
  )

# Step 2: Ordina i risultati per 'Arco'
arco_stats <- arco_stats %>%
  arrange(Arco)

# Visualizza i risultati
print(arco_stats)

# Step 3: Se desideri salvare i risultati in un file Excel
library(writexl)
write.xlsx(arco_stats, "C:/Users/Asus/Desktop/tesi/arco_stats.xlsx")

```


4 Computation of distances on the graph

Here is the Python script, developed with the support of ChatGPT, to measure the distances among points on the graph:

```
import geopandas as gpd
from shapely.geometry import Point
import pandas as pd

# 1. Caricamento dei dati dal grafo e dalle tracce GPS
arcs_gdf = gpd.read_file("C:/Users/Asus/Desktop/tesi/
    recap_sperimental_shp_excel/graph_reprojected.shp") # Shapefile con
    archi del grafo
gps_gdf = gpd.read_file("C:/Users/Asus/Desktop/tesi/
    shapefile_wholedataset/matched_processed_jan-apr.shp") # Shapefile
    con tracce GPS

# 2. Creazione di un dizionario per mappare ciascun Arco alla sua
    geometria e lunghezza
arcs_dict = {str(row.Arco): {'geometry': row.geometry, 'length': row.
    geometry.length} for _, row in arcs_gdf.iterrows()}

# 3. Raggruppamento delle tracce GPS per giorno, deviceId e datettime
gps_grouped = gps_gdf.groupby(['Giorno', 'Device_Id', 'Arco'])

# 4. Calcolo delle distanze sul grafo tra i punti sullo stesso arco
distances = [] # Lista per salvare i risultati

for (giorno, device_id, arc_id), group in gps_grouped:
    # Ordina per datettime
    group = group.sort_values(by='Datetime')

    # Prendi la geometria dell'arco corrispondente dal dizionario
    arc_data = arcs_dict.get(str(arc_id))
    if arc_data is None:
        continue # Se l'arco non esiste, passa oltre
    arc_geom = arc_data['geometry']
    arc_length = arc_data['length']

    # Itera sui punti consecutivi
    for i in range(len(group) - 1):
        # Coordinate e informazioni dei punti consecutivi
        p1, p2 = group.geometry.iloc[i], group.geometry.iloc[i + 1]
        datetime1, datetime2 = group.Datetime.iloc[i], group.Datetime.
            iloc[i + 1]
        speed1, speed2 = group.SpeedKmh.iloc[i], group.SpeedKmh.iloc[i
            + 1]
```

```

# Calcola la posizione dei punti GPS lungo la geometria dell'
    arco
position_p1 = arc_geom.project(p1)
position_p2 = arc_geom.project(p2)

# Calcola la distanza lungo la geometria dell'arco
distance = abs(position_p2 - position_p1)

# Calcola la distanza tra il primo punto e il nodo di inizio
distance_to_start = position_p1

# Calcola la distanza tra il secondo punto e il nodo di fine
distance_to_end = arc_length - position_p2

# Aggiungi le informazioni al dizionario dei risultati
distances.append({
    'Giorno': giorno,
    'Device_Id ': device_id,
    'Arco ': arc_id,
    'Datetime1 ': datetime1,
    'Datetime2 ': datetime2,
    'Speed1 ': speed1,
    'Speed2 ': speed2,
    'Distance ': distance,
    'Arc_Length ': arc_length,
    'Distance_to_Start ': distance_to_start,
    'Distance_to_End ': distance_to_end
})

# 5. Creazione di un DataFrame finale con i risultati
distance_df = pd.DataFrame(distances)

# 6. Salvataggio del risultato in un file CSV
output_path = "C:/Users/Asus/Desktop/tesi/distanze_device_arco_giorno.
    csv"
distance_df.to_csv(output_path, index=False)
print(f" Risultato salvato in: {output_path}")

```

5 Computation of points' relative position on the arcs

Here is the Python script, developed with the support of ChatGPT, to determine the relative position of the points on each arc, in order to estimate the free flow speed:

```

import geopandas as gpd
import pandas as pd

# Caricamento dei dati degli archi e dei punti GPS

```

```

arcs_gdf = gpd.read_file("C:/Users/Asus/Desktop/tesi/
    recap_sperimental_shp_excel/graph_reprojected.shp") # Sostituire
    con il percorso corretto
gps_gdf = gpd.read_file("C:/Users/Asus/Desktop/tesi/
    shapefile_wholedataset_copy/merged_matched_processed.shp") #
    Sostituire con il percorso corretto
print(gps_gdf.columns)
print(arcs_gdf.columns)

