From Data to Digital Twin for Smart Mobility

Course for the PhD program Information Engineering
2023-2024

Lesson #2

Traffic Flow Reconstruction

20th June 2024

Marco Fanfani – Stefano Bilotta

Traffic Flow Reconstruction

FROM DATA TO DIGITAL TWIN

FOR SMART MOBILITY

Introduction

- **Smart Mobility** is a tool to achieve **sustainable** development of cities, including: technology, mobility infrastructure, mobility solutions and people.
- Mobility solutions depend on traffic status in the road network.
- The knowledge of the **real-time traffic flow status in each segment** of a whole road network in a city or area is becoming fundamental for a large number of smart services such as: routing, planning, dynamic tuning services, healthy walk, etc.

Motivations

- Often, traffic flow estimation is related to a monitored area based on few fixed points/sensors and thus no information is provided in other connected road segments free of sensors.

- Traffic density measures are typically obtained by stationary sensors on fixed positions and they are usually of different kinds: TV cameras, road spires, etc.

Motivations

- Due to sustainability reasons, the number of deployed sensors has to be **limited**.

- Thus, it is mandatory to adopt some reconstruction algorithms to obtain the traffic flow condition in each road segment of the city in order to have dense traffic flows in the unmeasured road segments.

Definitions

- The **Traffic Flow Reconstruction**, *TFR*, is the process to estimate dense traffic density (flow) – e.g., vehicle per meter (or vehicles per second) – for each road segment within the road network by starting from a limited number of traffic flow sensors having fixed positions in the network (or data providing traffic density (flow) in the roads, or velocity in some cases) at the same time instant.

- It can be regarded as an extrapolation approach passing, for example, from 100 sensors data to 10.000 traffic flow data of road segments.

Overview

Goal

- Obtain a **traffic reconstruction** in every road in **real-time**, at **low-cost**, using a general and self-adaptive model

How

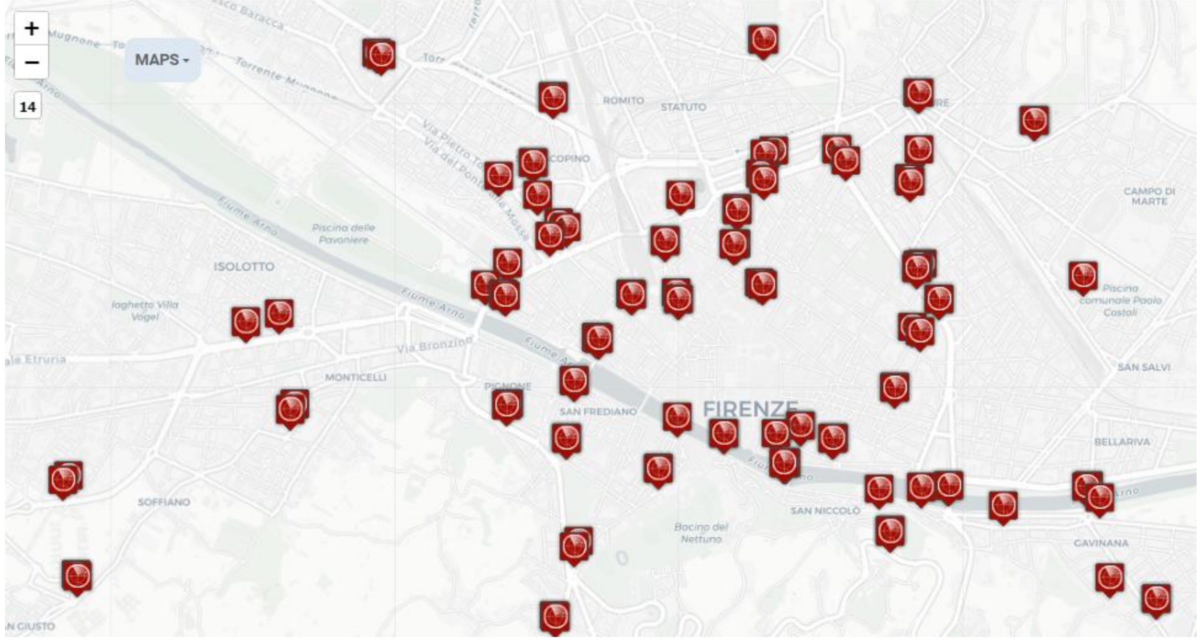
- Exploit a **fluid-dynamic model** adapted for the road network
- **Road graph** and possible **restriction** obtained from KM4City Knowledge Base, using Open Street Map data
- Traffic measurements from **IoT sensors** scattered over the municipality

Features

- **Dense**: the reconstruction is obtained at every location in the area of interest
- **General**: no or minimal simplistic assumption
- **Low-cost**: the algorithm use sensors already available, not requiring specific deployment
- **Real-time**: the reconstruction is updated frequently, each time a new traffic measurement is available
- **Verified**: the reconstruction accuracy is rigorously evaluated
- **Visual**: the reconstruction can be displayed over the road graph exploiting a given color map
- **Easy to use**: the user does not need to take any action (install apps, submit data, etc.)
- **Open**: methods and software are made available under open licenses

Goal

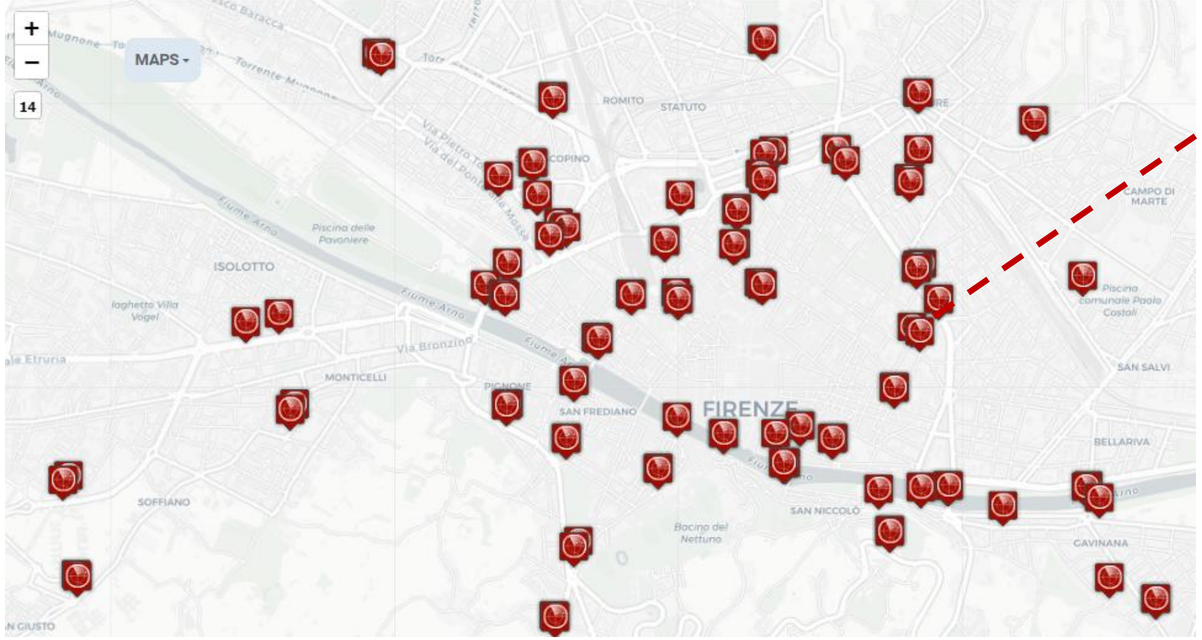
- Starting from fixed traffic sensors scattered in the city, our scope is the prediction/reconstruction of the real-time vehicular traffic density in the whole urban network.

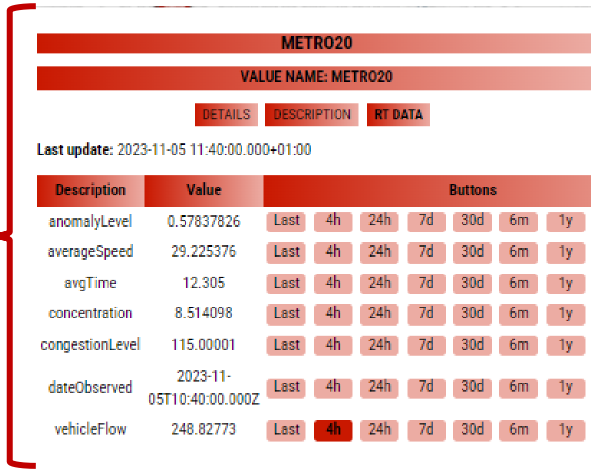


https://www.snap4city.org/dashboardSmartCity/view/Gea.php?iddasboard=MzQ4OA==#

Goal

- Starting from fixed traffic sensors scattered in the city, our scope is the prediction/reconstruction of the real-time vehicular traffic density in the whole urban network.





METRO20

VALUE NAME: METRO20

DETAILS DESCRIPTION RT DATA

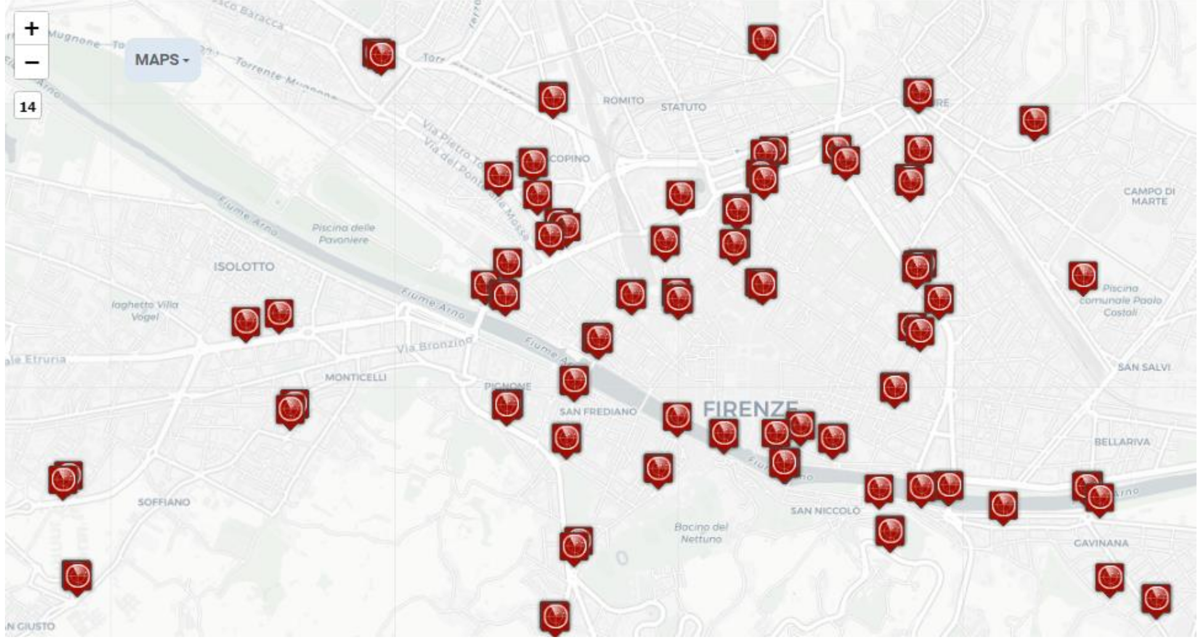
Last update: 2023-11-05 11:40:00.000+01:00

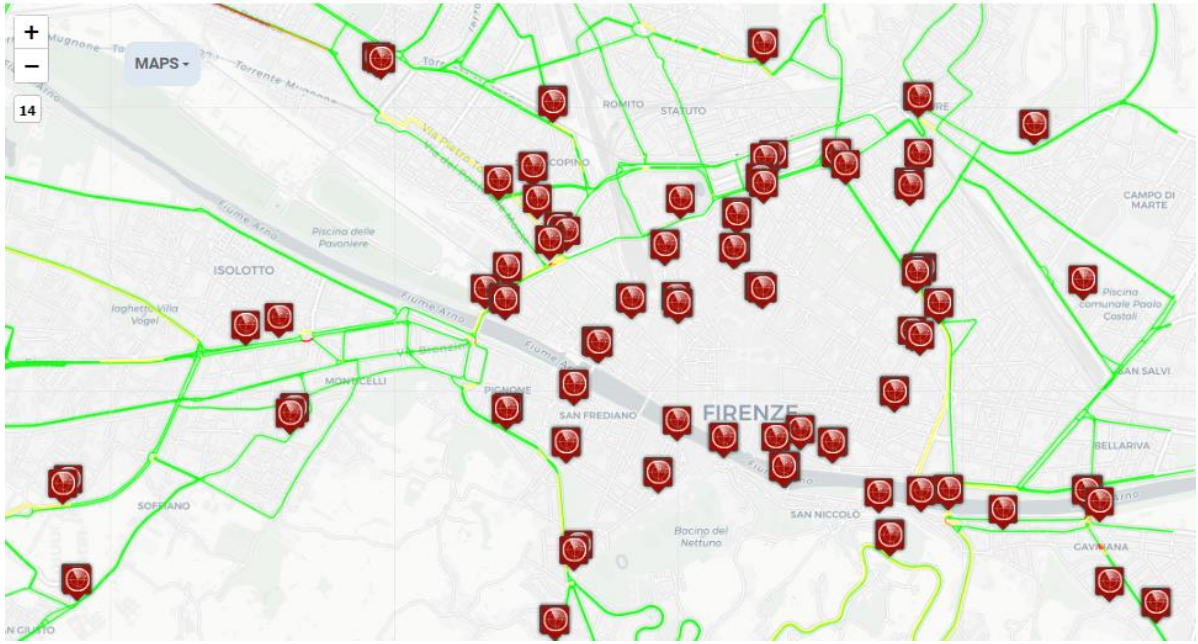
Description	Value	Buttons						
anomalyLevel	0.57837826	Last	4h	24h	7d	30d	6m	1y
averageSpeed	29.225376	Last	4h	24h	7d	30d	6m	1y
avgTime	12.305	Last	4h	24h	7d	30d	6m	1y
concentration	8.514098	Last	4h	24h	7d	30d	6m	1y
congestionLevel	115.00001	Last	4h	24h	7d	30d	6m	1y
dateObserved	2023-11-05T10:40:00.000Z	Last	4h	24h	7d	30d	6m	1y
vehicleFlow	248.82773	Last	4h	24h	7d	30d	6m	1y

https://www.snap4city.org/dashboardSmartCity/view/Gea.php?iddasboard=MzQ4OA==#

Goal

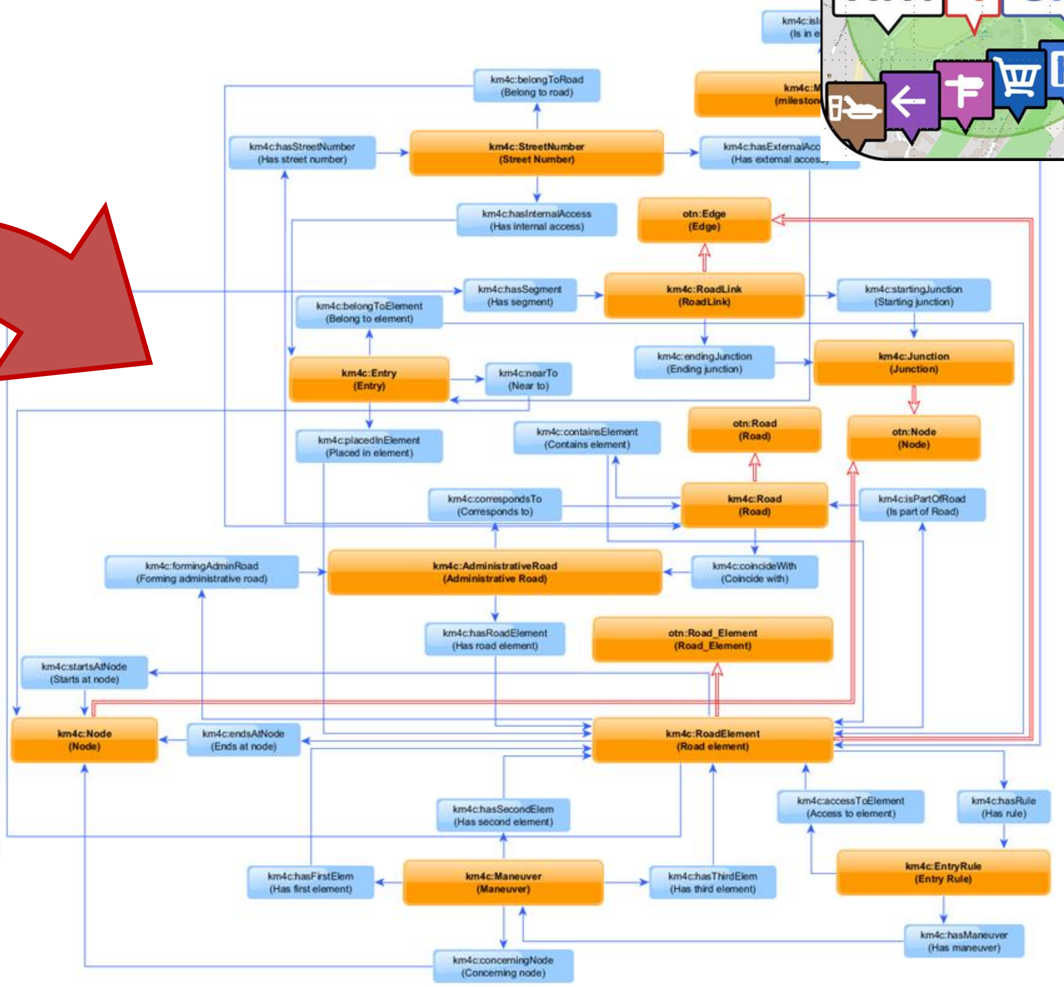
- Starting from fixed traffic sensors scattered in the city, our scope is the prediction/reconstruction of the real-time vehicular traffic density in the whole urban network.



