From Sensors Data to Urban Traffic Flow Analysis

Conference Paper · October 2019

4 authors:

Laura Po
Università degli Studi di Modena e Reggio Emilia
41 PUBLICATIONS **324** CITATIONS
SEE PROFILE

Federica Rollo
Università degli Studi di Modena e Reggio Emilia
5 PUBLICATIONS **9** CITATIONS
SEE PROFILE

Chiara Bachechi
Università degli Studi di Modena e Reggio Emilia
3 PUBLICATIONS **6** CITATIONS
SEE PROFILE

Alberto Corni
Università degli Studi di Modena e Reggio Emilia
13 PUBLICATIONS **165** CITATIONS
SEE PROFILE

From Sensors Data to Urban Traffic Flow Analysis

Laura Po, Federica Rollo, Chiara Bachechi, Alberto Corni
'Enzo Ferrari' Engineering Department
Via Vivarelli, 10 , 42125 Modena, ITALY
Email: name.surname@unimore.it

***Abstract*—By 2050, almost 70% of the population will live in cities. As the population grows, travel demand increases and this might affect air quality in urban areas. Traffic is among the main sources of pollution within cities. Therefore monitoring urban traffic means not only identifying congestion and managing accidents but also preventing the impact on air pollution. Urban traffic modeling and analysis is part of the advanced traffic intelligent management technologies that has become a crucial sector for smart cities. Its main purpose is to predict congestion states of a specific urban transport network and propose improvements in the traffic network that might result into a decrease of the travel times, air pollution and fuel consumption.**

This paper describes the implementation of an urban traffic flow model in the city of Modena based on real traffic sensor data. This is part of a wide European project that aims at studying the correlation among traffic and air pollution, therefore at combining traffic and air pollution simulations for testing various urban scenarios and raising citizen awareness about air quality where necessary.

I. Introduction

Traffic is a significant source of air pollution, especially of particulate matter (PM) and nitrogen dioxide (NO2), and remains the main source of environmental noise in Europe [1], although some emissions have been reduced in the last decade. An estimated 3.7 million premature deaths are attributed to ambient (outdoor) air pollution, based on WHO data from 2012. Overall, higher urban air pollution concentrations increase the risk for cardiovascular and respiratory disease, cancer and adverse birth outcomes, and also are associated with higher death rates.

As the population grows, the travel demands increase and, consequently, the volume of vehicles that are moving on urban roads also explodes. This continuing growth in vehicle use means that efforts to reduce emissions from individual vehicles are in danger of being overtaken by increases in the volume of traffic. Since traffic is among the main sources of pollution within an urban area, monitoring traffic flows means not only identifying congestion and managing accidents but also preventing the impact on air quality. Smart cities should be an integrated ecosystem able to offer a better quality of life to citizens. Modelling traffic is a critical issues for several cities [2]–[4].

In this paper, we describe how a smart city can take advantage of a traffic model generated from the collection and integration of real traffic data. This will be the first step in order to understand the effect of traffic on urban air quality. This work is part of the TRAFAIR project [5], [6]. An EU co-funded project that brings together 9 partners from two European countries (Italy and Spain) to develop innovative and sustainable services combining air quality, weather conditions, and traffic flows data to increase the awareness on urban air quality for the benefit of citizens and government decision-makers. TRAFAIR aims to supervise the level of pollution on an urban scale in 6 European cities by producing a provision of real-time estimates of air pollution on an urban scale exploiting a network of air quality sensors and by developing a service for forecasting urban air quality based on weather forecasts and traffic flows.

The main goal of this paper is to describe our integrated traffic simulation model which would provide traffic flow within the city of Modena. Firstly some related research work are presented in section II. Secondly, we concentrate on collecting all the relevant information on the sensor network such as sensor position, type, and data provided (this is described in Section III). Thirdly, we focus on creating a road network of the city as much precise as possible, by integrating data from Open Street Map, adding and fixing information on road restrictions (described in Section IV). Then, we build a microscopic traffic simulation model using real data sets from the local traffic sensors (outlined in Section V). In the end, we perform some in-depth analysis on the output obtained by the traffic model such as an investigation of peaks and trends on the average working day, an evaluation on the most congested roads, a comparison between different simulation configurations (detailed in Section VI). Moreover, we experimented some scenario of real-time estimation of traffic flows within the city of Modena (in Section VII) in order to evaluate performance and reliability of the traffic model.