# Dizionario per associare ciascun arco alla geometria, lunghezza e
    velocit
arcs_dict = {
    str(row.Arco): {
        'geometry': row.geometry,
        'length': row.geometry.length,
        'Vf_kmh': row['Vf [km/h]'],
        'Main_Class': row.Main_Class # Aggiunge la colonna Main_Class
    }
    for _, row in arcs_gdf.iterrows()
}

# Lista per raccogliere i risultati
points = []

# Raggruppa i punti GPS per deviceId e Arco
gps_grouped = gps_gdf.groupby(['Giorno', 'Device_Id', 'Arco'])

for (day, device_id, arc_id), group in gps_grouped:
    # Ordina i punti del gruppo per dateTime
    group = group.sort_values(by='Datetime')

    # Ottieni la geometria, la lunghezza e la velocit dell'arco dal
        dizionario
    arc_data = arcs_dict.get(str(arc_id))
    if arc_data is None:
        continue # Passa oltre se l'arco non trovato
    arc_geom = arc_data['geometry']
    arc_length = arc_data['length']
    arc_speed = arc_data['Vf_kmh']
    road_class = arc_data['Main_Class'] # Velocit dal dizionario

    # Itera sui punti del gruppo
    for _, row in group.iterrows():
        point = row.geometry
        datetime = row.Datetime # Data e ora del punto
        speed = row['SpeedKmh'] # Velocit istantanea direttamente
            dal file

```

```

# Calcola la distanza del punto dall'inizio dell'arco
distance_to_start = arc_geom.project(point)

points.append({
    'Device_Id ': device_id ,
    'Arco ': arc_id ,
    'distance_to_start ': distance_to_start ,
    'speed ': speed ,
    'Datetime ': datetime ,
    'arc_length ': arc_length , # Lunghezza dell'arco
    'Vf_kmh ': arc_speed ,
    'Main_Class ': road_class })

# Creazione di un DataFrame con i risultati
points_df = pd.DataFrame(points)

# Salvataggio in un file CSV
output_points_path = "C:/Users/Asus/Desktop/tesi/
    points_with_distances_speed_length_datetime.csv" # Sostituire con
    il percorso desiderato
points_df.to_csv(output_points_path , index=False)
print(f" Risultato salvato in: {output_points_path}")

```

6 Calculation of mean travel time for each deviceId on each arc

Here is the Python script, developed with the support of ChatGPT, to determine the average travel time of each deviceId traveling on the same arc:

```

library(dplyr)

# Leggi il file CSV
# Sostituisci 'percorso_del_file.csv' con il percorso effettivo del
    tuo file
data <- read.csv("C:/Users/Asus/Desktop/tesi/
    recap_sperimental_shp_excel/distanze_elaborazione.csv")

# Raggruppamento per Arco e deviceId e calcolo della media di TT..min
risultati <- data %>%
    group_by(Arco , deviceId) %>%
    summarise(media_TT_min = mean(TT..min. , na.rm = TRUE))

# Aggiungi la colonna al dataset originale
# Effettua un join per unire il tempo medio al dataset originale
data <- data %>%

```

```
left_join(risultati, by = c("Arco", "deviceId"))