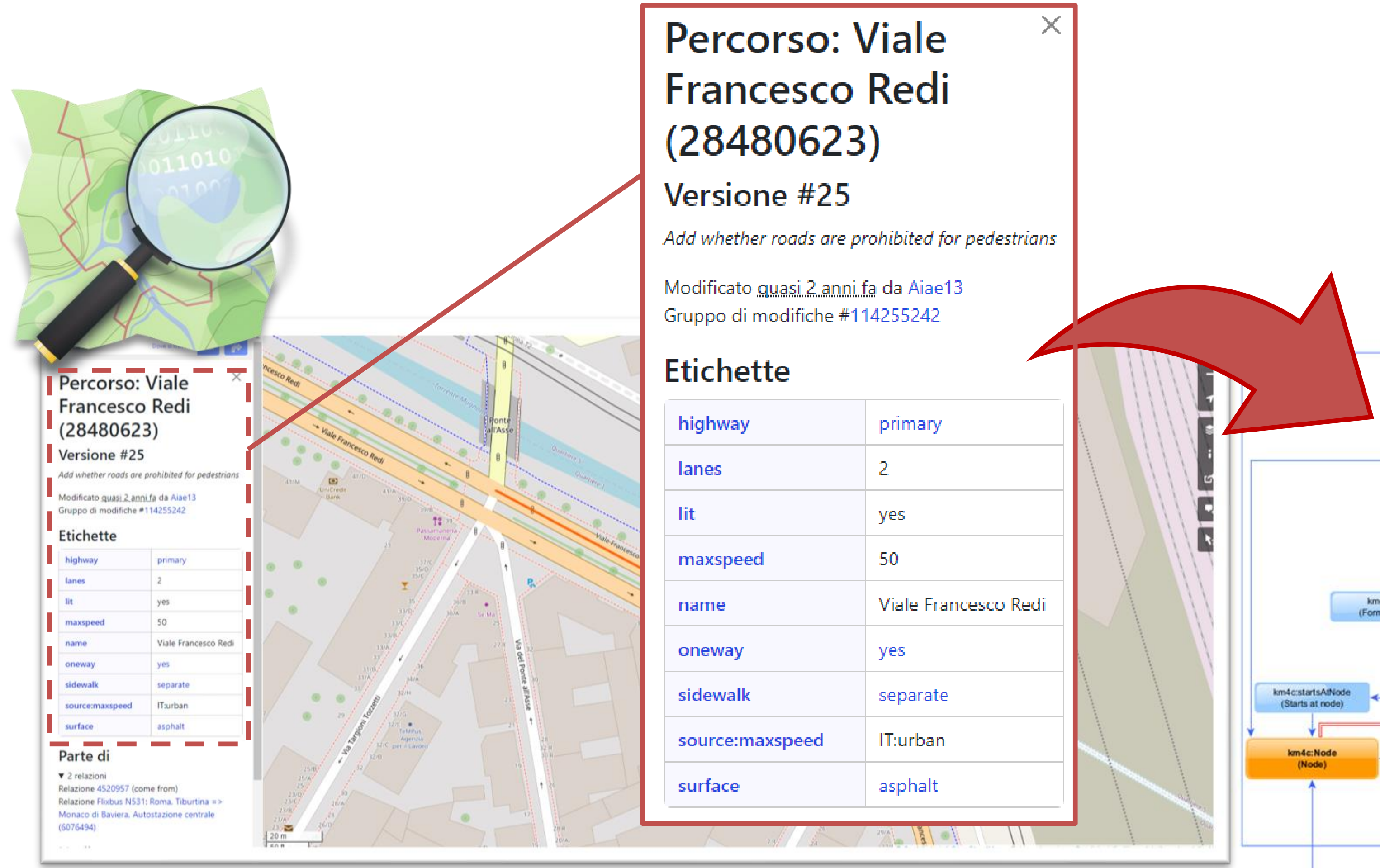


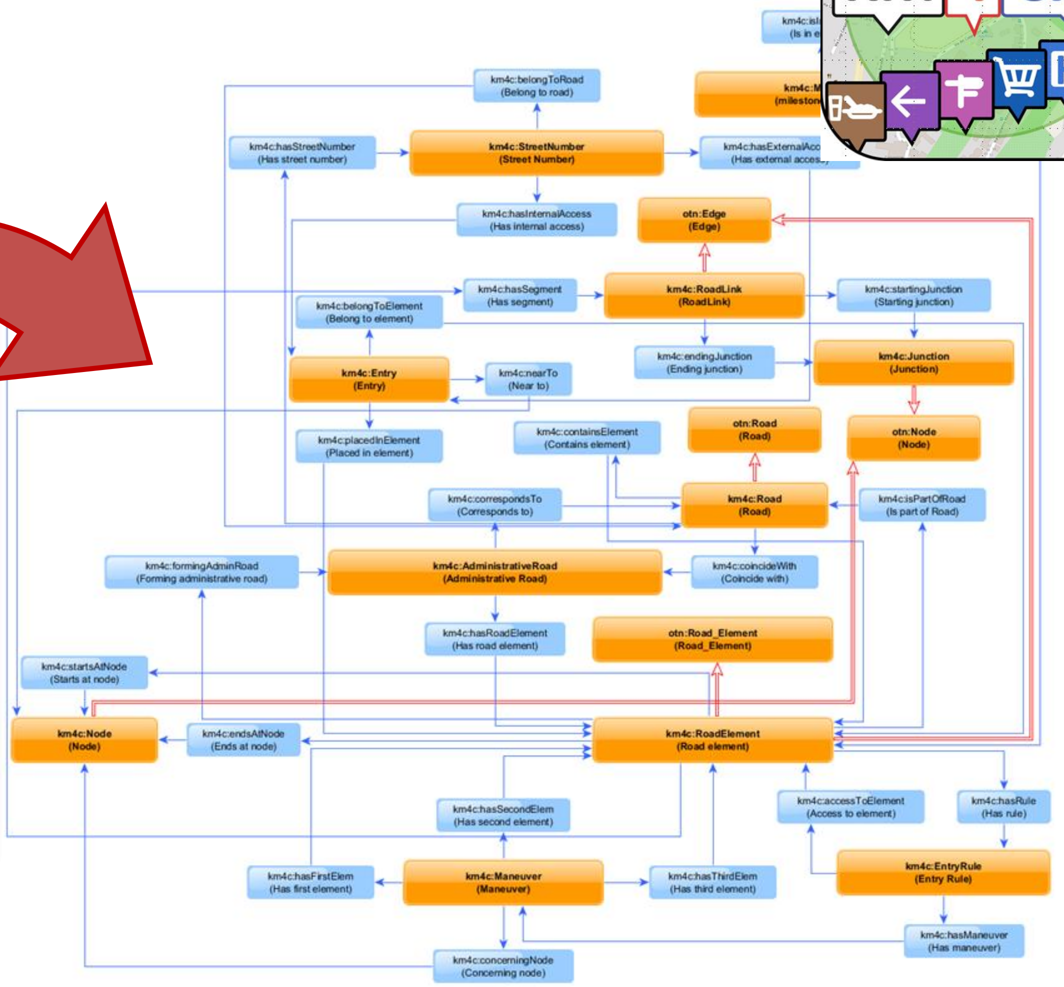
https://www.snap4city.org/dashboardSmartCity/view/Gea.php?iddasboard=MzQ4OA==#

OSM to KM4City



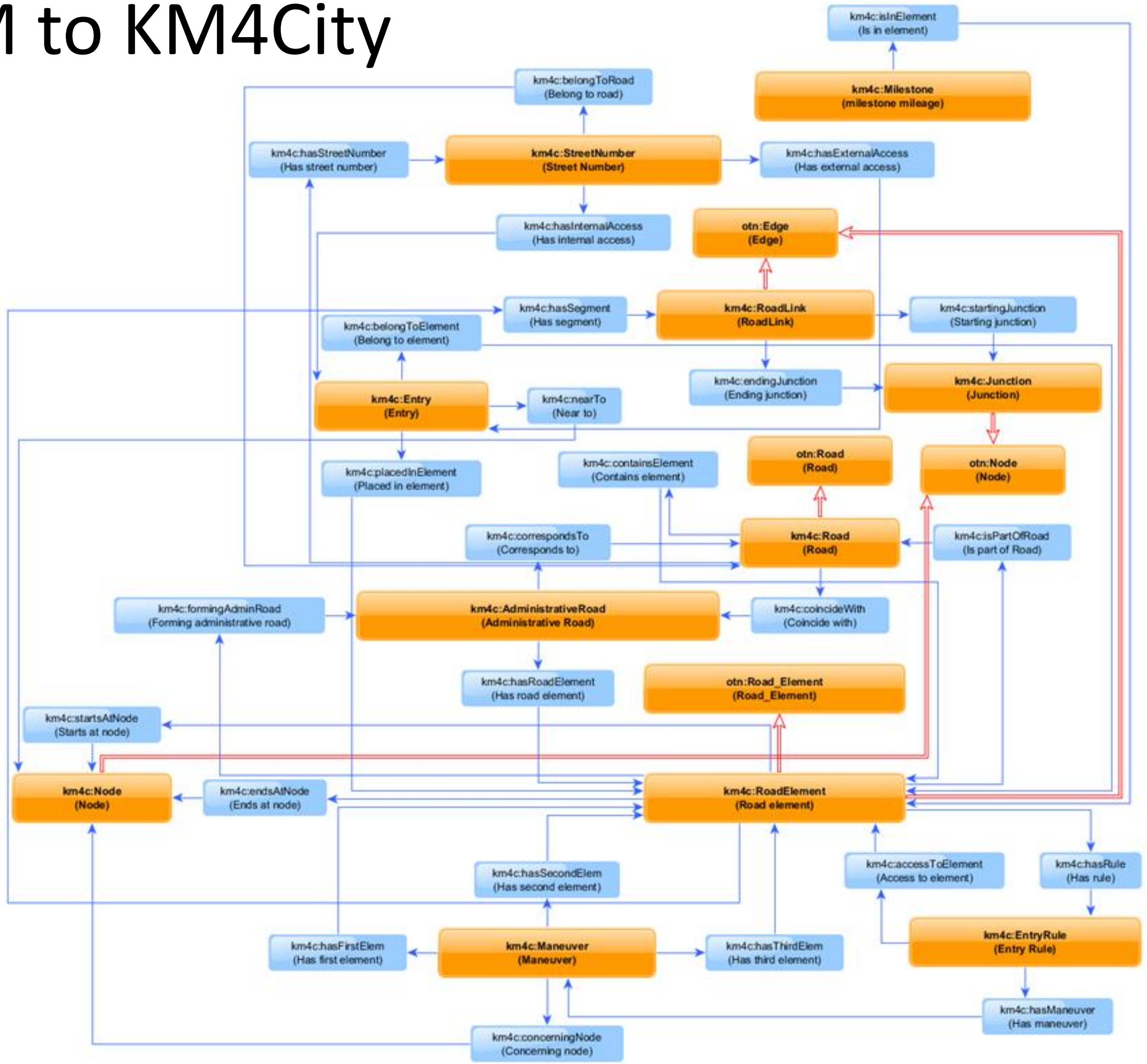
OSM to KM4City





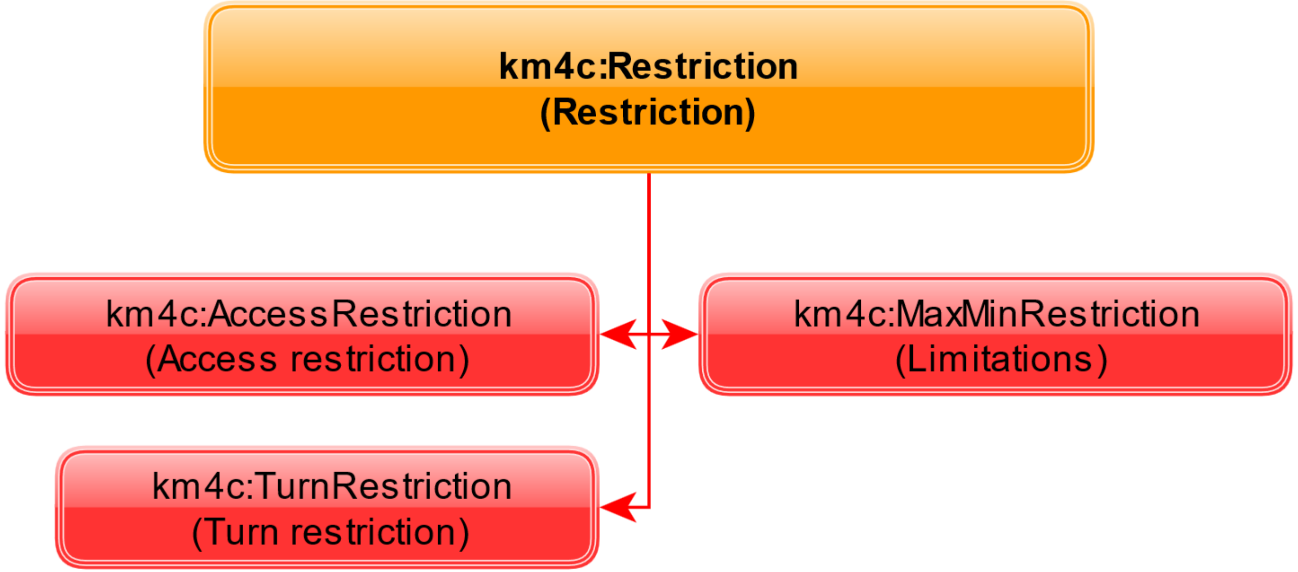
OSM to KM4City

- Road
- AdministrativeRoad
- RoadElement
- Node
- StreetNumber
- Entry
- Lanes/Lane
- Restriction
- …



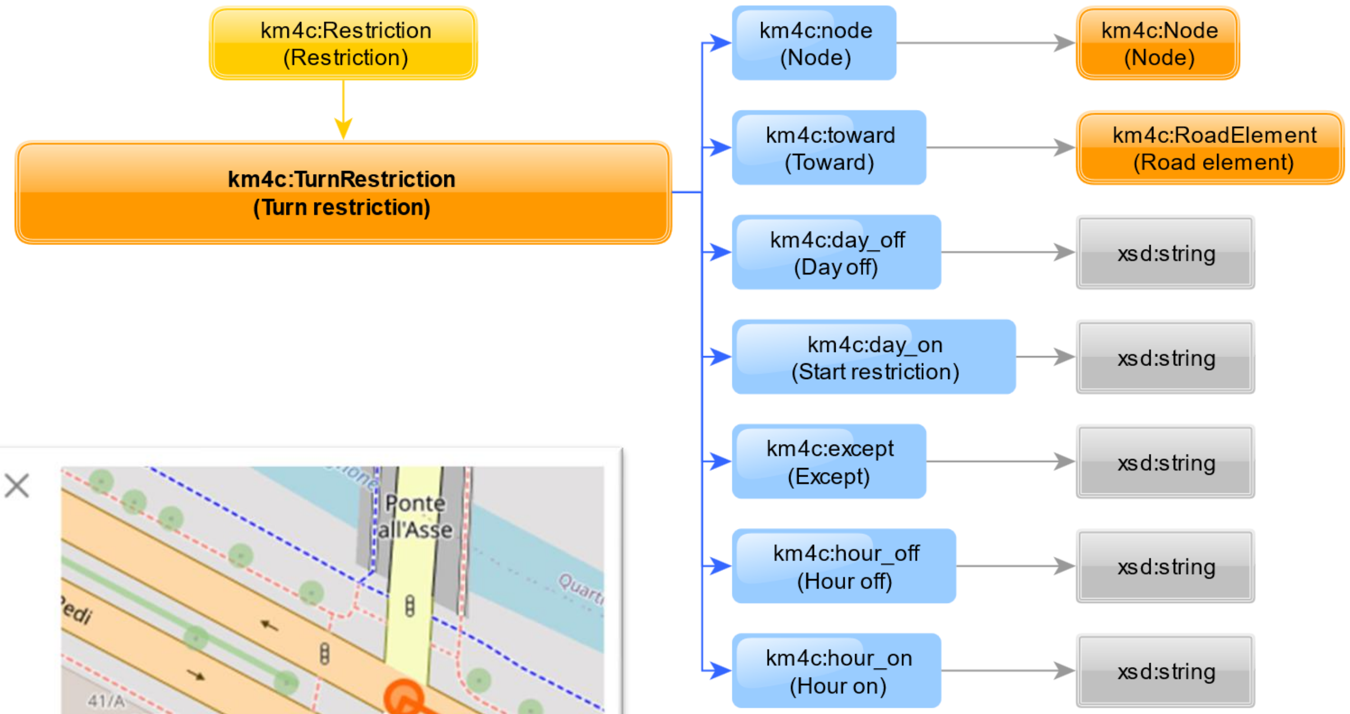
OSM to KM4City

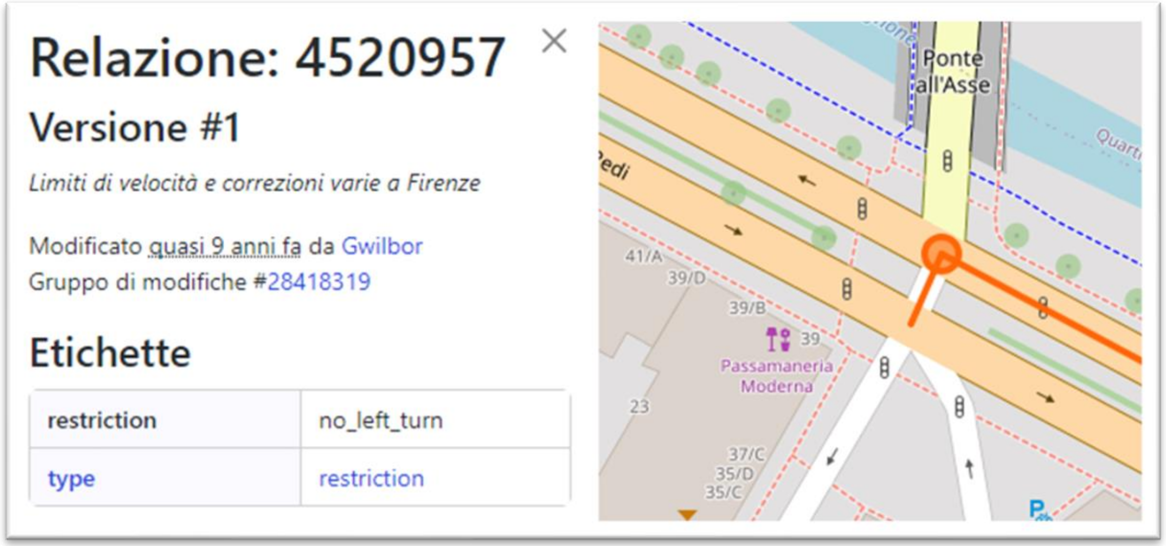
- Road
- AdministrativeRoad
- RoadElement
- Node
- StreetNumber
- Entry
- Lanes/Lane
- **Restriction**
- …



OSM to KM4City

- Road
- AdministrativeRoad
- RoadElement
- Node
- StreetNumber
- Entry
- Lanes/Lane
- **Restriction**
- …





SPARQL Query

- The road graph can be queried from KM4City KB using SPARQL

```
PREFIX km4c: <http://www.disit.org/km4city/schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdfsn: <http://www.w3.org/2003/01/geo/wgs84_pos#>
PREFIX dct: <http://purl.org/dc/terms/>
SELECT ?strada ?nomeStrada ?elementostradale ?highwaytype ?startlat ?startlong ?endlat ?endlong
?compositiontipo ?operatingstatus ?latrafficDir ?lalunghezza WHERE {
  ?strada a km4c:Road.
  ?strada km4c:extendName ?nomeStrada.
  ?strada km4c:inMunicipalityOf ?municip.
  ?municip foaf:name "Firenze".
  ?strada km4c:containsElement ?elementostradale.
  ?elementostradale km4c:startsAtNode ?startnode.
  ?elementostradale km4c:highwayType ?highwaytype.
  ?elementostradale km4c:composition ?compositiontipo.
  ?elementostradale km4c:operatingStatus ?operatingstatus.
  ?elementostradale km4c:trafficDir ?latrafficDir.
  ?elementostradale km4c:length ?lalunghezza.
  ?startnode rdfsn:lat ?startlat.
  ?startnode rdfsn:long ?startlong.
  ?elementostradale km4c:endsAtNode
  ?endnode. ?endnode rdfsn:lat ?endlat.
  ?endnode rdfsn:long ?endlong.
}
```

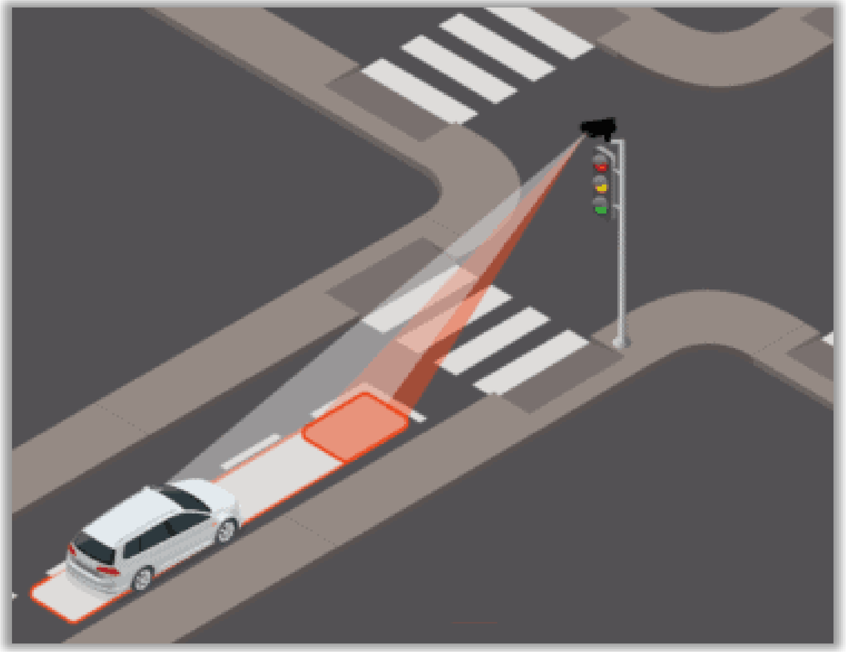
Query Results

Visual Results Mode

#	strada	nomeStrada	elementostradale	highwaytype	startlat	startlong	endlat	endlong	compositiontipo	operatingstatus	latrafficDir	lalunghezza
1	http://www.disit.org/km4city/resource/OS00509333832SR	Via Lambertesca	http://www.disit.org/km4city/resource/OS00509333832RE/0	pedestrian	43.7685	11.2555	43.7685	11.2555	carreggiata unica	in esercizio	tratto stradale aperto in entrambe le direzioni (default)	1.0e0
2	http://www.disit.org/km4city/resource/OS00509333834SR	Piazzale degli Uffizi	http://www.disit.org/km4city/resource/OS00509333834RE/0	pedestrian	43.7679	11.2553	43.7679	11.2553	carreggiata unica	in esercizio	tratto stradale aperto in entrambe le direzioni (default)	1.0e0
3	http://www.disit.org/km4city/resource/OS00524781006SR	Via del Fiordaliso	http://www.disit.org/km4city/resource/OS00524781006RE/0	pedestrian	43.7699	11.2526	43.7699	11.2526	carreggiata unica	in esercizio	tratto stradale aperto in entrambe le direzioni (default)	1.0e0
4	http://www.disit.org/km4city/resource/OS00507103663SR	Viale Alessandro Guidoni	http://www.disit.org/km4city/resource/OS00507103663RE/4	unclassified	43.7976	11.2182	43.7976	11.2183	carreggiata unica	in esercizio	tratto stradale aperto nella direzione positiva (da giunzione NOD_INI a giunzione NOD_FIN)	10.0e0