Modena is an Italian city of 186 000 inhabitants, capital of the homonymous province in the Emilia-Romagna region.

Within the TRAFAIR project, all relevant information for creating a traffic model for the city has been collected and stored in a database, the TRAFAIR DB. As shown in Figure 1, real-time traffic related information are collected from regional sensors (A) and urban sensors (C). The road network of the city is created filtering information from Open Street Map and enriched this with traffic restrictions and rules (B). Information about the position and type of traffic sensors has also been added. All these information are stored in a relational PostgreSQL database[1]. The simulation model (SUMO) queries the database to retrieve real data from sensors and implement

[1] https://www.postgresql.org/

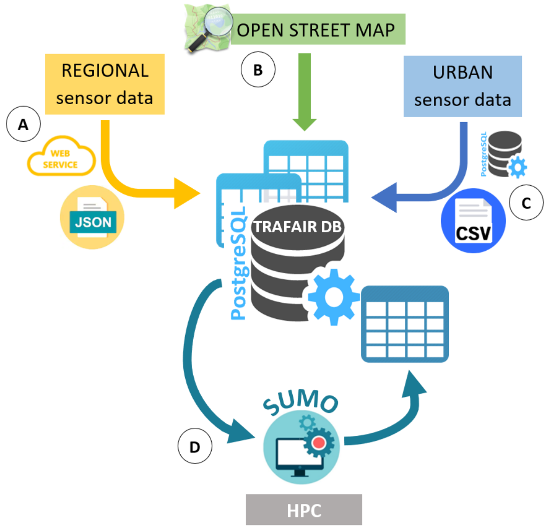

Fig. 1. Trafair data platform.

a traffic model; the simulated traffic flows all around the road network is also stored in the database (D). SUMO is executed remotely on a HPC (High Performance Computing) platform since it requires computing resources and time.

II. Related Work

Urban mobility has, in recent years, become an object of study in an attempt to improve it in order to increase citizens well-being. Smart mobility could be perceived as a set of coordinated actions addressed at improving the efficiency, the effectiveness and the environmental sustainability of cities [7]. The literature is full of paper that address this issue [8], and the European Union has defined numerous programs and funding opportunities such as European Fund for Strategic Investments, European Structural and Investment Funds, Connecting Europe Facility funds, Urban Innovative Actions, URBACT, Interreg Europe and many others.

A critical aspect that qualifies smart mobility is the use of a huge volume of data [9]. For implementing an effective mobility, these data must be up-to-date, and in many cases, real-time information. In order to obtain this information there are two main possibilities: employ crowdsourced data that monitor citizen movements, using GPS (usually taxi GPS trajectories data are employed as in [10]) and Bluetooth data from mobile phones (as in [11] and [12]) or take advantage of traffic sensor data using them as input for a traffic simulation model.

Crowdsourcing is useful to improve urban mobility, several projects rely on up-to-date information supplied by users. On the other hand, when measurements are the key elements, we should make use of a set of sensors spread in the urban context.

Google Traffic [13], INRIX [14], [15], HERE traffic API [16] and Open Transport map [17], TOMTOM [18], BING Traffic API [19], are monitoring traffic tools that can characterize urban traffic conditions using crowdsourced data.

In [17], Open transport Map is introduced, Open Transport Map is a web-based map derived from Open Street Map. The data collected are all stored in an infrastructure suitable for network analyses. This is an example of how traffic data can be stored and analysed. Traffic data used to produce the map are historical data (not real-time), but demographic data are also taken into account to obtain an annual average traffic volume for every road.

An example of traffic simulation using traffic sensors data is described in [20], here are presented three real word traffic scenarios in the city of Bologna, selected areas are small and traffic demand is evaluated starting from real sensor data, then SUMO is used to simulate traffic.