# Visualizza i risultati
print(head(data))

# Salva il dataset aggiornato con la nuova colonna
data <- write.csv(data, "C:/Users/Asus/Desktop/tesi/
  recap_sperimental_shp_excel/dataset_aggiornato.csv", row.names =
  FALSE)
```

7 Different categories of Map Matching methods

Basis of Classification	Classes	Policies/ Papers
Based on Use of Road Network and Trajectory Details	Topological, Geometrical	Topological (Greenfeld, 2002)(Liu et al., 2017)(Schwertfeger & Yu, 2016)(Velaga et al., 2009) Geometrical (Abdallah et al., 2011)
Based On Range of Used Trajectory	Incremental (online, real-time), Global (post-process, offline)	Incremental (White et al., 2000) (J. Yang et al., 2005) (Carola A Blazquez & Vonderohe, 2005) (Li et al., 2008) (Greenfeld, 2002) (M. A. Quddus et al., 2003) (Velaga et al., 2009) (M. Quddus & Washington, 2015) (Griffin et al., 2011) (Mazhelis, 2010), Global (Marchal et al., 2005) (Lou et al., 2009) (Oliver Pink & Hummel, 2008) (Thiagarajan et al., 2009) (Newson & Krumm, 2009)
Based On Type of Algorithm	Weighted-algorithms, fuzzy-logic, Machine-Learning, HMM, Fuzzy-Logic, Particle-Filter, Kalman	Weighted-algorithms (Carpin, 2008)(Abdallah et al., 2011) fuzzy-logic (Gupta & Shanker, 2020d) (Gupta & Shanker, 2022b) (Gupta & Shanker, 2021b) Smoothing (Hsueh & Chen, 2018)(Cao & Krumm, 2009) ARIMA (Yan, 2010) Kalman Filter (O Pink & Hummel, 2008)(Cho & Choi, 2014) (M. A. Quddus et al., 2003) Non-Para Metric Regression (Nagaraj & Mohanraj, 2020) (Sharath et al., 2019) Neural Network (K. Zheng et al., 2012)
Based on Type of Sensors	GPS, DR, DEM	GPS (Greenfeld, 2002) (M. Quddus & Washington, 2015) (Hashemi & Karimi, 2014) (Wu & Wu, 2003) DR (M. A. Quddus et al., 2003) (Velaga et al., 2009)(Pyo et al., 2001) (M. A. Quddus et al., 2006) DEM (Ahmad et al., 2017)
Based on Type of Environment	Outdoor, Indoor	Outdoor (Y. Zheng et al., 2011)(Abowd et al., 1997) Indoor (Tian et al., 2015)(Petrou et al., 2014)
Based on the Type of Tracked object	Wheelchair, Vehicle, Pedestrian	Wheelchair (Ren & Karimi, 2009)(Ren, 2012) Vehicle (Jagadeesh et al., 2004)(M. Quddus & Washington, 2015) Pedestrian (Shin et al., 2010)(Ren, 2012)
Based on Type of Sampling	Low-Sampling, High-Sampling	Low-Sampling (M. Quddus & Washington, 2015)(Lou et al., 2009)(K. Zheng et al., 2012) High-Sampling
Based on Processing of Algorithm	Post-processing, Real-Time	Post-processing (Knappen et al., 2018)(Rappos et al., 2018) Real Time (Algizawy et al., 2017)(Goh et al., 2012)

Figure 80: Review of Map Matching methods, from [12]

8 Association of deviceId to correct Trip

```
import pandas as pd

# Percorsi dei file CSV
file_viaggi = "D:/NAS/ViaggiHCD/polito_viaggi_2019_12_12.csv"
file_tracce = "D:/NAS/12-12_FCD/MERGED/merged_FCD_20-24_1212.csv"

# 1      Caricare i file con parsing delle date
```

```

df_viaggi = pd.read_csv(file_viaggi , parse_dates=['Datetime_partenza ',
'Datetime_arrivo '])
df_tracce = pd.read_csv(file_tracce , parse_dates=['dateTime'])

# 2      Assicurarsi che gli ID dei dispositivi siano stringhe per
        evitare problemi di join
df_viaggi['Device_Id'] = df_viaggi['Device_Id'].astype(str)
df_tracce['deviceId'] = df_tracce['deviceId'].astype(str)

# 3      Unire i dati basandosi sul deviceId mantenendo TUTTE le
        tracce GPS
df_merge = df_tracce.merge(df_viaggi , left_on='deviceId' , right_on='
Device_Id' , how='left ')

# 4      Aggiungere una colonna che indica se la traccia      associata
        a un viaggio
df_merge['Associato_a_viaggio'] = (
    (df_merge['dateTime'] >= df_merge['Datetime_partenza']) &
    (df_merge['dateTime'] <= df_merge['Datetime_arrivo']))
).fillna(False) # Se non c'      un viaggio , riempire con False

# 5      Rimuovere la colonna duplicata 'Device_Id' poich      gi
        presente come 'deviceId'
df_merge.drop(columns=['Device_Id'] , inplace=True, errors='ignore')

# 6      Salvare il dataset unito con TUTTE le tracce GPS
output_file = "C:/Users/Asus/Desktop/tesi/fcd_excel/
daimportare12122019/merged_FCD_20-24_1212+.csv"
df_merge.to_csv(output_file , index=False)

print(f"      File '{output_file}' creato con successo!")

```