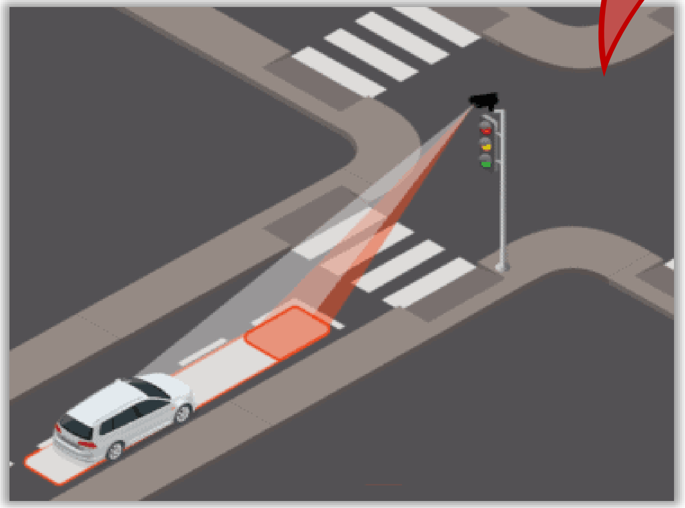
Sensors and detections

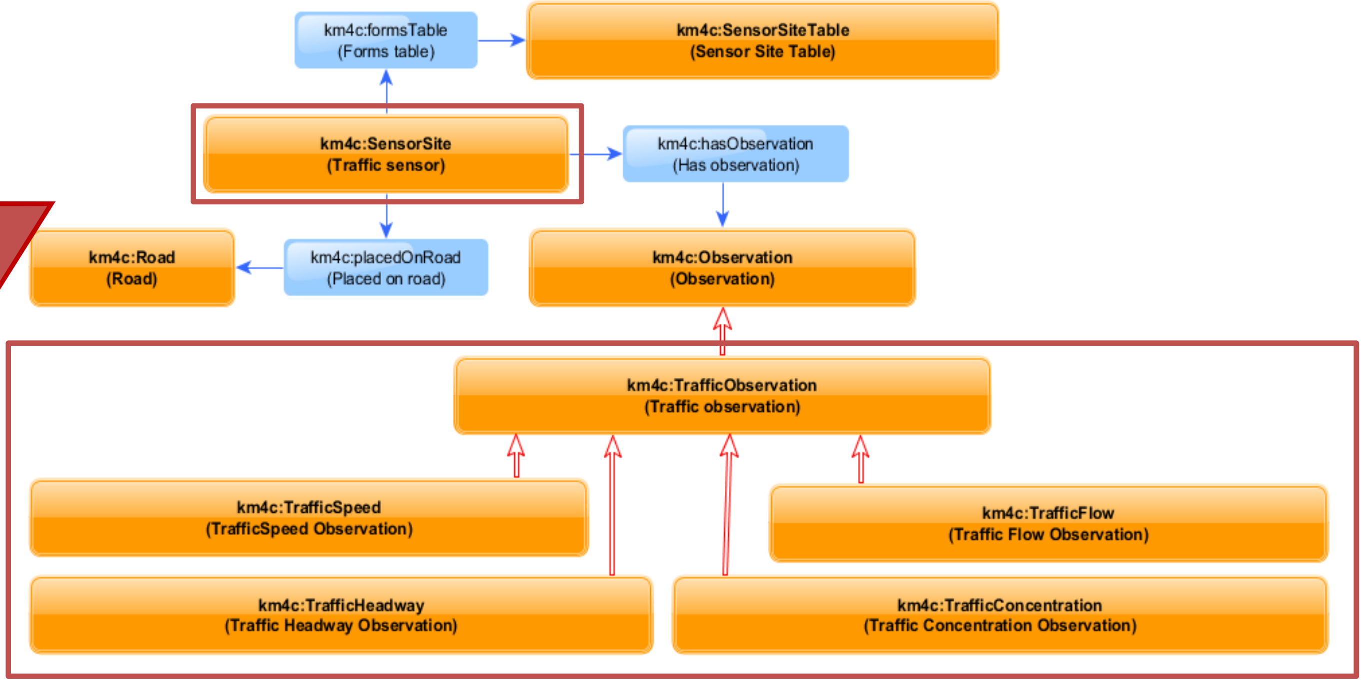
- The traffic sensors in a municipality (e.g., spire road sensors and cameras) give the state of the traffic **counting the number of vehicles** which pass through the supervised area
- Traffic Sensors come from Open Data and have
  – Static information
    - identifier,
    - geolocation,
    - street address,
    - technical specifications
    - …
  – Real-time traffic flow detections
    - timestamp,
    - detected traffic flow,
    - estimated speed
    - …



Sensors to KM4City

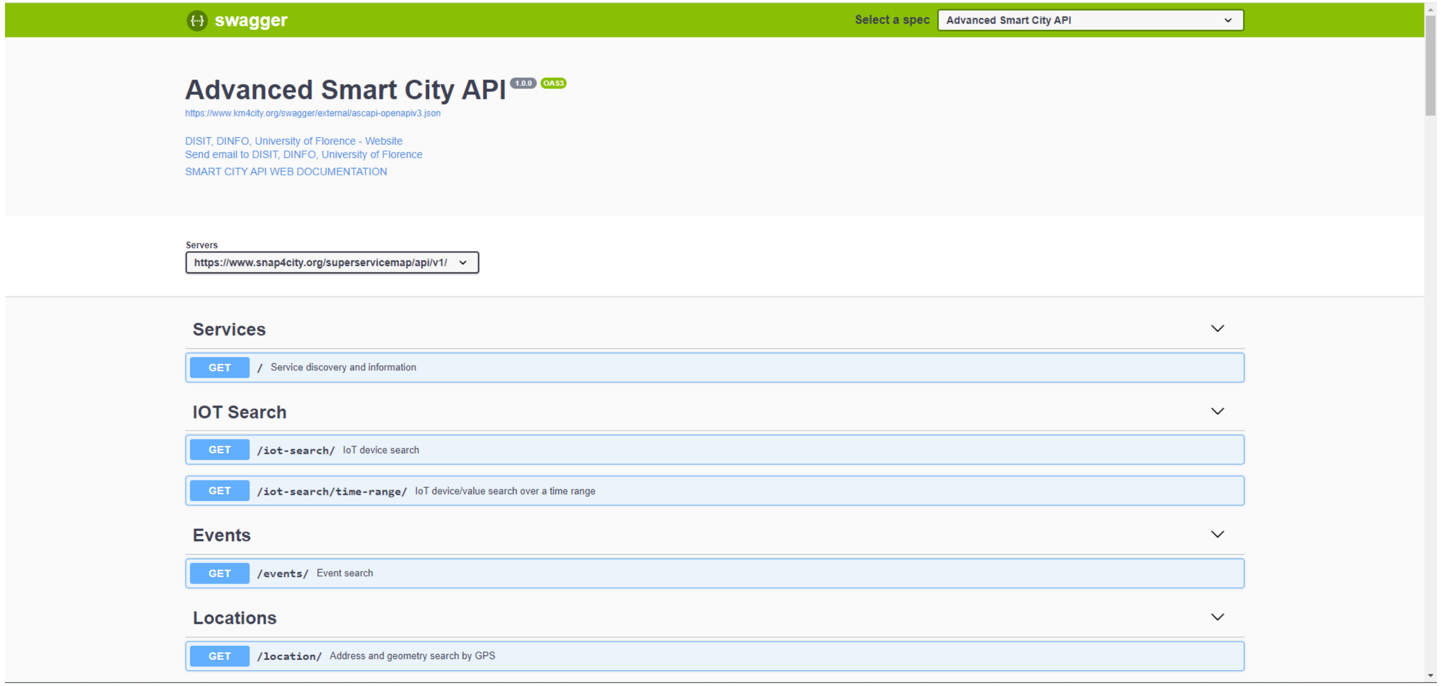
- SensorSite
- TrafficObservation
  - TrafficSpeed
  - TrafficFlow
  - TrafficHeadway
  - TrafficConcentration





Smart City API

- The traffic reconstruction model implementation accesses traffic data through **dedicated APIs**
- Traffic flows are read every **10 minutes**, the refresh frequency of the traffic sensors.



https://www.snap4city.org/drupal/node/20

Mathematical model

- The traffic sensor detections are interpreted as **sources of traffic** leading into the outcoming roads of the nodes where sensors are located.

- We consider a mathematical model for **fluid dynamic flows** on networks which is based on conservation laws.

- Road network is studied as a **directed graph** composed by arcs that meet at some nodes, corresponding to junctions.

Mathematical model

- Roads are modelled as if they were water pipelines.
- Crossroads are modelled as if they were pipeline junctions.
- The flow of the vehicles is modelled as if it was a water flow.
- The **law of conservation of the flow (of the vehicles)** applies:

$$\frac{\partial \rho(t,x)}{\partial t} + \frac{\partial f\big(\rho(t,x)\big)}{\partial x} = 0$$

where
– $\rho(t,x)$ is the vehicular density,
– $f(\rho(t,x)) = \rho(t,x)v(t,x)$ is the vehicular flux, and
– $v(t,x)$ is the local speed of the vehicles.

Mathematical model

- Roads are modelled as if they were water pipelines.
- Crossroads are modelled as if they were pipeline junctions.
- The flow of the vehicles is modelled as if it was a water flow.
- The **law of conservation of the flow (of the vehicles)** applies:

Partial derivative of the **density** w.r.t. the **time**

$$\frac{\partial \rho(t,x)}{\partial t} + \frac{\partial f\big(\rho(t,x)\big)}{\partial x} = 0$$

Partial derivative of product of the **density** and the **velocity** w.r.t. the **space**

where
– $\rho(t,x)$ is the vehicular density,
– $f(\rho(t,x)) = \rho(t,x)v(t,x)$ is the vehicular flux, and
– $v(t,x)$ is the local speed of the vehicles.

Mathematical model

- Roads are modelled as if they were water pipelines.
- Crossroads are modelled as if they were pipeline junctions.
- The flow of the vehicles is modelled as if it was a water flow.
- The **law of conservation of the flow (of the vehicles)** applies:

Partial derivative of the **density** w.r.t. the **time**

$$\frac{\partial \rho(t,x)}{\partial t} + \frac{\partial f\big(\rho(t,x)\big)}{\partial x} = 0$$

Partial derivative of product of the **density** and the **velocity** w.r.t. the **space**

$$\rho(t,a) = \rho_a(t) \text{ and } \rho(t,b) = \rho_b(t)$$

$$\rho(0,x) = \rho_0(x).$$

Mathematical model

- $\rho(t, x)$ denotes the car density which admits values from 0 to $\rho_{\max}$, where $\rho_{\max} > 0$ is the **maximal vehicular density** on the road.

- The function $f(\rho(t, x))$ is the **vehicular flux** which is defined as the product $\rho(t, x)v(t, x)$, where $v(t, x)$ is the **local speed** of the cars.

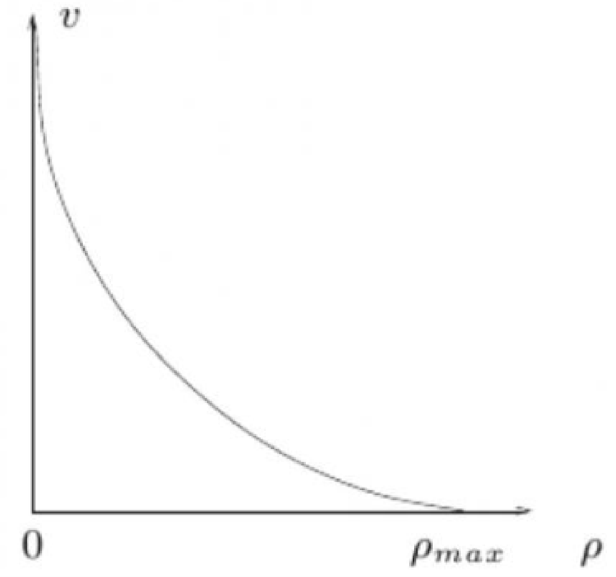
- If we assume that $v(t, x)$ is a decreasing function, only depending on the density, then the corresponding **flux is a concave function**

Fundamental diagram

We consider the local cars' speed as

$$v(\rho) = v_{max}(1 - \frac{\rho}{\rho_{max}})$$

$$\begin{cases} \rho = 0 \Rightarrow v(\rho) = v_{max} \\ \rho^{\uparrow} \Rightarrow v(\rho)^{\downarrow} \\ \rho = \rho_{max} \Rightarrow v(\rho) = 0 \end{cases}$$



Fundamental diagram

We consider the local cars' speed as

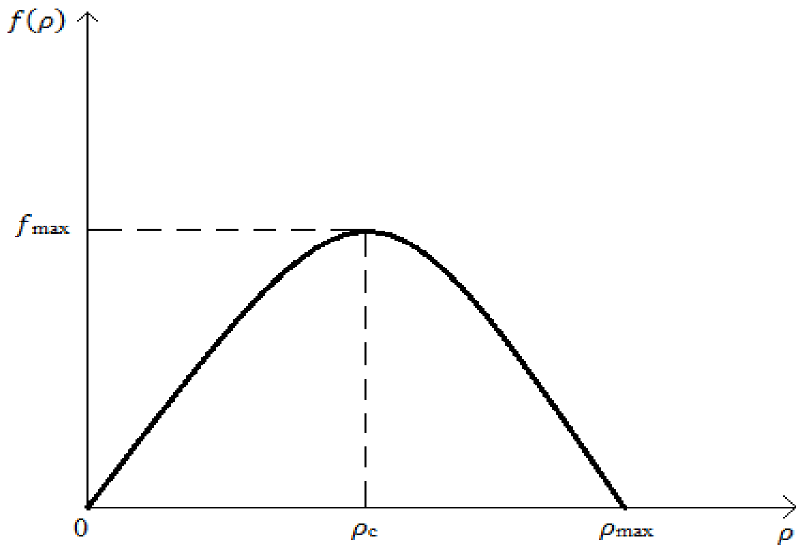
$$v(\rho) = v_{max}(1 - \frac{\rho}{\rho_{max}})$$

$$\begin{cases} \rho = 0 \Rightarrow v(\rho) = v_{max} \\ \rho^{\uparrow} \Rightarrow v(\rho)^{\downarrow} \\ \rho = \rho_{max} \Rightarrow v(\rho) = 0 \end{cases}$$

obtaining that

$$f(\rho) = v_{max}\left(1 - \frac{\rho}{\rho_{max}}\right)\rho,$$

where v_{max} is the limit speed



Fundamental diagram

We consider the local cars' speed as

$$v(\rho) = v_{max}(1 - \frac{\rho}{\rho_{max}})$$

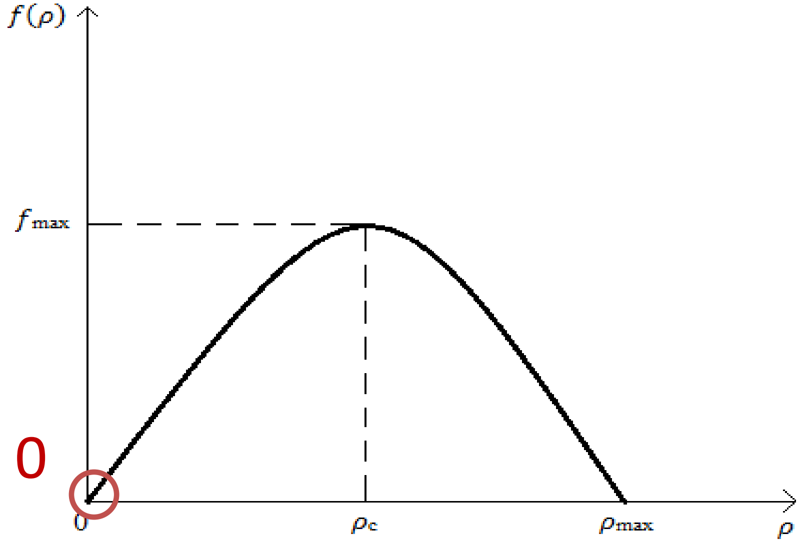
$$\begin{cases} \rho = 0 \Rightarrow v(\rho) = v_{max} \\ \rho^{\uparrow} \Rightarrow v(\rho)^{\downarrow} \\ \rho = \rho_{max} \Rightarrow v(\rho) = 0 \end{cases}$$

obtaining that

$$f(\rho) = v_{max}\left(1 - \frac{\rho}{\rho_{max}}\right)\rho,$$

where v_{max} is the limit speed

$\rho = 0 \ \Rightarrow f(\rho) = 0$



Fundamental diagram

We consider the local cars’ speed as

$$v(\rho) = v_{max}\left(1 - \frac{\rho}{\rho_{max}}\right)$$

$$\begin{cases} \rho = 0 \Rightarrow v(\rho) = v_{max} \\ \rho^{\uparrow} \Rightarrow v(\rho)^{\downarrow} \\ \rho = \rho_{max} \Rightarrow v(\rho) = 0 \end{cases}$$

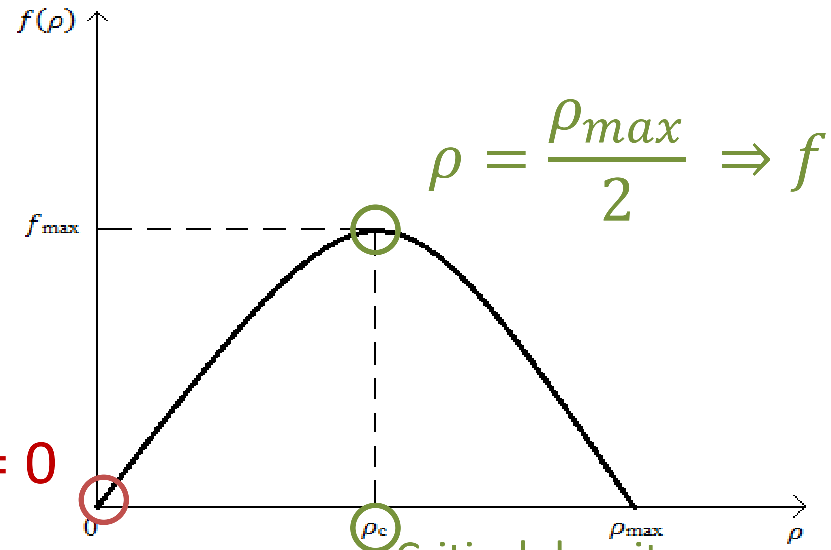
obtaining that

$$f(\rho) = v_{max}\left(1 - \frac{\rho}{\rho_{max}}\right)\rho,$$

where v_{max} is the limit speed

$$\rho = \frac{\rho_{max}}{2} \Rightarrow f(\rho) = f_{max}$$

$$\rho = 0 \ \Rightarrow f(\rho) = 0$$



Critical density

Fundamental diagram

We consider the local cars' speed as

$$v(\rho) = v_{max}(1 - \frac{\rho}{\rho_{max}})$$

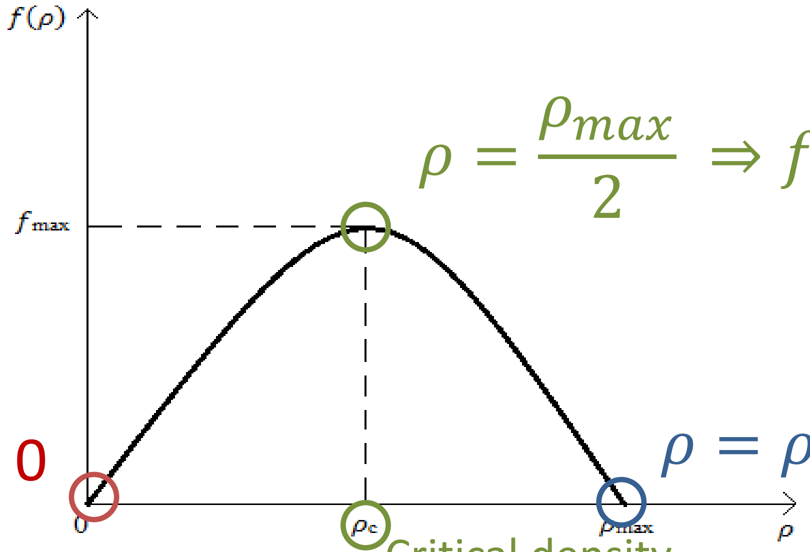
$$\begin{cases} \rho = 0 \Rightarrow v(\rho) = v_{max} \\ \rho^{\uparrow} \Rightarrow v(\rho)^{\downarrow} \\ \rho = \rho_{max} \Rightarrow v(\rho) = 0 \end{cases}$$

obtaining that

$$f(\rho) = v_{max}\left(1 - \frac{\rho}{\rho_{max}}\right)\rho,$$

where v_{max} is the limit speed

$$\rho = \frac{\rho_{max}}{2} \Rightarrow f(\rho) = f_{max}$$



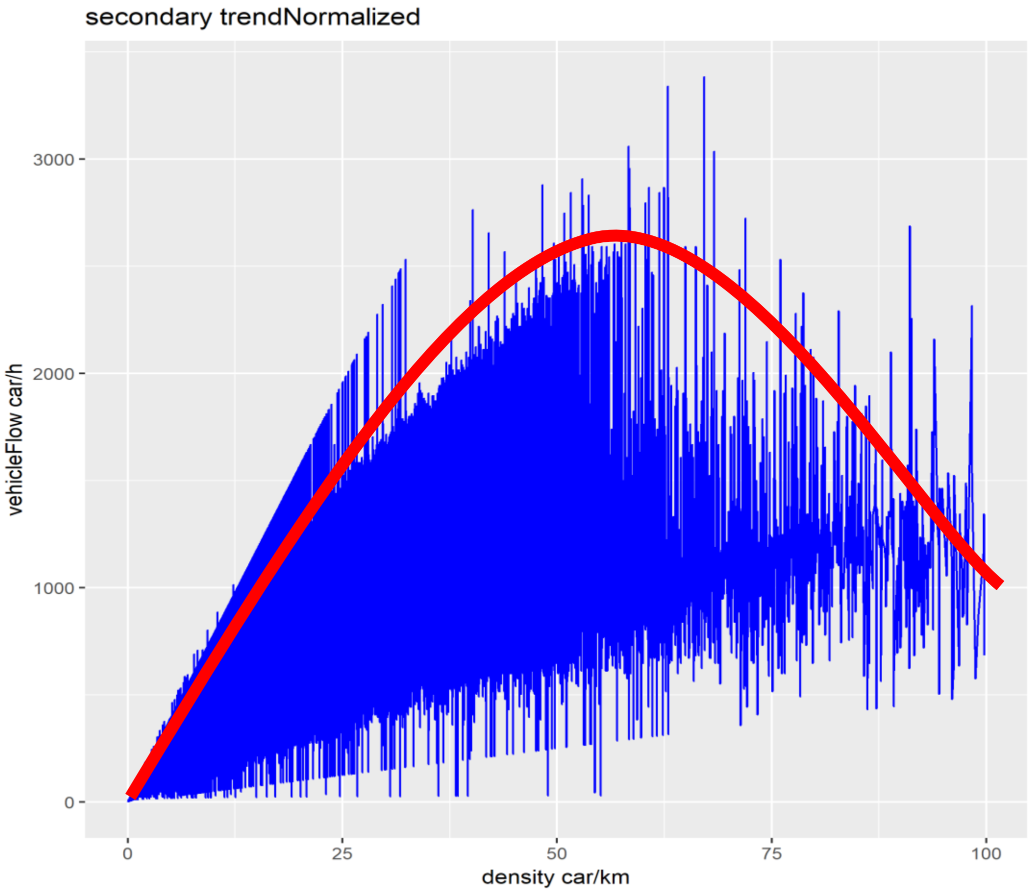
$$\rho = 0 \ \Rightarrow f(\rho) = 0$$

$$\rho = \rho_{max} \Rightarrow f(\rho) = 0$$

Critical density

Fundamental diagram: data analysis

- Observation from traffic sensors (Florence, year 2019)