In our traffic model, we have employed a similar solution but how sensor data are used is quite different. Routes of vehicles are dynamically adapted considering the number of vehicles counted by every traffic sensor. The simulated area is bigger and covers the whole urban area of the city of Modena.

Another possibility could be to use crowdsourced data and traffic sensor data in the same model. An example of this solution is described in [21], data coming from different sources like traffic counters and floating car data are aggregated together with origin-destination matrix to provide a view of the traffic situation in the west of Austria.

We are considering to assemble data from crowdsourced resources and use them to enrich our model following this example.

III. Sensor data collection

Traffic speed, flow and density are the fundamental traffic state parameters crucial for setting up a traffic flow model. Having information about the real number of vehicles in the street allows to realize more realistic dynamic traffic models. Several types of detectors could be used: preformed loops, magnetic detectors, pressure detectors, radars, microwaves, video cameras and induction loops. In the contest of this project we are using induction loops, since the city of Modena already has them installed in several roadways. They are the most commonly used detectors present in a lot of cities. They can be placed on one or more lanes as in Figure 2 or positioned on a service lane reserved to buses and taxi. Induction loops are insulated, electrically conducting loop installed under the road surface; generally, a lead-in cable connects the loop to the detector: an electronic unit that detects the presence of vehicles above the loop as described in [22]. The inductance of a coil is calculated as:

$$L = N * B * A/I \tag{1}$$

where L is the inductance (H), N is the number of turns, B is the magnetic flux density (webers/m^2), A is a cross-sectional area of a coil (m^2) and I is the current flowing through the coil (A). When a vehicle is over an induction loop the value of the inductance is reduced due to Eddy currents as described in [23]. The Eddy currents create another magnetic field in the metal object (the vehicle passing). This induced magnetic field opposes the magnetic field of the inductive loop, and therefore reduces the magnetic flux density of the loop, which in turn reduces the inductance of the loop, as the inductance is proportional to the magnetic flux density (as is explained by Formula 1). Eddy currents induced in the vehicle decrease the inductance of the loop, therefore the resulting effect is a decrease in inductance. The electronic unit of the induction loop senses the reduction in the value of inductance as a decrease in frequency and sends a pulse to the controller. The other measure collected is the average speed of vehicles passing through the induction loop. The speed is calculated for every vehicle as the ration between the length of the induction loop and the difference between the time when the reduction of the induction loop is sensed and the time when the inductance increases because the vehicle is passed and there is no more Eddy current. This is the procedure used in dual loop sensors, single loop procedure to estimate speed is discussed in [24].

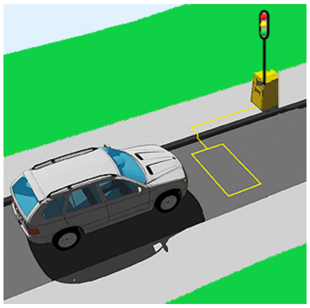

Fig. 2. Location of traffic sensors on the road surface.

In the city of Modena, there are about 400 traffic sensors and they are located into two types of roads:

- the urban traffic sensors (blue spots in Figure 3) are placed on municipality roads within the ring road near the traffic lights; the provider of their data is the Municipality of Modena[2],
- the regional traffic sensors (purple spots in Figure 3) are located on provincial and state roads, that are the main access roads to the city; the provider of this data is Lepida S.c.p.A.[3], a regional company.

All these sensors are induction loops and are able to count the number of vehicles passing through them and the average speed; in addition, the regional traffic sensors are able to identify the type of vehicle passing through the sensor and provide the number of each type of vehicle (medium and large trucks, buses, motorcycles, vans, pickup trucks, articulated lorries, cars). Each urban traffic sensor provides one measurement every minute, while the sampling rate of the regional traffic sensors is 15 minutes. Before starting the TRAFAIR project, the localization of the urban traffic sensors was stored by the Municipality only through ".PNG" images of the junctions with small rectangles indicating the sensors. After a preliminary analysis of these images, a human-supervised comparison using Google Maps was needed to understand the exact GPS coordinates of each sensor. The coordinates, the name of the street and the lane of each sensor were stored in a ".csv" file and then imported in the TRAFAIR DB. The same information related to the regional traffic sensors, instead, was provided by Lepida S.c.p.A. through a ".csv" file, which was imported in the TRAFAIR DB.