Discretization scheme

The following discretization and simplification of the model is operated:

- Each road is partitioned in segments Δ*x* long.
- The time is partitioned in intervals Δ*t* long.
- Denote (h,m) a bounded time-space region (cell) of duration h and length m.
- Let $u_m^h = u(t_h, x_m) = u(h\Delta t, m\Delta x)$ be a continuous function defined on (h,m).
- Denote *F* the numerical flux. Then, the vehicular density results from:

$$u_m^{h+1} = u_m^h - \frac{\Delta t}{\Delta x}\left(F\left(u_m^h, u_{m+1}^h\right) - F\left(u_{m-1}^h, u_m^h\right)\right)$$

Discretization scheme

The following discretization and simplification of the model is operated:

- Each road is partitioned in segments Δ*x* long.
- The time is partitioned in intervals Δ*t* long.
- Denote (h,m) a bounded time-space region (cell) of duration h and length m.
- Let $u_m^h = u(t_h, x_m) = u(h\Delta t, m\Delta x)$ be a continuous function defined on (h,m).
- Denote *F* the numerical flux. Then, the vehicular density results from:

$$u_m^{h+1} = u_m^h - \frac{\Delta t}{\Delta x}\left(F\left(u_m^h, u_{m+1}^h\right) - F\left(u_{m-1}^h, u_m^h\right)\right)$$

Density at time *h+1* in segment *m*

Density at time *h* in segment *m*

Changes due to neighborhood segments

Discretization scheme

where F denotes the flux, which is computed by taking into account the physical constraints of the selected road and its connections to the neighbor road-segments according to:

$$F(w,z) = \begin{cases} \min\left(f(w), f(z)\right), & w \leq z \\ f(w), & z < w < \rho_c \\ f_{max}, & z < \rho_c < w \\ f(z), & \rho_c < z < w \end{cases}$$

Sensors' measurements

- The measured data sensor is interpreted as the **source of traffic** leading into the outcoming roads of the considered junction.
- Suppose to assign a condition at the incoming boundary for $x = 0$ as $\rho(t, 0) = \rho_b^{inc}(t)$.
- We proceed by inserting an **incoming ghost cell** and the discretization becomes

$$u_0^{h+1} = u_0^h - \frac{\Delta t}{\Delta x}\Big(F\big(u_0^h, u_1^h\big) - F\big(v_{(inc)}^h, u_0^h\big)\Big)$$

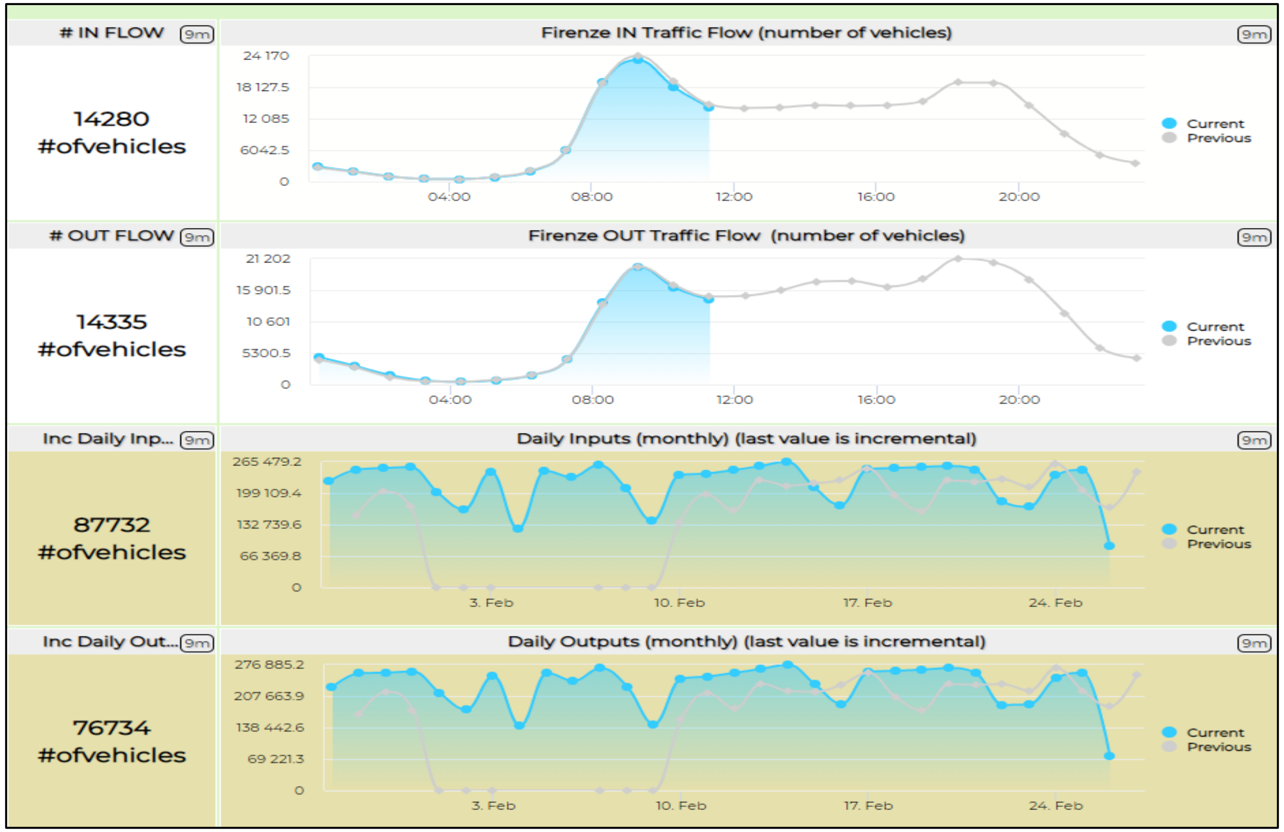
where

$$v_{(inc)}^h = \frac{1}{\Delta t}\int_{t_h}^{t_h+1} \rho_b^{inc}(t)\, dt$$

replaces the ghost value u_{-1}^h

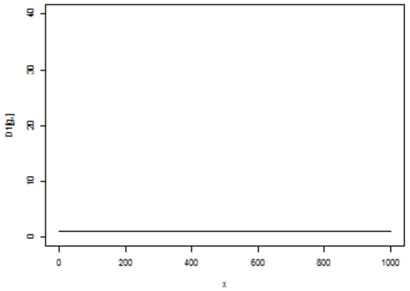
Flow conservation

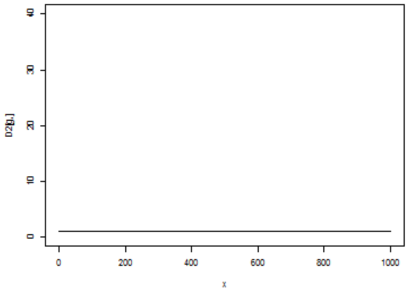
- The assessment has been performed verifying in real-time the conservation of the flow in the area. Figure reports the real-time dashboard for controlling the conservation of flow

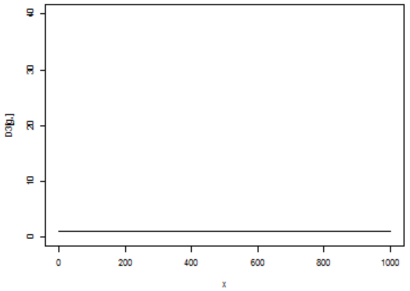


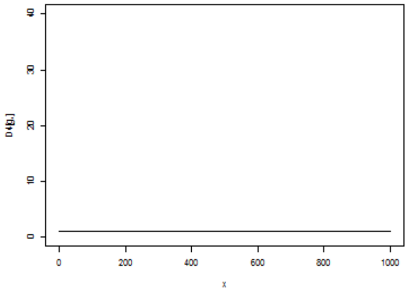
https://www.snap4city.org/dashboardSmartCity/view/index.php?iddasboard=MTc2MQ==

Phisical Principle of Narrowing in roads







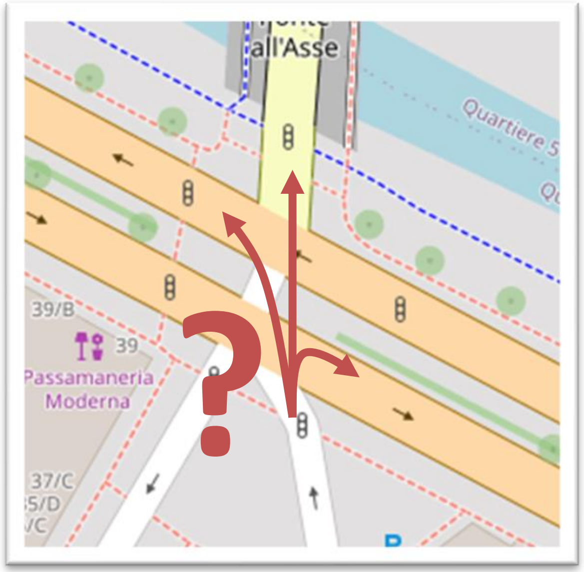


Numerical meaning

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]	[,9]	[,10]	[,11]
[1,]	9.343250	8.938677	8.563212	8.201665	7.848900	7.502591	7.161543	11.50547	19.58907	20.61252	20.74469
[2,]	9.346269	8.943543	8.569777	8.209852	7.858653	7.513860	7.174282	12.09272	19.73380	20.63228	20.75818
[3,]	9.349261	8.948364	8.576283	8.217966	7.868318	7.525029	7.186909	12.70006	19.86189	20.65113	20.77141
[4,]	9.352225	8.953141	8.582730	8.226007	7.877897	7.536099	7.199425	13.32546	19.97522	20.66917	20.78439
[5,]	9.355162	8.957875	8.589120	8.233976	7.887391	7.547071	7.234340	13.94458	20.07550	20.68647	20.79712
[6,]	9.358073	8.962567	8.595452	8.241873	7.896801	7.557947	7.301140	14.54628	20.16430	20.70311	20.80961
[7,]	9.360957	8.967216	8.601727	8.249701	7.906128	7.568729	7.404075	15.12479	20.24300	20.71915	20.82187
[8,]	9.363816	8.971824	8.607947	8.257460	7.915373	7.579416	7.546541	15.67538	20.31287	20.73464	20.83391
[9,]	9.366649	8.976391	8.614111	8.265150	7.924538	7.590011	7.730996	16.19439	20.37501	20.74963	20.84572
[10,]	9.369457	8.980917	8.620222	8.272773	7.933623	7.600514	7.958933	16.67929	20.43040	20.76415	20.85733
[11,]	9.372240	8.985404	8.626279	8.280330	7.942629	7.610928	8.230920	17.12855	20.47992	20.77825	20.86873
[12,]	9.374999	8.989851	8.632282	8.287821	7.951557	7.621252	8.546680	17.54164	20.52431	20.79196	20.87993
[13,]	9.377733	8.994260	8.638234	8.295247	7.960409	7.631488	8.905205	17.91883	20.56425	20.80531	20.89094
[14,]	9.380444	8.998630	8.644134	8.302610	7.969186	7.641638	9.304891	18.26110	20.60031	20.81832	20.90176
[15,]	9.383131	9.002962	8.649984	8.309909	7.977887	7.651701	9.743678	18.56995	20.63300	20.83100	20.91240
[16,]	9.385794	9.007257	8.655783	8.317146	7.986515	7.661680	10.219185	18.84726	20.66275	20.84339	20.92286
[17,]	9.388435	9.011515	8.661533	8.324322	7.995070	7.671576	10.728829	19.09520	20.68994	20.85550	20.93315
[18,]	9.391053	9.015736	8.667233	8.331437	8.003553	7.681388	11.269937	19.31607	20.71491	20.86734	20.94328
[19,]	9.393649	9.019922	8.672886	8.338492	8.011965	7.691119	11.839829	19.51221	20.73794	20.87894	20.95324
[20,]	9.396222	9.024072	8.678491	8.345487	8.020306	7.700770	12.435887	19.68596	20.75928	20.89029	20.96304
[21,]	9.398774	9.028188	8.684048	8.352424	8.028579	7.714870	13.051077	19.83957	20.77913	20.90142	20.97269
[22,]	9.401305	9.032268	8.689560	8.359304	8.036783	7.752828	13.663633	19.97518	20.79769	20.91233	20.98219
[23,]	9.403814	9.036315	8.695025	8.366126	8.044919	7.819804	14.266206	20.09478	20.81511	20.92304	20.99155
[24,]	9.406302	9.040327	8.700445	8.372892	8.052989	7.920452	14.852134	20.20021	20.83154	20.93355	21.00076
[25,]	9.408769	9.044307	8.705820	8.379603	8.060992	8.058718	15.415642	20.29316	20.84708	20.94387	21.00984
[26,]	9.411216	9.048254	8.711151	8.386258	8.068931	8.237704	15.951981	20.37513	20.86184	20.95401	21.01878
[27,]	9.413643	9.052168	8.716438	8.392859	8.076805	8.459583	16.457502	20.44748	20.87592	20.96397	21.02758
[28,]	9.416049	9.056050	8.721682	8.399407	8.084615	8.725588	16.929655	20.51143	20.88938	20.97376	21.03626
[29,]	9.418436	9.059900	8.726883	8.405902	8.092363	9.036048	17.366946	20.56805	20.90229	20.98339	21.04482
[30,]	9.420804	9.063719	8.732042	8.412344	8.100048	9.390478	17.768828	20.61828	20.91472	20.99286	21.05325
[31,]	9.423152	9.067507	8.737160	8.418734	8.107672	9.787692	18.135581	20.66297	20.92670	21.00218	21.06157
[32,]	9.425481	9.071265	8.742237	8.425074	8.115236	10.225939	18.468159	20.70282	20.93829	21.01135	21.06976
[33,]	9.427792	9.074992	8.747273	8.431363	8.122740	10.703041	18.768044	20.73849	20.94951	21.02038	21.07785

Application of the model

- The vehicular traffic flow is propagated in the network according to the fluid dynamic model



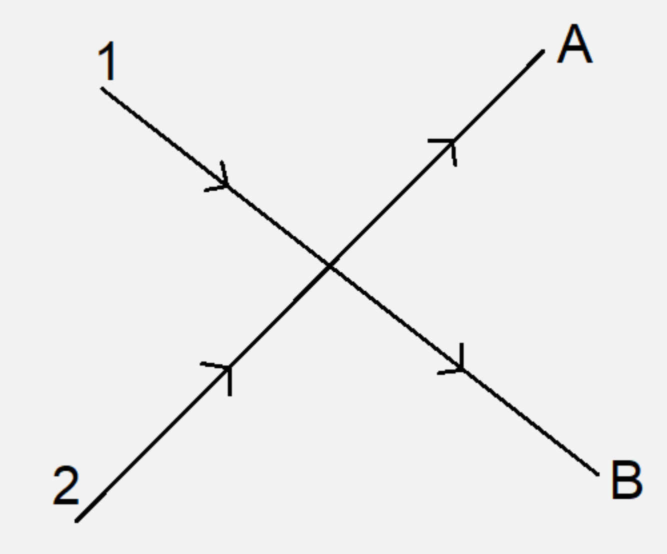
- The distribution of the traffic at crossroads is governed by a **Traffic Distribution Matrix** whose coefficients are based on the **weights** of the segments of roads that make the crossroad.

Traffic Matrix Distribution (TDM)

- The TDM is a distribution matrix describing the percentage of vehicles getting out each outcoming road with respect to those getting in each incoming road.

- Thus, it is defined as $TDM = \{w_{ji}\}_{j=n+1,\dots,n+m, i=1,\dots,n}$ so that $0 < w_{ji} < 1$ and $\sum_{j=n+1}^{n+m} w_{ji} = 1$, for $i = 1, \dots, n$ and $j = n+1, \dots, n+m$, where w_{ji} coefficients (called weights) are the percentages of vehicles arriving from the i-th incoming road and taking the j-th outcoming road (assuming that, on each junction, the incoming flux coincides with the outcoming flux).

Traffic distribution on a junction



$$\begin{pmatrix} d_{1A} & d_{2A} \\ d_{1B} & d_{2B} \end{pmatrix} \begin{pmatrix} f_1 \\ f_2 \end{pmatrix} = \begin{pmatrix} f_A \\ f_B \end{pmatrix}$$

Weight initialization

Weights are **initialized** based on the following:

- **Road type**: motorway, trunk, primary, secondary, tertiary, unclassified, residential, service;
- **Lanes**: how many lanes are drawn on the asphalt, also considering possible restrictions (e.g. lanes reserved to public transport);
- **Traffic restrictions**: examples are mandatory/forbidden directions at crossroads, speed limits, limited traffic zones.

Basic computational approach of TFR

Load road network and details**.**
For each *Time t*:
    Get traffic sensors' values
    Load *TDM(t) and weights*
    For *h: 1 → H*
        Compute the model for each sensor' location to the sensors' value
        Compute the model for all traffic distribution in each junction
        Compute the model for all traffic density in each road-segments
    End For h
    Compute graphics representation
End For each Time

For each time slot t, after H iterations all the road-segments in the road network have an estimated traffic density value.

Stochastic learning of the weights (traffic distribution)

It has been observed that:

- The way how vehicles distribute at crossroads varies depending on the day of the week, and of the time of the day;
- A random variation of some weights is very likely to lead to an improved accuracy;
- If no improvements are achieved after *n* attempts, it is reasonable to move anyway to the best of the last *n* configs.

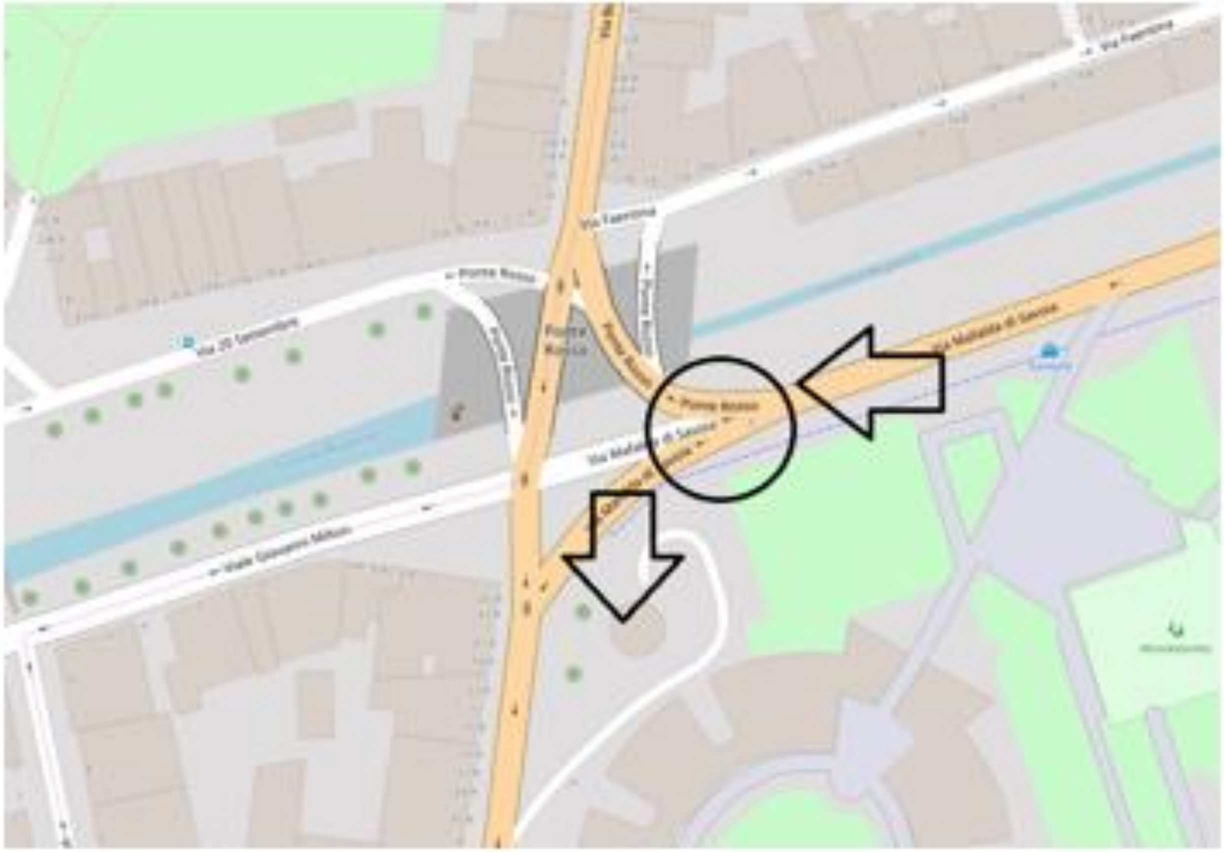
An offline process is run, based on the above, that leads to (an optimal) time-based weights assignment, aimed at an improved accuracy.

Junction distribution learning

The fork of via Mafalda di Savoia (East), in via Mafalda di Savoia (South), Viale Giovanni Milton (West) and Via del Ponte Rosso (North), in Florence.

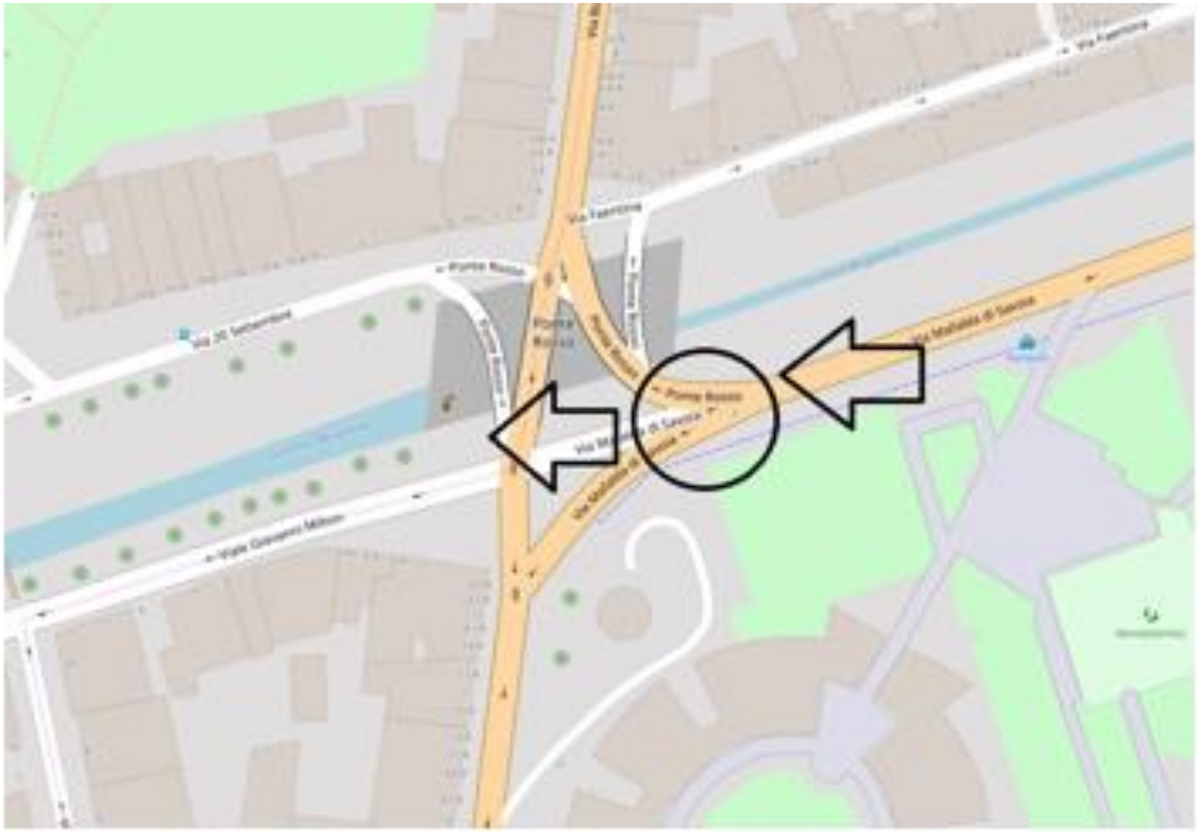
Junction distribution learning

Road Type: primary
Lanes: 2
Designated Lanes: 0
Restrictions: none
Learning Factor: 61
Elem. Type: T.O.C.
Length: 63
Direction: positive
...
Weight: 31.122%



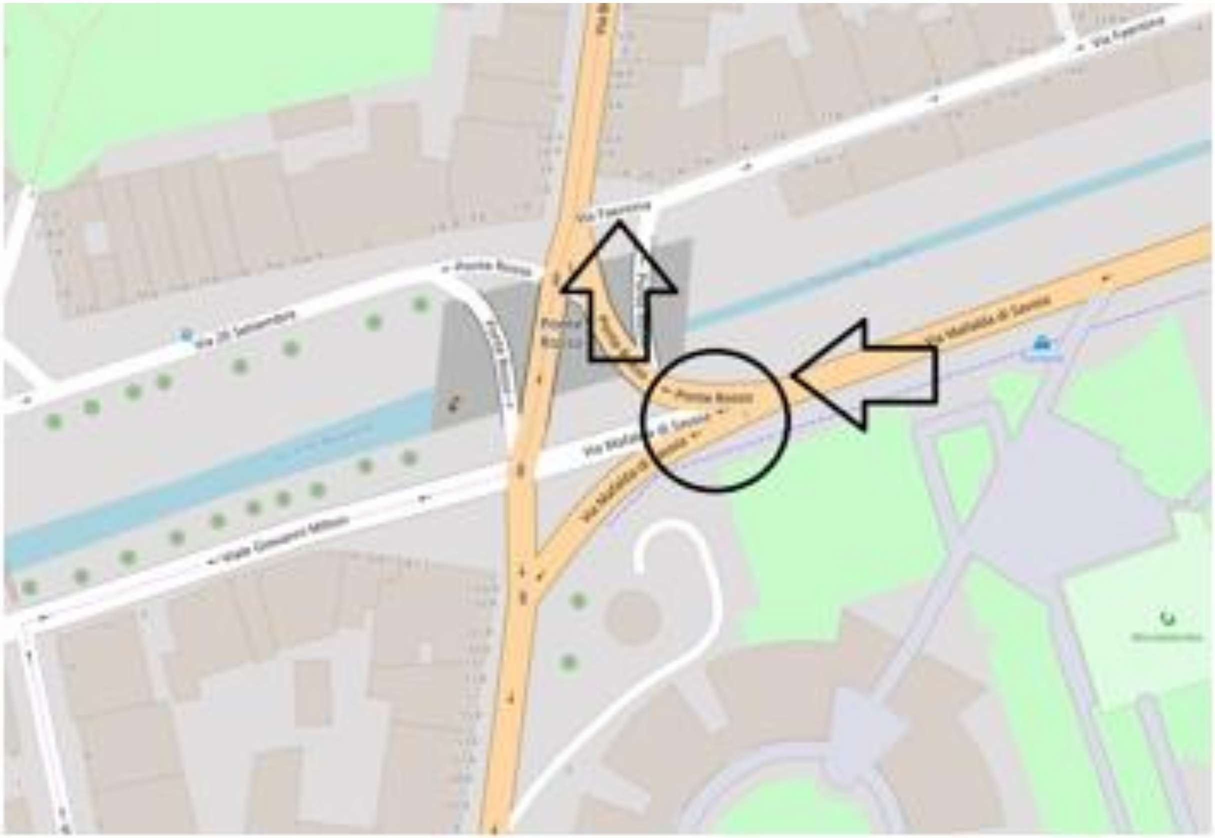
Junction distribution learning

Road Type: tertiary
Lanes: 1
Designated Lanes: 0
Restrictions: none
Learning Factor: 24
Elem. Type: T.O.C.
Length: 51
Direction: positive
…
Weight: 12.245%



Junction distribution learning

Road Type: primary
Lanes: 2
Designated Lanes: 0
Restrictions: none
Learning Factor: 111
Elem. Type: T.O.C.
Length: 60
Direction: positive
…
Weight: 56.633%



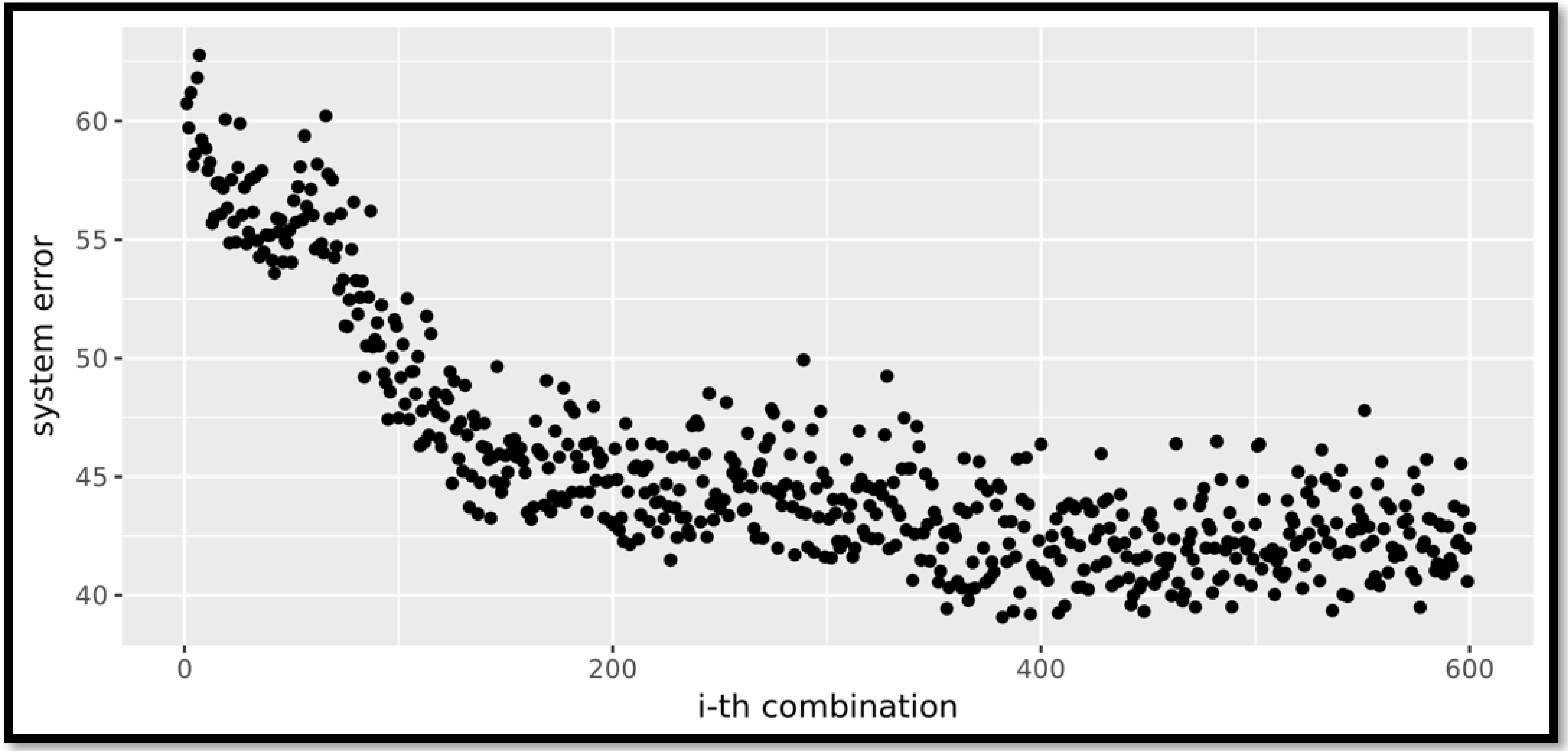
LOOCV approach

- This is performed by computing the solution which excludes data from each different sensor (all of them), so as to estimate the deviation from the calculated traffic density $\rho_c(t)$ in the road where the selected sensor is located, with respect to the density $\rho_M(t)$ measured by the sensor, for each time t. (Leave-One-Out-Crossing-Validation Approach)

- At a given location the **RMSE** is estimated as $\sqrt{\frac{\Sigma_{t=1}^{T}(\rho_C(t)-\rho_M(t))^2}{T}}$, where *T* is the total number of observations

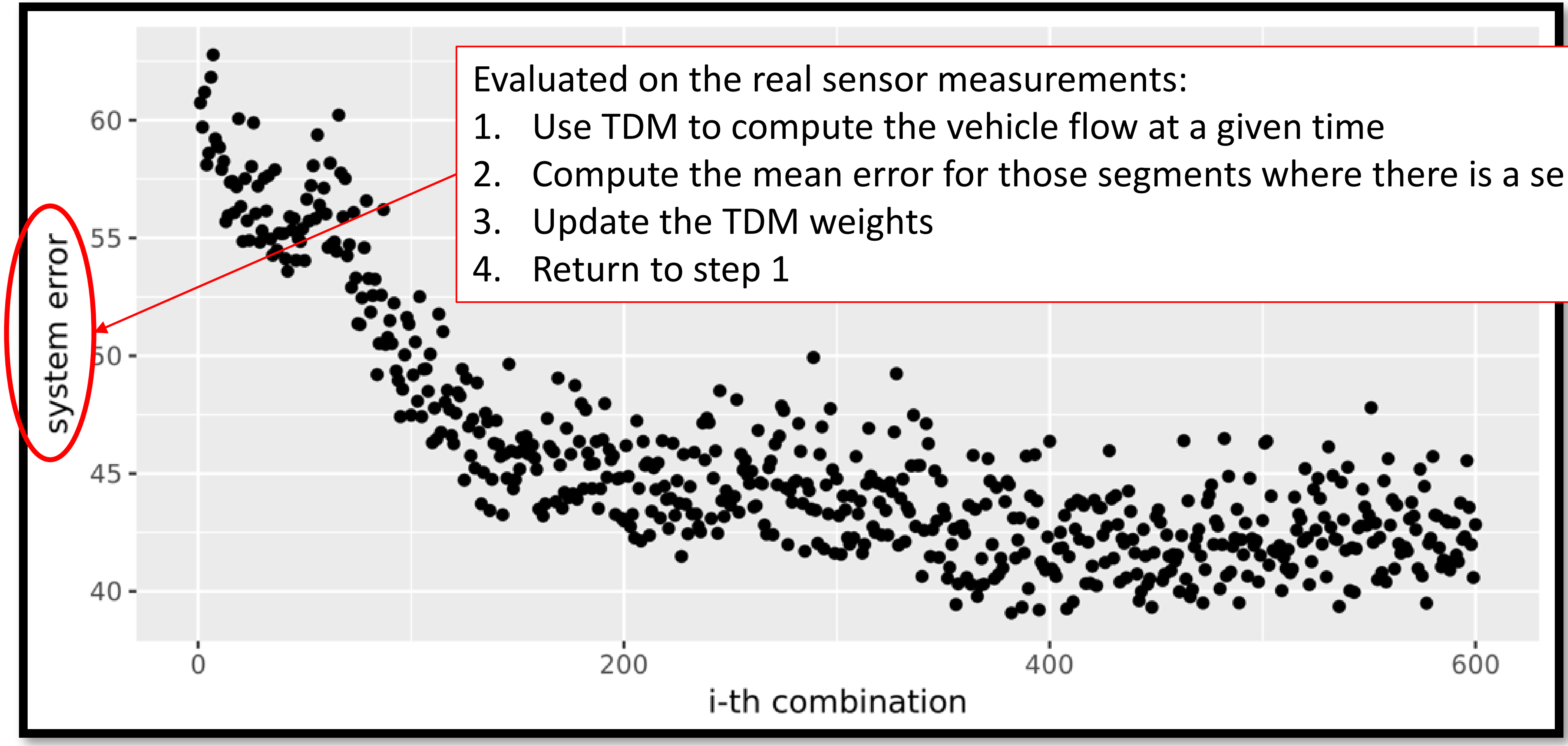
- At each iteration the RMSE for each sensor has been measured and also the so-called **system RMSE (system error)**, which is the average value of the measured RMSE of all the sensors.

Stochastic learning



In the x axis, the number of the learning iterations. In the y axis, the (decreasing) system error.

Stochastic learning

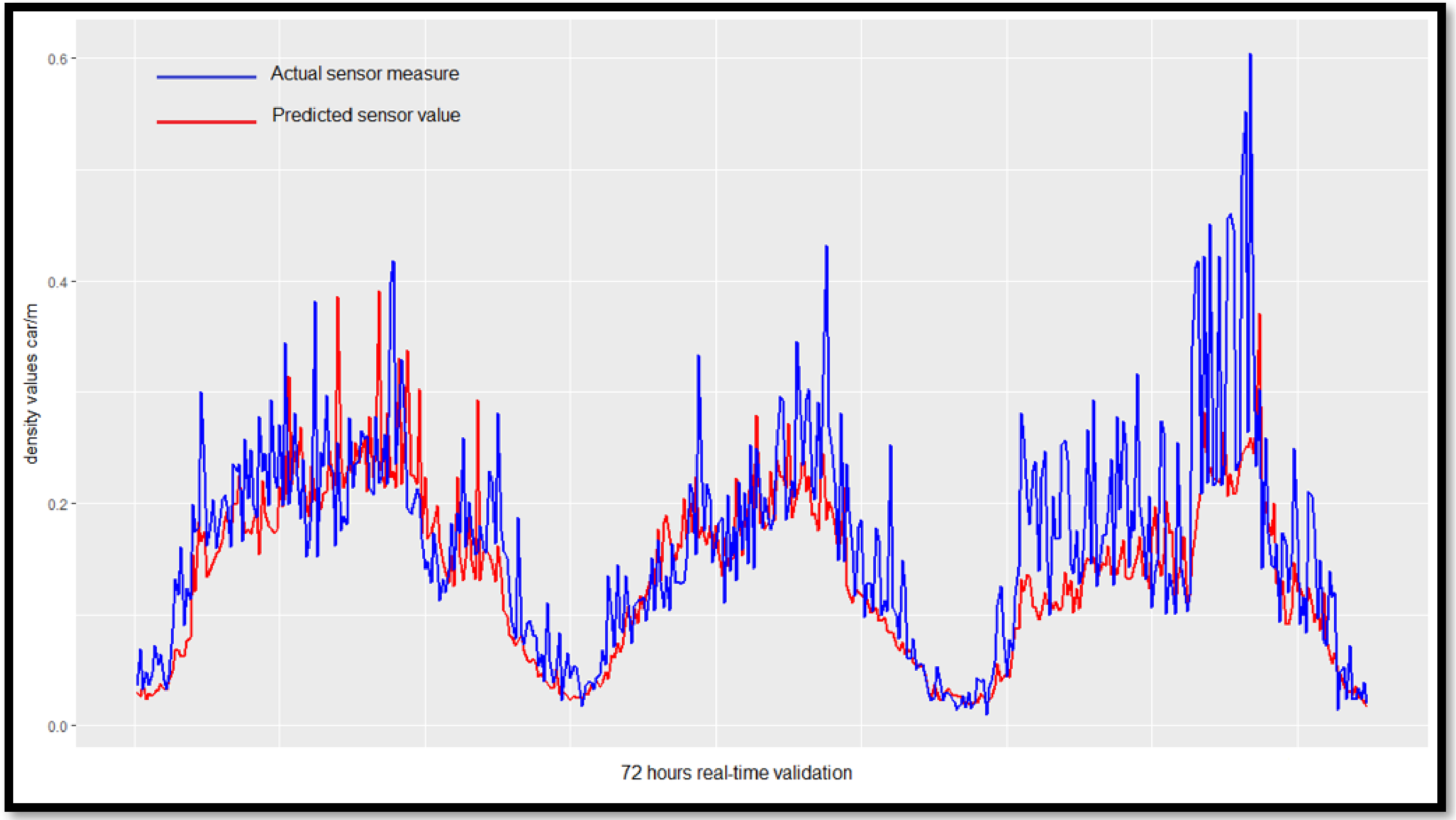


Evaluated on the real sensor measurements:

1. Use TDM to compute the vehicle flow at a given time
2. Compute the mean error for those segments where there is a sensor
3. Update the TDM weights
4. Return to step 1

In the x axis, the number of the learning iterations. In the y axis, the (decreasing) system error.

Validation approach



The validation is conducted by means of LOOCV in the case of optimal weights assignments in road graph.

The system error has been computed to be around 30%

The diagram refers to one in particular of the sensors, and it displays the predicted vs actual values over the time in the 72 hours validation.

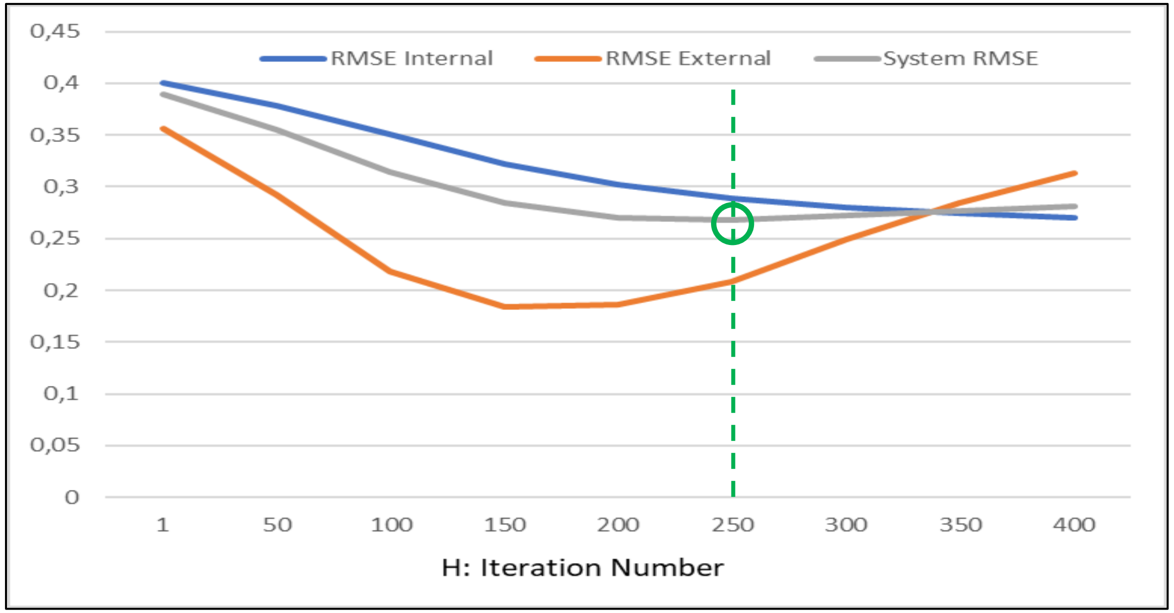
Error analysis

A deeper analysis of the results to be achieved by the solution we presented can be obtained by assessing the resulted traffic flow reconstruction during the real-time execution in order to understand:

- Identification of the most suitable number of iterations *H*
- Solution accuracy
- If the error in reconstruction depends on structural parameters of the urban network (i.e., sensor location)

Error analysis

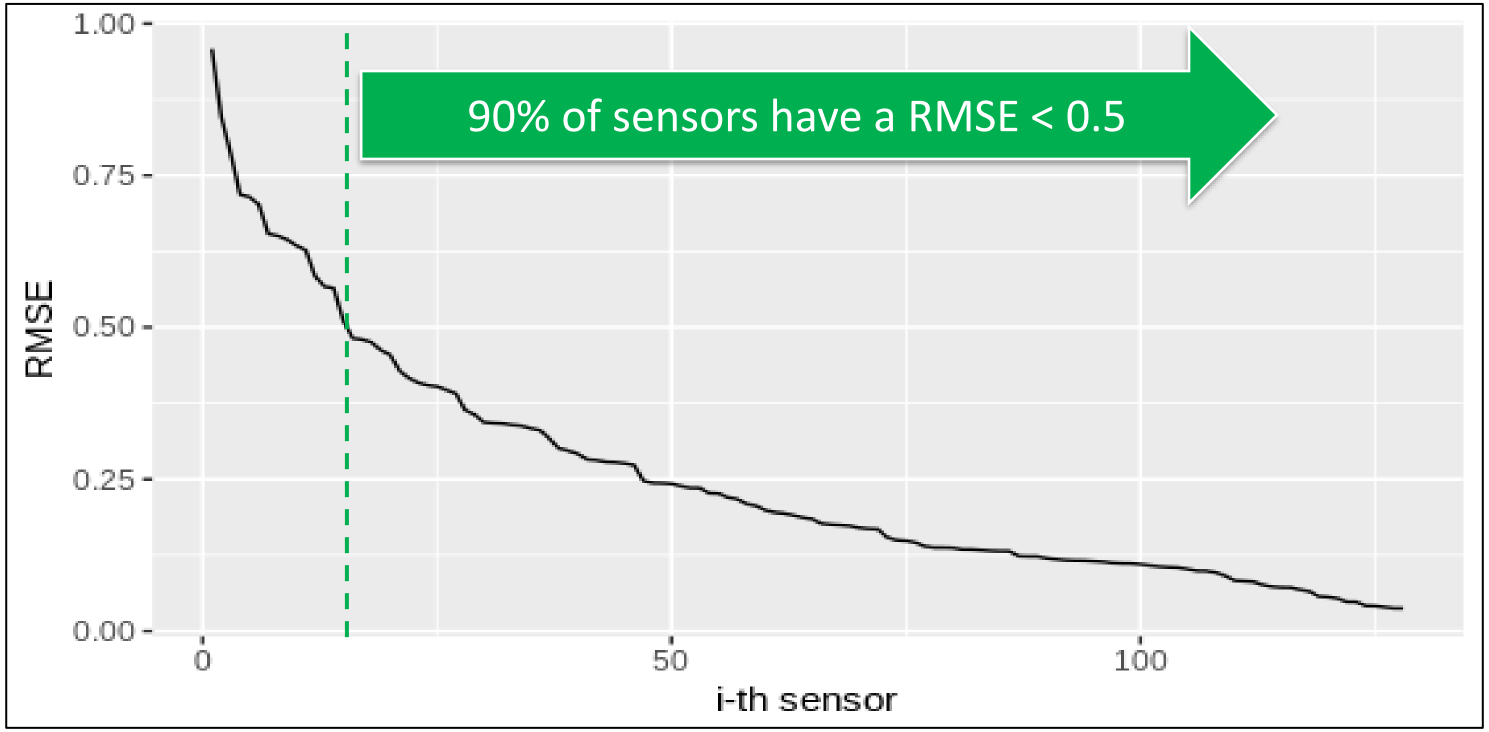
- The RMSE trends with respect to the iterations number *H* in the traffic flow reconstruction are shown.
- The average RMSE trend of the internal sensors is represented by the blue line, the average RMSE trend of the external sensors is represented by the orange line.
- In grey is reported the System RMSE having its minimum when H=250.



- Sensors on internal roads
- Sensors on external (edge) roads

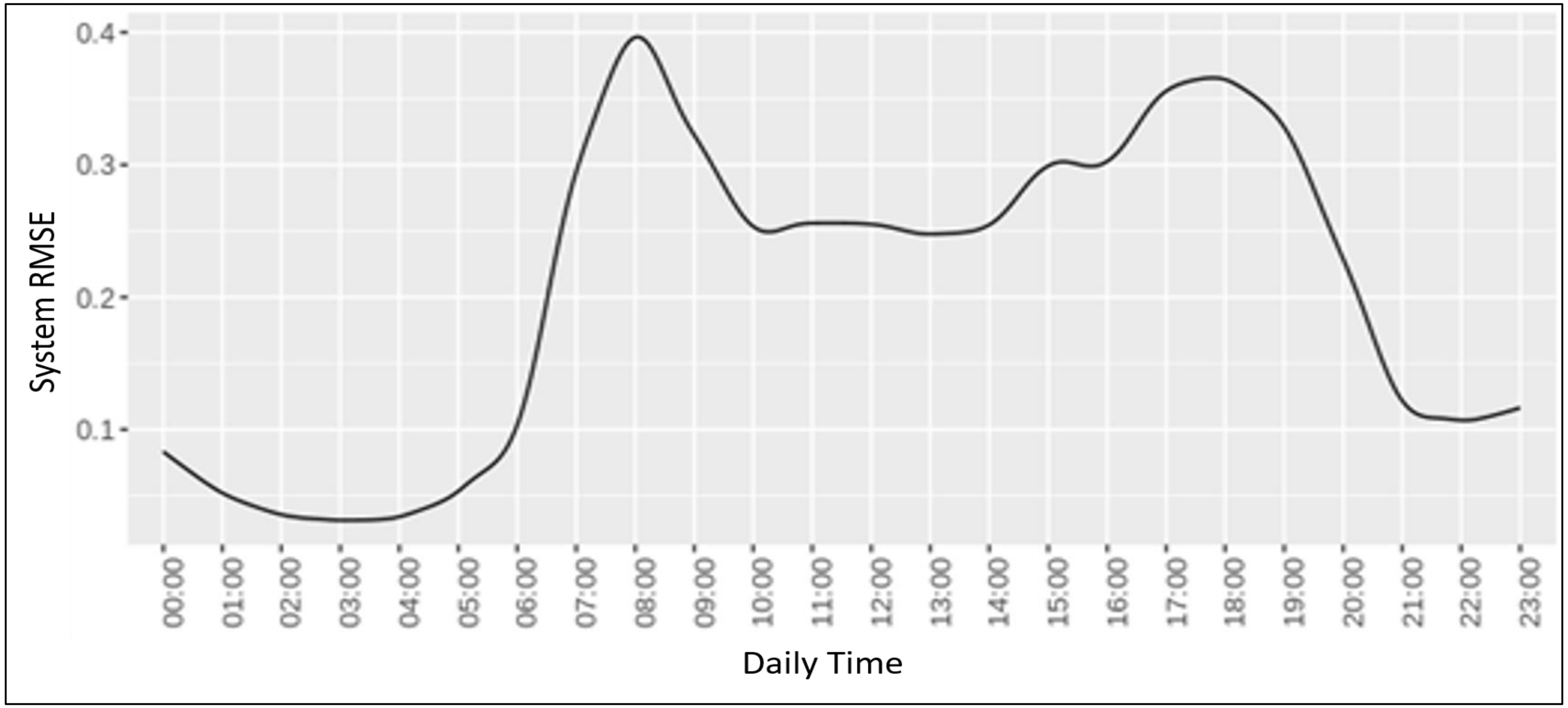
Error analysis

- The distribution of RMSE for each traffic sensor using H=250
- 90% of sensors have a RMSE value less than 0.5



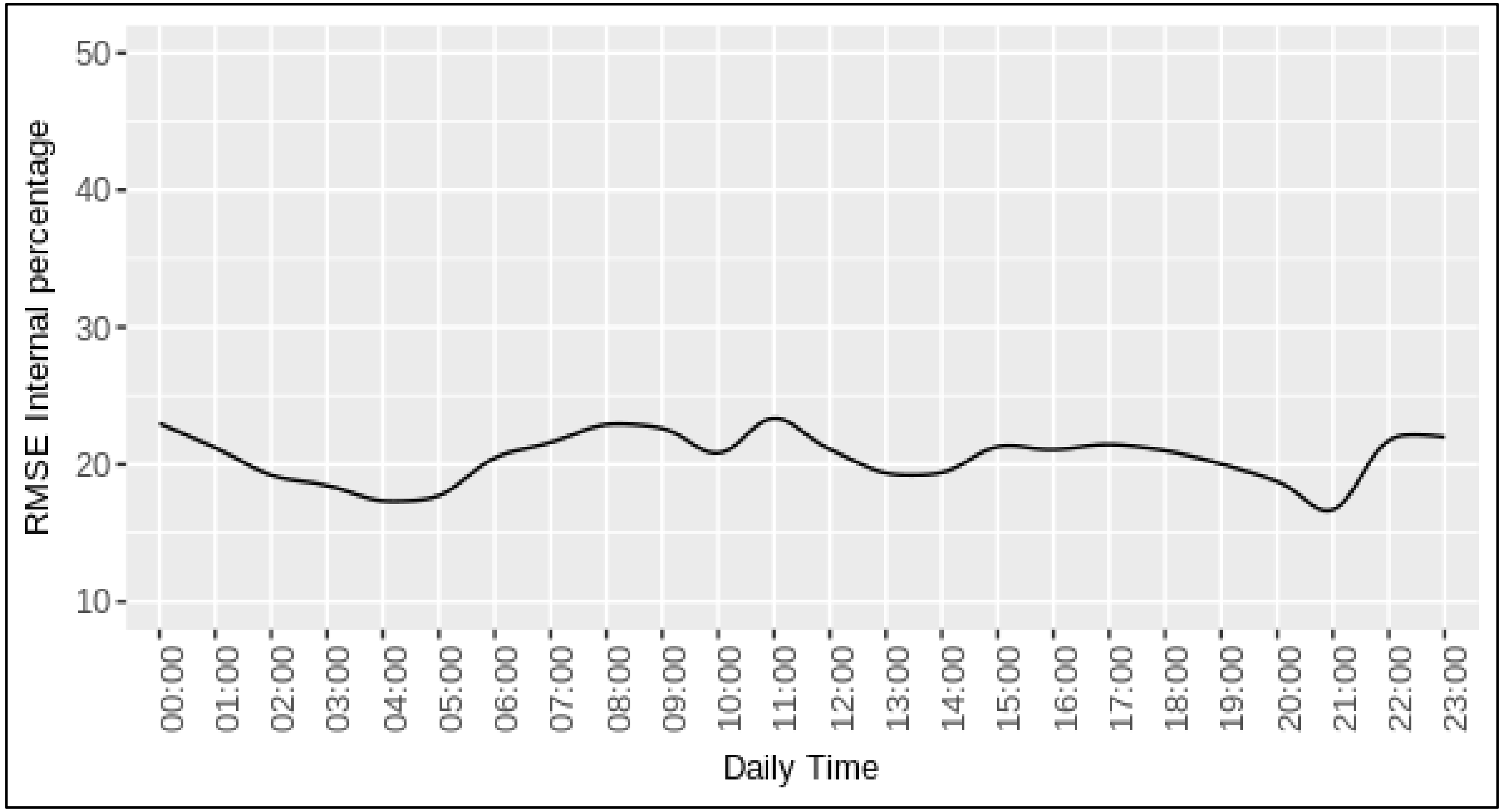
Error analysis

- Since traffic congestion in the city is typically related to the city incoming/outcoming flow according to the working activities of citizens, then also the RMSE value is affected to such behavior in the day.



Error analysis

- The RMSE is an absolute error measure with respect to the traffic density.
- The **ratio between the RMSE and the traffic density** (actual values) is almost constant



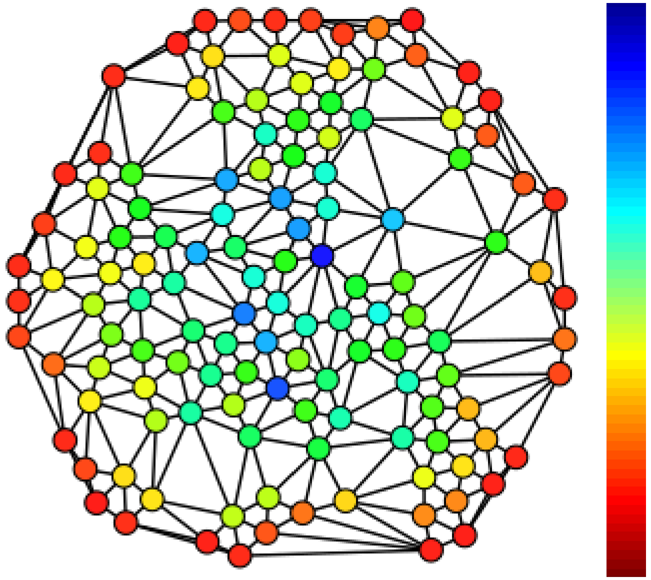
Error analysis

- The RMSE has a certain non-uniform distribution and a clear dependency on traffic volume.
- The error behavior is related to the topological characteristics of the road network.
- The error behavior of sensors are related to two topological features: **betweenness** and **eccentricity**

Error analysis

- The RMSE has a certain non-uniform distribution and a clear dependency on traffic volume.
- The error behavior is related to the topological characteristics of the road network.
- The error behavior of sensors are related to two topological features: **betweenness** and **eccentricity**

The **vertex betweenness** is a measure of centrality in a graph based on shortest paths. For every pair of vertices in a connected graph, there exists at least one shortest path between the vertices such that the number of edges that the path passes through (unweighted graphs) is minimized. The betweenness for each vertex is the number of these shortest paths that pass through the vertex



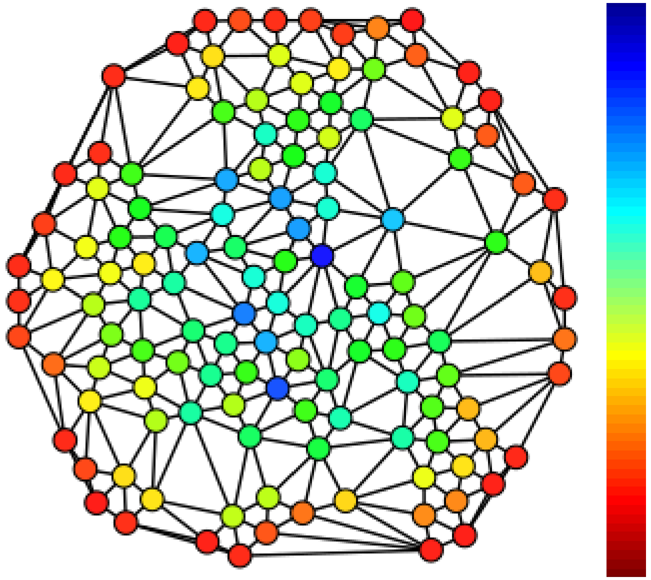
Error analysis

- The RMSE has a certain non-uniform distribution and a clear dependency on traffic volume.
- The error behavior is related to the topological characteristics of the road network.
- The error behavior of sensors are related to two topological features: **betweenness** and **eccentricity**

The *vertex betweenness* (also known as *betweenness centrality*) of a node v is the number of shortest paths which pass through v, formally we have

$$b(v) = \sum_{i \neq j, i \neq v, j \neq v} g_{ivj} / g_{ij}$$

where g_{ij} is the total number of the shortest paths from node i to node j and g_{ivj} is the number of those paths passing-through v. The vertex betweenness represents the degree to which nodes stand between each other and it measures the extent to which a vertex lies on paths between other vertices.

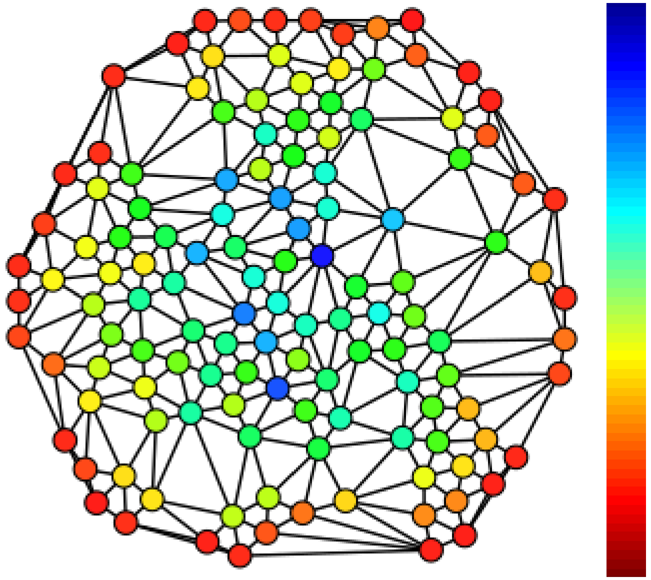


Error analysis

- The RMSE has a certain non-uniform distribution and a clear dependency on traffic volume.
- The error behavior is related to the topological characteristics of the road network.
- The error behavior of sensors are related to two topological features: **betweenness** and **eccentricity**

Nodes having high betweenness may have considerable influence within a road network by virtue of their control over traffic data passing between others.

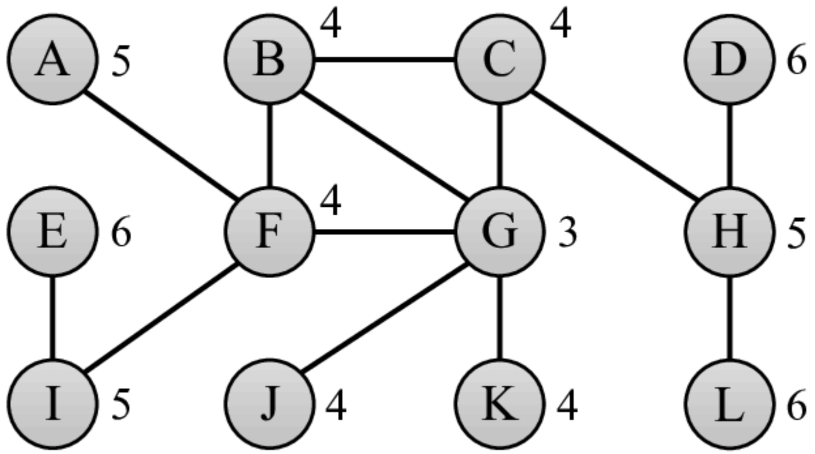
Such nodes are also the ones whose removal from the network will most disrupt communications between other vertices, because they lie on the largest number of paths inside the network.



Error analysis

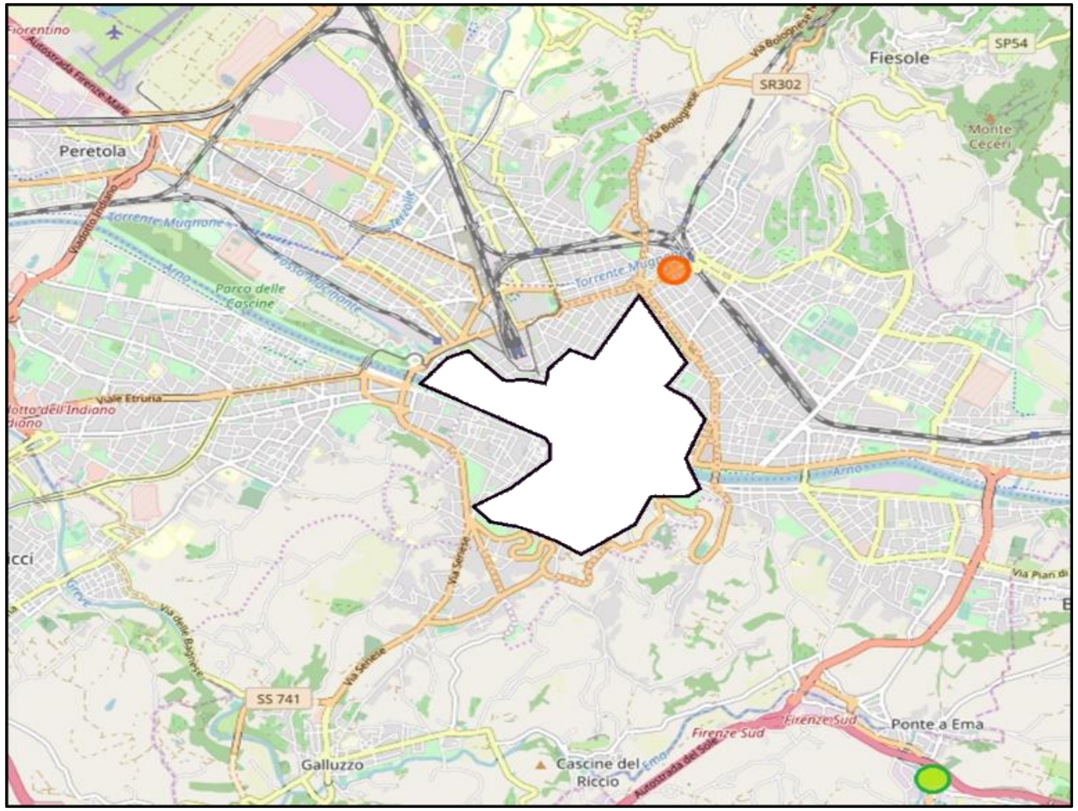
- The RMSE has a certain non-uniform distribution and a clear dependency on traffic volume.
- The error behavior is related to the topological characteristics of the road network.
- The error behavior of sensors are related to two topological features: **betweenness** and **eccentricity**

The **vertex eccentricity** is defined as the greatest shortest path distance between a vertex and any other vertex in the graph



Error analysis

- In orange the node having the maximum betweenness value, while in green the node having the maximum eccentricity value. The main restricted traffic zone is depicted in the center of the city in white.



Note that **betweenness** is located in proximity of one of the typical areas where traffic congestion often occurs.

On the other hand, nodes having high **eccentricity** are located in the decentralized zones of the urban graph admitting more distance from the other side of the network.

Error analysis

- A multilinear regression model has been conceived to verify the presence of an effective relationship between the RMSE and the topological metrics.
- Results:

Coefficient	Estimate		Std. Error	t-value	p-value
betweenness	β	0.80224	0.13097	6.125	< 0.05
eccentricity	γ	0.23256	0.02657	8.752	< 0.05

- The identified model is $Y_i = \beta x_i + \gamma z_i$ where Y_i, x_i, z_i are the RMSE, betweenness and eccentricity, respectively

Error analysis

- A general representation of the *Y* over the considered urban map is depicted in the following where locations having an intense pigmentation are affected by a greater error model

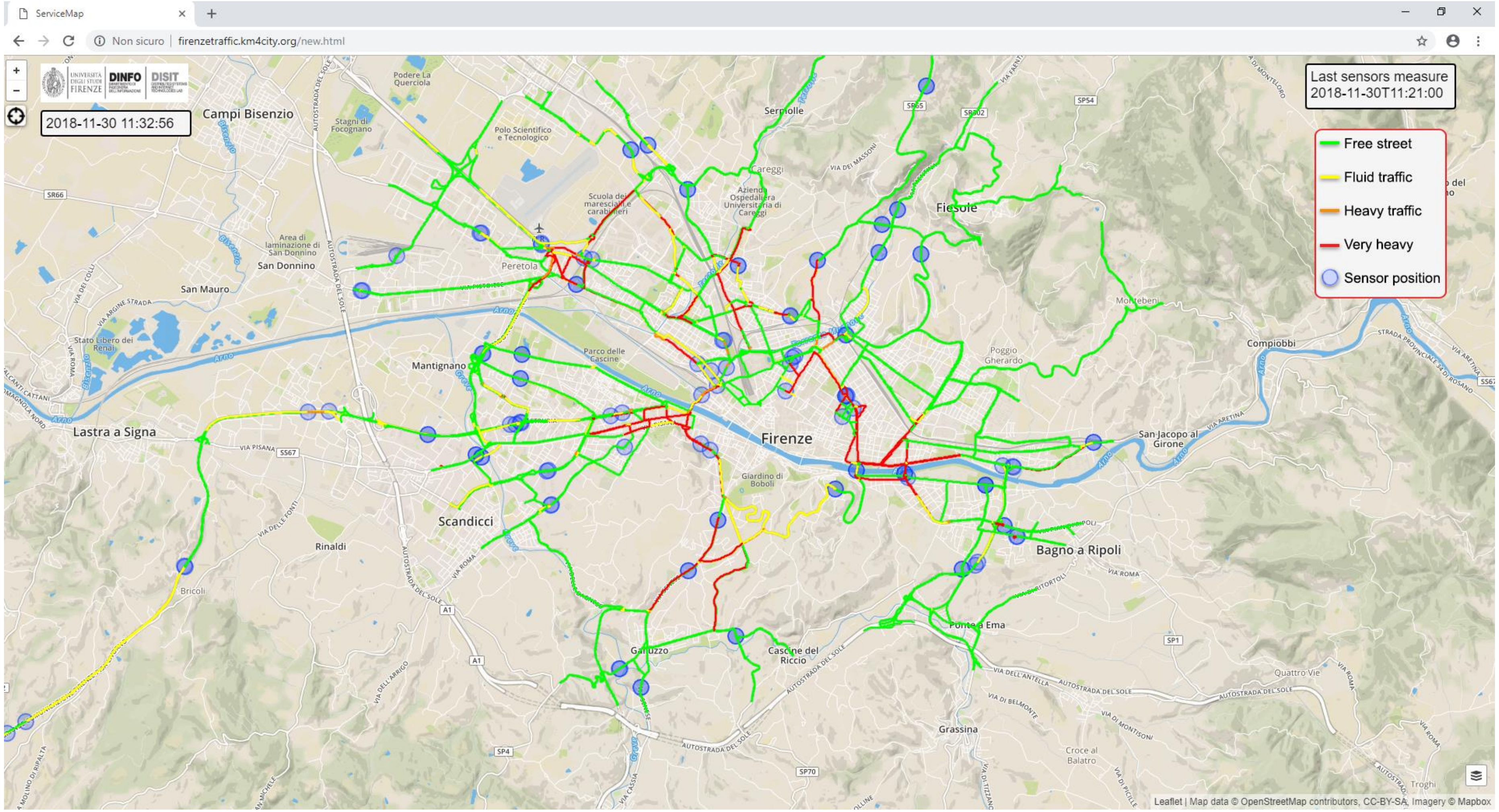
The critical values of the error function are not specifically located on segments with high traffic.

They are more related to the critical topological points of network, which are specific nodal cross points.

Displaying results

- Segments of road are categorized based on the road type and the number of lanes.
- Segments of each category that have one at least of the extremities that coincide with a traffic sensor, are used for determining the range of the traffic flows that can be observed on the specific category of segments.
- For each segment category, the range is partitioned into four subranges, that correspond to the four colors that you can find on the map.
- The reconstruction is presented to users through colored lines traced over the road paths on the city map.
- The date and time when the most up-to-date values from the sensors have been acquired can also be seen at the top-right corner of the map.

Displaying results



Data structure in real-time computing

- A network area of Florence consisting of
  - 173 data sensors (■)
  - 1532 junctions
  - 1377 road-segments

giving the estimation of the vehicular density in 31217 road-units having length 20/30 meters is considered to test the model

- Parallel computing solutions have been adopted

The idea of parallel computing

- Density array concept
- Numerical computation with respect to the position of the units inside a density array.

Units' position	Numerical method
First unit at the sensors' location	Eq. (3) with sensor measurement
First unit at the bound of the graph	Eq. (3) with no sensor measurement
Last unit at the bound of the graph	Eq. (4)
First unit	Eq. (6)
Last unit	Eq. (5)
Internal units	Eq. (2)

The idea of parallel computing

- Where:

$$u_m^{h+1} = u_m^h - \frac{\Delta t}{\Delta x}\left(F\left(u_m^h, u_{m+1}^h\right) - F\left(u_{m-1}^h, u_m^h\right)\right), \qquad (2)$$

$$u_0^{h+1} = u_0^h - \frac{\Delta t}{\Delta x}\left(F\left(u_0^h, u_1^h\right) - F\left(\rho_{(inc)}^h, u_0^h\right)\right) \qquad (3)$$

$$u_N^{h+1} = u_N^h - \frac{\Delta t}{\Delta x}\left(F\left(u_N^h, \rho_{(out)}^h\right) - F\left(u_{N-1}^h, u_N^h\right)\right). \qquad (4)$$

The idea of parallel computing

$$u_N^{h+1} = u_N^h - \frac{\Delta t}{\Delta x}\left(\gamma_i - F\left(u_{N-1}^h, u_N^h\right)\right) \quad (5)$$

$$u_0^{h+1} = u_0^h - \frac{\Delta t}{\Delta x}\left(F\left(u_0^h, u_1^h\right) - \gamma_j\right), \quad (6)$$

where: γ_i, γ_j are the incoming/outcoming flows such that $\gamma_j \simeq w_{ji}\gamma_i$

The idea of parallel computing

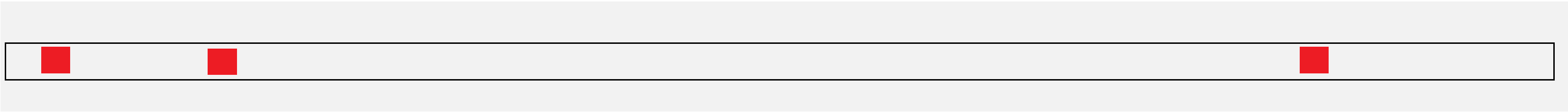
- Density array concept: Computing a given equation in different processing units at the same time.
- It seems to be like a sort of WHERE construct of many parallel programming languages.
- Traffic Distribution Matrix (TDM) constitutes an input data for the estimation of density arrays, so they are not independent processes (their parallelization is separately conducted).

The idea of parallel computing

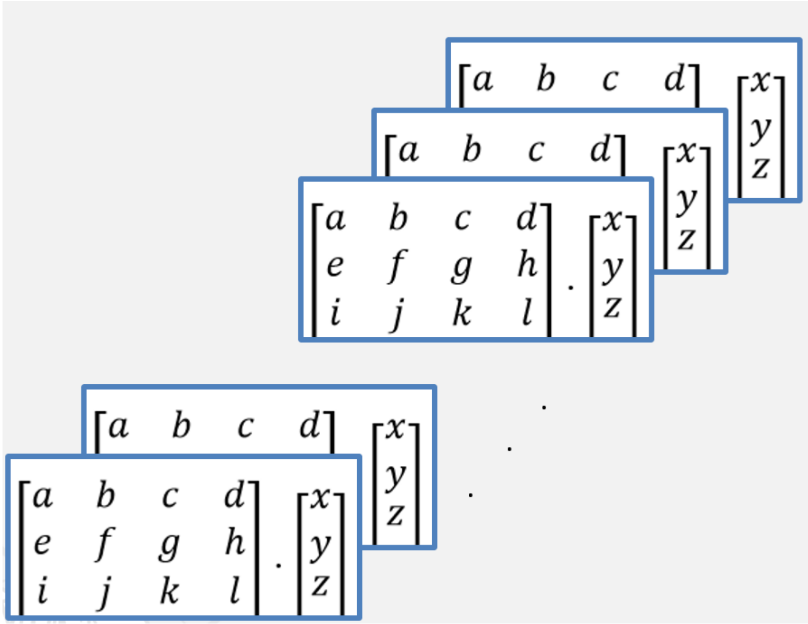
- Events involving the distribution at the nodes are necessarily independent one another
- Let I and O be maximal number of incoming and outcoming road-segments, then TDM maximizing the model and creating a 3D-structure of dimension O×I×V, where V is the number of junctions in the network.

Graphical idea of parallel computing

Density Array:

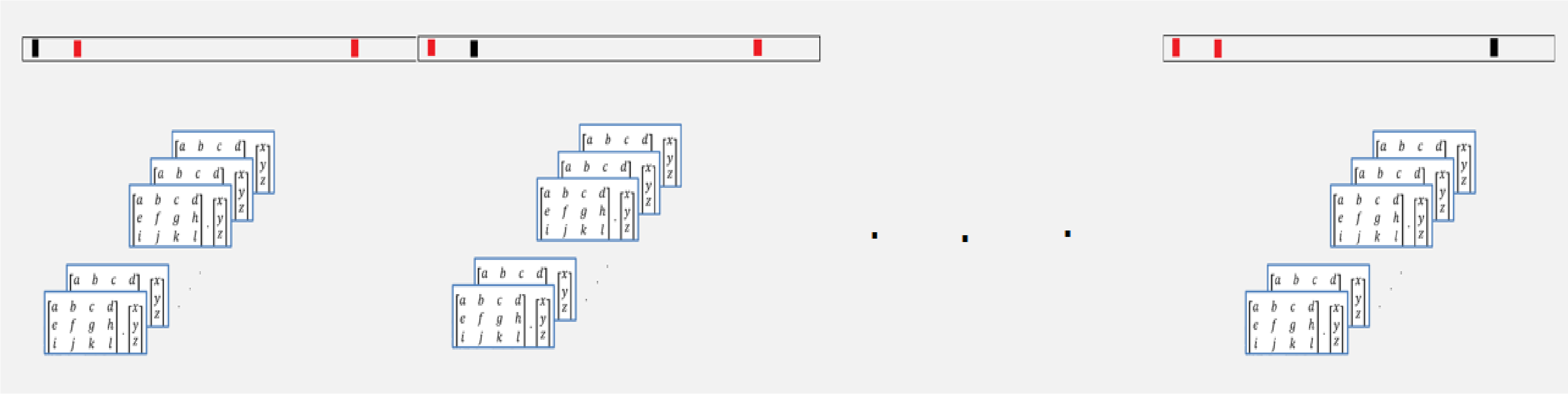


TDM 3D-structure:



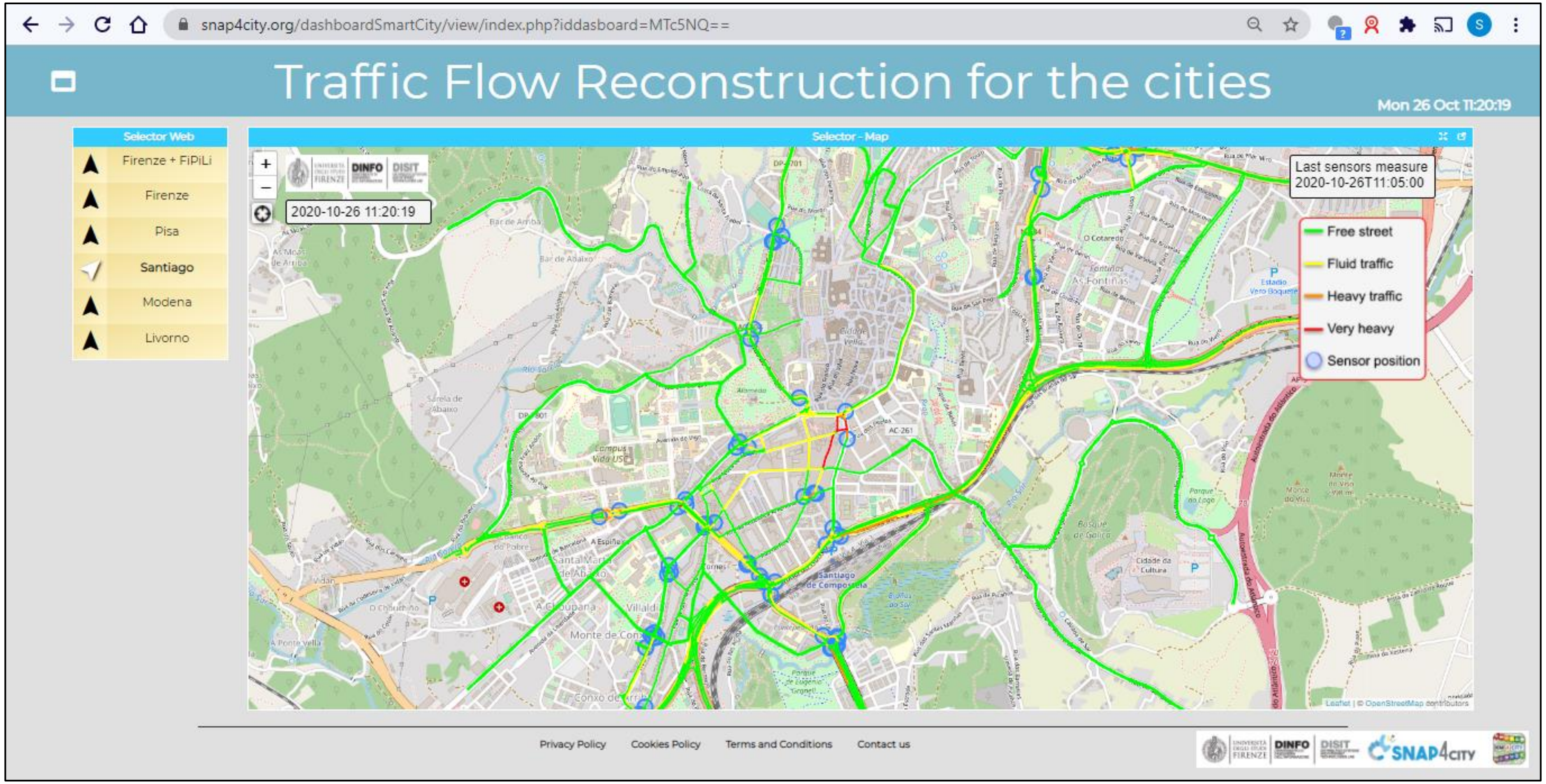
Graphical idea of parallel computing

- Leave-One-Out-Crossing-Validation Approach: Parallel data structures concatenation



General Approach

More cities



https://www.snap4city.org/dashboardSmartCity/view/index.php?iddasboard=MTc5NQ==

Hybrid Model for Traffic Reconstruction

FROM DATA TO DIGITAL TWIN

FOR SMART MOBILITY

Hybrid model for Traffic Reconstruction

- Traffic Flow Reconstruction (*TFR*) approaches can be classified into three main categories: *model-driven*, *data-driven* and *mixed* approaches.

- **TFR Model-driven** approaches are generally those taking into account the physical model of traffic in the spatio-temporal domain, such as both agent-based and those solving differential equations.

Hybrid model for Traffic Reconstruction

- **TFR Data-driven** approaches should derive traffic state by means of the dependences learned from observed data using statistical or machine learning methods. They should rely on real time and historical data in each road segment to extrapolate data in each and every segment.

- Generally, **Hybrid** approaches combine a model-driven method with a data-driven method to achieve more accurate and efficient results.

Background and Notation

- TFR Model-driven solution via PDEs is denoted by SRA4TF

- At each timestamp, the SRA4TF solution produces a value of traffic flow density in each road segment of the network, typically of 20 meter, as unit, that is the TFR. The accuracy of SRA4TF solution mainly depends on the stochastic relaxation approach for estimating Traffic Distribution Matrices (TDMs), which are the traffic flow distributions at junctions. TDMs describe the percentage of vehicles getting out each outcoming road with respect to those getting in from each incoming road of a junction.

Background and Notation

- The TDM is defined as $TDM = \{w_{ji}\}_{j=n+1,\dots,n+m, i=1,\dots,n}$ so that $0 < w_{ji} < 1$ and $\sum_{j=n+1}^{n+m} w_{ji} = 1$, for $i = 1, \dots, n$ and $j = n+1, \dots, n+m$,

- where w_{ji} is the percentage of vehicles arriving from the i-th incoming road and taking the j-th outcoming road (assuming that, on each junction, the incoming flux coincides with the outcoming flux).

- The values of w_{ji} depend on the time of the day, on the road size, cross light settings, etc., and thus, it is unknown a priori.

Background and Notation

- The computing of TFR is progressively performed on a parallel architecture. The estimation of traffic flow density for a city at time instant t would depend on traffic flows at time t-1 in the whole network, and on the new measures coming from sensors at time t.

- Once $TDM(t)$ are estimated, the SRA4TF solution computes the TFR in the road network and verify the Root Mean Square Error, RMSE, (or Mean Absolute Error, MAE) with respect to actual values in sensor locations.

Background and Notation

- Error estimation is performed by computing the solution excluding data from each different sensor (all of them) by means of a Leave-One-Out Crossing-Validation approach (LOOCV), so as to estimate the deviation from the reconstructed traffic density $\rho^R(t)$, with respect to the observed density by the sensor $\rho^O(t)$, for each time t in T.

- We refer to *R* and *O* to denote reconstructed and observed traffic flow densities, respectively. Then, in a road network having m traffic sensors, the LOOCV approach consists in the application of the model to the set of the observed data at time t, that is $\boldsymbol{O}(t) = \{O_1(t), \ldots, O_m(t)\}$, by excluding the k-th observation $O_k(t)$ from $\boldsymbol{O}(t)$, for each $k = 1, \ldots, m$.

Background and Notation

- The model is applied to the remaining set of $m-1$ sensors' observations and the reconstructed density $R_k(t)$ in the road segment (unit) where the k-th sensor is located can be estimated and compared with $O_k(t)$ via RMSE or MAE estimation as follows:

- RMSE(k) = $\sqrt{\frac{\Sigma_{t=1}^{T}\left(R_k(t)-O_k(t)\right)^2}{T}}$,

- MAE(k) = $\frac{\Sigma_{t=1}^{T}(|R_k(t)-O_k(t)|)}{T}$.

Background and Notation

- For each round, the stochastic relaxation may produce a new minimum of the *RMSE* that is taken as a reference status together with the produced $TDM(t)$, for the next iterations.

- At each timestamp, the *RMSE(k)* for each sensor in the LOOCV is measured and the *RMSE(system)* is considered:

- *RMSE(system)* = $\frac{1}{m}\sum_{k=1}^{m} \; RMSE\ (k)$.

Hybrid Architecture for improving precision

GOALS:

- improvement of precision in dense traffic flow estimation, reduction of *RMSE(system)*.
- reduction of execution time.
- usage of ML together with the exploitation of knowledge about the road network traffic and the SRA4TF solution.

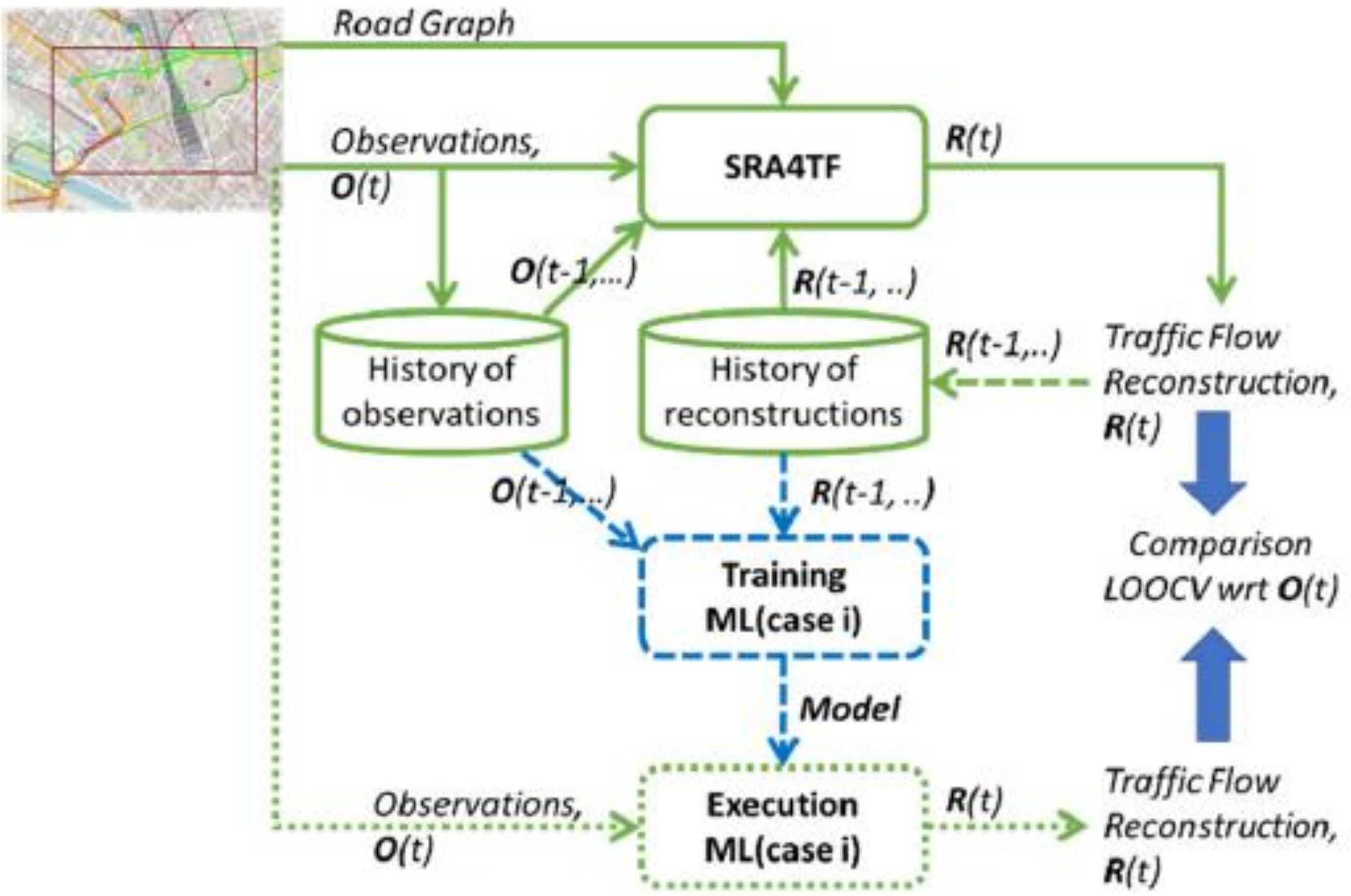
Summary for SRA4TF

- In the following, with $\boldsymbol{O}(t)$ is denoted the vector of the observations (measures) from the sensors at time t, while $\boldsymbol{R}(t)$ is the vector of the traffic density reconstructed in the other segments of the road network at time t. The SRA4TF produces a traffic density for the whole road network which can be regarded a vector $\boldsymbol{R}(t)$ as follows:

$$SRA4TF(\boldsymbol{O}(t-1), \boldsymbol{R}(t-1), \boldsymbol{O}(t), RoadGraph) \rightarrow \boldsymbol{R}(t).$$

- Having *m* traffic sensors in a road network, we obtain that the total road segments (units) in the road network is m+n considering $\boldsymbol{O}(t) = \{O_1(t), \ldots, O_m(t)\}$ and $\boldsymbol{R}(t) = \{R_1(t), \ldots, R_n(t)\}$.

Hybrid Architecture



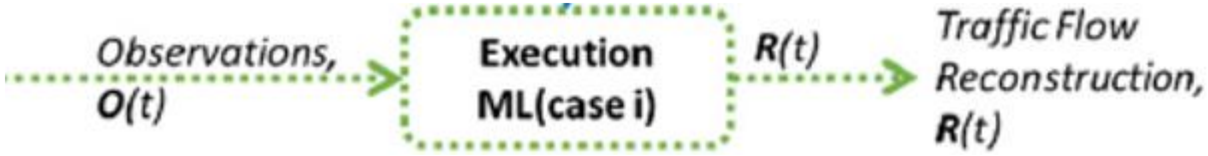
The training data flows are reported as *dashed lines*, while the execution data flows are represented as *dotted lines*.

The training phase is fed by using data produced by both observation and SRA4TF solution (*green lines*).

Hybrid Architecture

- ML approach learns a Model able to produce a full set of traffic flow densities on the basis of observations, that is the TFR, at each time instant, disregarding its temporal evolution (*case (i)*).

$$\hat{f}(\boldsymbol{O}(t)) \rightarrow \boldsymbol{R}(t)$$

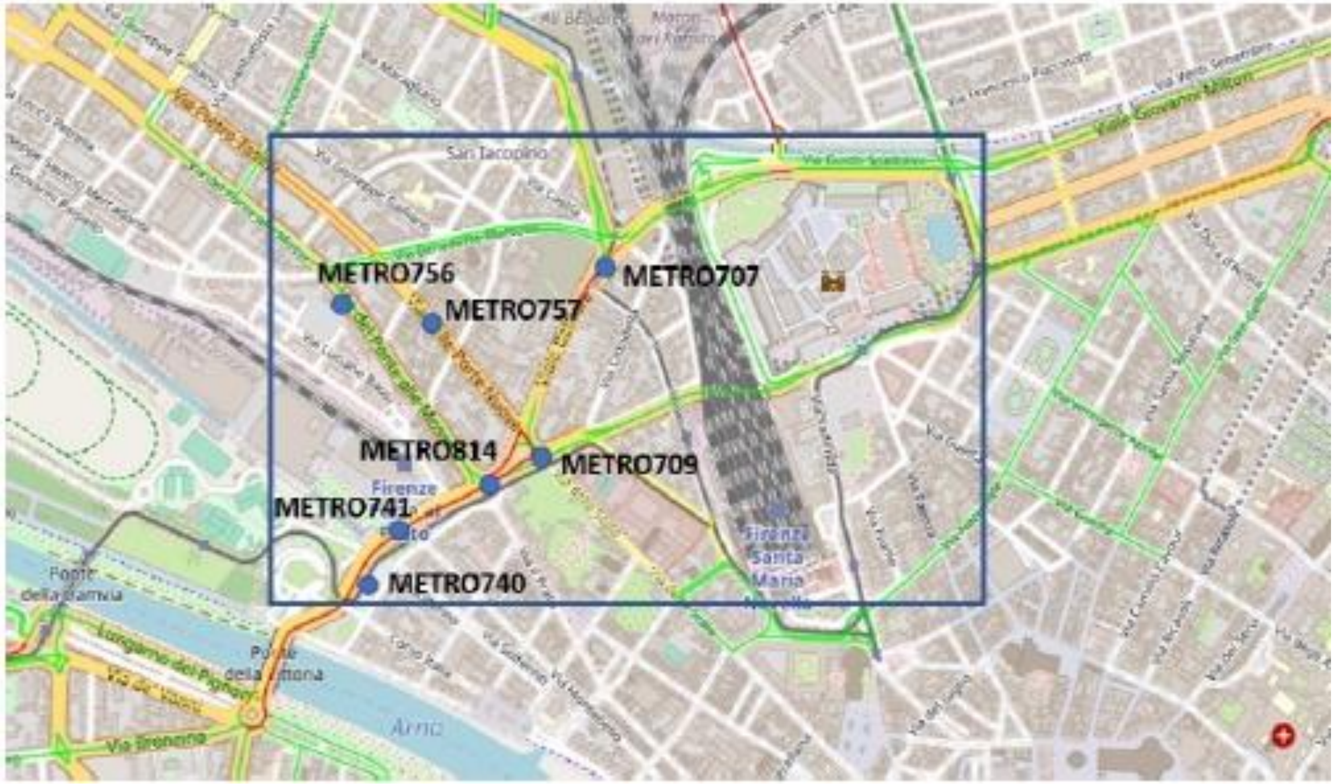


Hybrid Architecture

Description:

- The SRA4TF is used for generating dense traffic flow training data with respect to observed values, for the ML function $\hat{f}(.)$
- function $\hat{f}(.)$ learns how to compute the TFR according with $\boldsymbol{R}(t)$ on the basis of the observed values $\boldsymbol{O}(t)$
- Once trained the ML solution, it could be used at run time to produce the dense traffic flow results in faster manner (with respect to the PDE iterative solution)
- the results can be compared with the measured values obtained by sensors by using the LOOCV approach and estimating the RMSE

The case of study



The considered area is constituted by **735** road segments (units) and **103** intersections/junctions or nodes, thus TDMs for SRA4TF.

7 traffic sensors

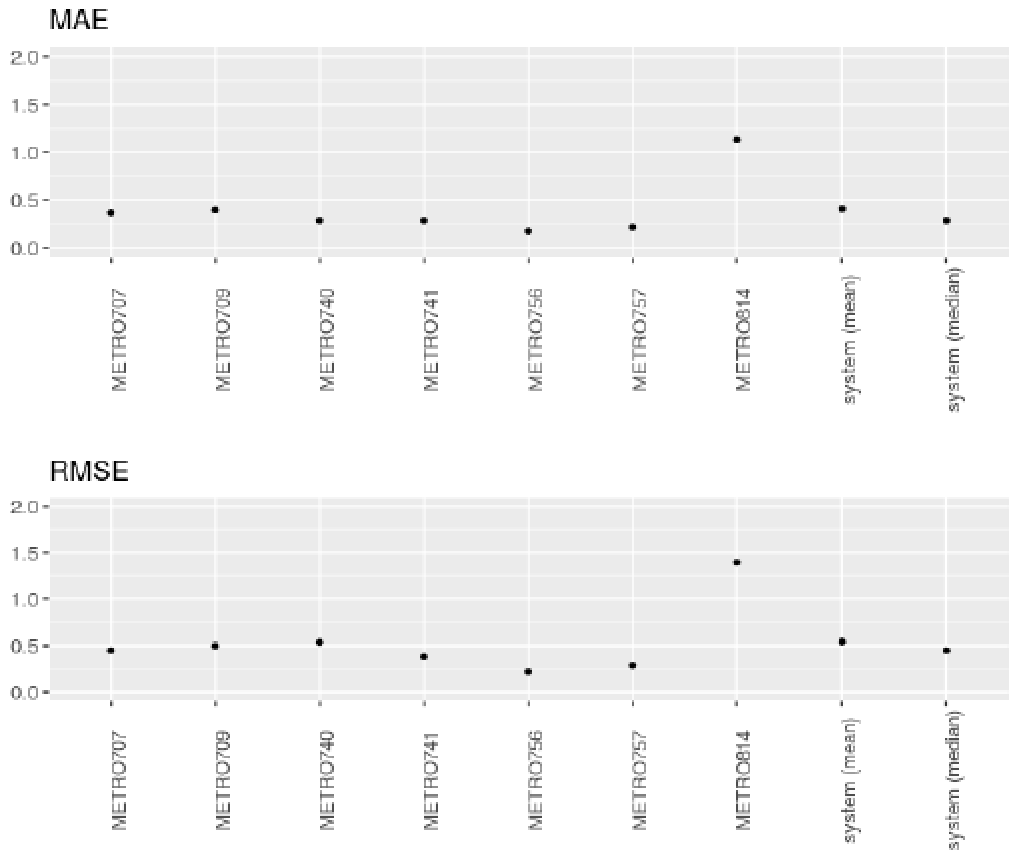
The case of study

- The training set is based on traffic sensor data updates every (about) 10 minutes and 144 measures are observed per day per sensor.

- Period: 24 (hours) per 121 (days).

- The entire dataset is composed by 13208 observations O(.) from the 7 sensors, while the 13208 reconstructions R(.) of the traffic density can be computed in the 728 units that compose the selected subnet of **735**.

The case of study (SRA4TF)

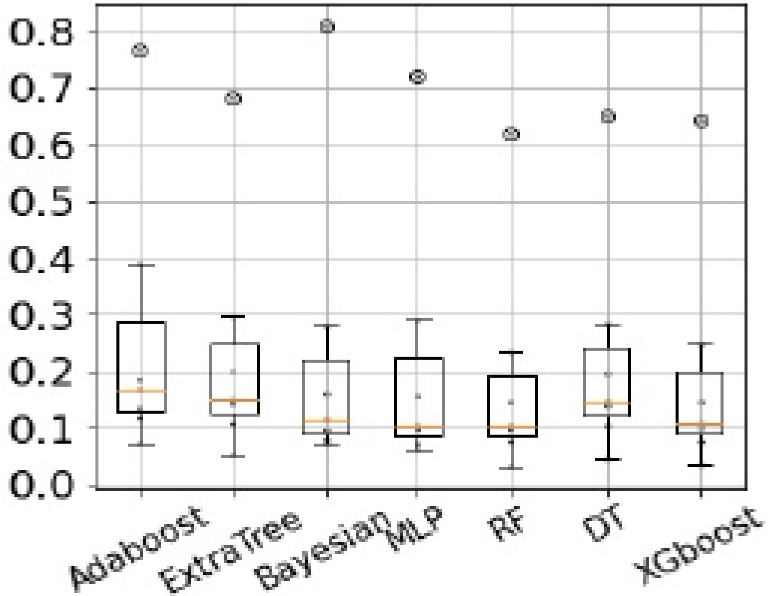


The assessment reported in reports the MAE and RMSE (at level of sensor location using LOOCV), over 3500 timestamps (which constituted about the 30% of the above-described dataset).

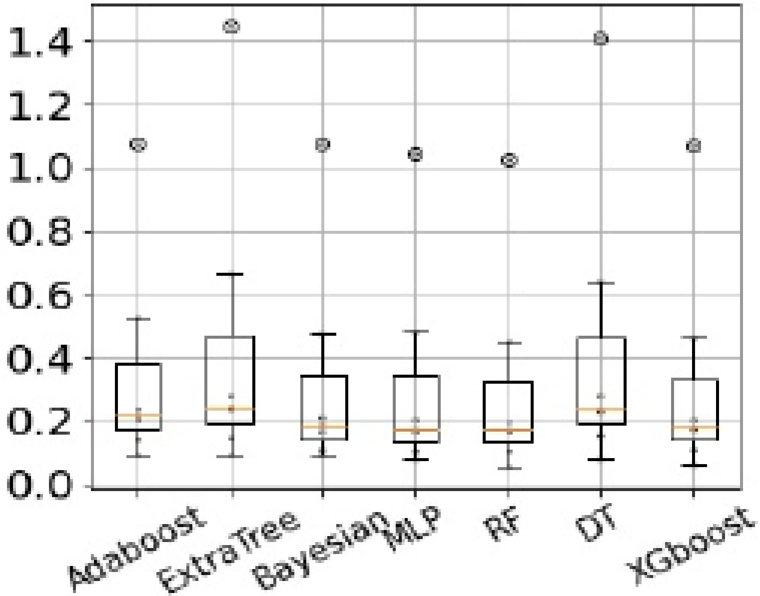
The case of study (ML)

- Different ML solutions have been compared according to the proposed architecture, with the aim of identifying the best solution to learn and compute TFR.
- To this end, we have considered ensemble learning techniques such as **Adaboost**, Random Forest, **RF**, and **Xgboost**
- However, we took into considerations also more concise and interpretable models such as a **Bayesian** regressor, a Decision Tree, **DT**, **ExtraTree**, and multi-layer perceptron, **MLP.**
- validation data set is constituted of about 30% of the entire dataset and the remaining 70% is devoted to the training phase

Results



MAE(system)



RMSE(system)

For STR4FT:
MAE(system) = 0.4
RMSE(system) = 0.53

Results

TFR DEVIATION ΔR ACCORDING TO THE DIFFERENT MODELS IN CASE (i)

Model	ΔR
Bayesian	0.0942
Adaboost	0.0848
MLP	0.0676
ExtraTree	0.0552
DT	0.0519
XGboost	0.0467
RF	0.0435

$$\Delta R = \frac{1}{T}\sum_{t=1}^{T} \Delta R(t).$$

where instant deviation is:

$$\Delta R(t) = \frac{1}{n}\sum_{z=1}^{n} |\hat{R}_z(t) - R_z(t)|$$

and where: $R_z(t)$ is the traffic density value of the z^{th} reconstructed unit at timestamp t using SRT4TF and $\hat{R}_z(t)$ is the reconstructed traffic density value by the data-driven model of the z^{th} unit at timestamp t by the data-driven model.

Results

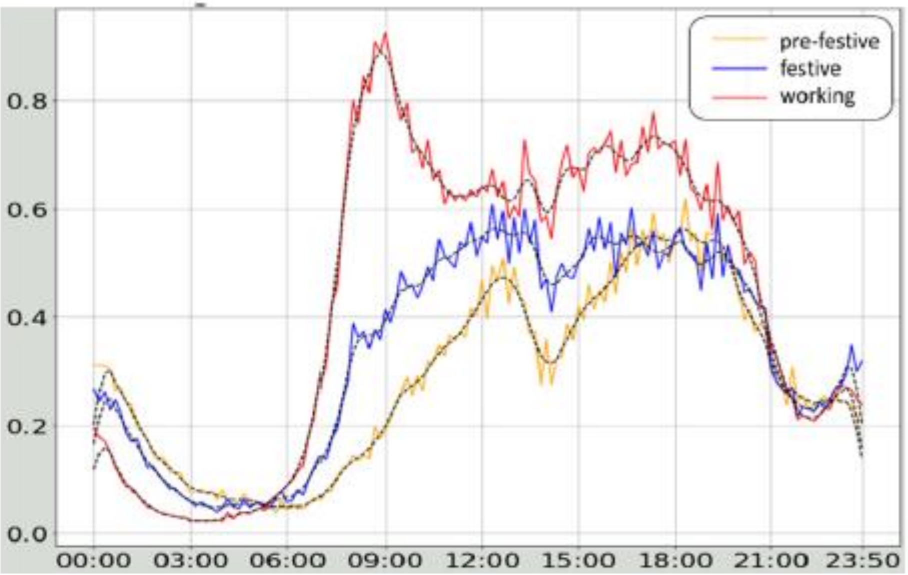
EXECUTION TIMES FOR THE TFR PERFORMED WITH SRA4TF ONLY, AND VIA ML MODELS FOR CASE (i)

Model	Test Time (s)
SRA4TF	3685.15
RF	1627.30
XGboost	744.50
Adaboost	43.66
DT	19.52
ExtraTree	18.47
Bayesian	4.69
MLP	0.22

Advanced Approaches

- Coding Temporal Information
- addressing problems related to discontinuous input data (missing data)
- data seasonality: festive, pre-festive and working days (clustering)
- Typical trends

$$\hat{f}(\mathbf{O}(t), \quad \ldots \quad) \rightarrow \mathbf{R}(t)$$



Thanks for the attention