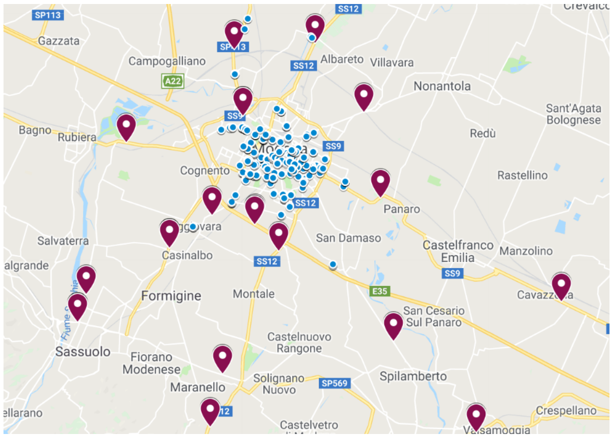

Fig. 3. Modena map with regional (purple spots) and urban (blue spots) traffic sensors (map from [25]).

All the real-time data of the traffic sensors coming from different providers are stored in the TRAFAIR DB. As shown in Figure 1, the real-time data of the traffic sensors are collected using two different procedures according to the providers of the data. The regional sensor data are queried through a web service, while the urban sensor data are first stored in a local PostgreSQL database and then copied in the TRAFAIR DB.

The regional web service is queried by using a link with the following structure:

http://195.62.186.135/sensornet/getAllHourData/measure_ID/YYYY-MM-DD%20HH:MM:SS,

where *measure_ID* is the identifier of the traffic sensor of which we want to obtain the real-time data, and *YYYY-MM-DD HH:MM:SS* is the timestamp indicating the hour for which we want to obtain the real-time data.

The web service approximates the timestamp to the hour removing any minutes and seconds and provides a JSON file[4] with four observations (one observation for every 15 minutes in the interval). Each observation contains its timestamp, the total number of the vehicles passing through the traffic sensor in the following 15 minutes starting from the timestamp, the number of each type of vehicle and the average speed of the vehicles. To obtain the real-time data, we implemented a script in Python that sends one request to the web service every 15 minutes for each regional traffic sensor. The script uses the Python module request[5] of the urllib library to open the URL of the web service and the library json[6] to decode

[2] https://www.comune.modena.it/
[3] https://www.lepida.net/

[4] An example of a JSON file is available at http://trafair.eu/lepida-json/
[5] https://docs.python.org/3.0/library/urllib.request.html
[6] https://docs.python.org/3/library/json.html

the output of the web service. Before sending the request, the script selects the timestamp of the last observation of the specified regional traffic sensor. Indeed, we implemented a table in the database in which we store the timestamp of the observation of each regional traffic sensor every time we insert a new observation. After the request, our script parses the information of the JSON file and stores it in the TRAFAIR DB. Then, the script updates the last timestamp of that traffic sensor. The 54 regional traffic sensors produce an average of 48900 observations per day.

The urban sensor data are made available through the ssh connection to the local PostgreSQL database implemented by the Municipality of Modena, that is the owner of the urban traffic sensors. Each urban traffic sensor provides one observation each minute. Each observation contains the timestamp, the total number of the vehicles in one minute starting from the timestamp and the average speed. To obtain the real-time data, we send the query to the local PostgreSQL database to select the observations starting from the latest timestamp for the urban traffic sensors in the TRAFAIR DB. Then, the result of the query is copied on the TRAFAIR DB. The 345 urban traffic sensors produce around 379204 observations every day.

IV. The road network

In order to run a simulation model, a road network of the city is needed and it can be a simplified version of the real road network containing the restrictions and the traffic signs. Within the TRAFAIR project, the road network of the city of Modena was extracted from Open Street Map[7] (OSM) [26], a collection of geographical data born as a collaborative project, and stored in the TRAFAIR DB. Data of OSM are distributed under the Open Database License. Everyone can create an account on the OSM web site to integrate or revise already available information or insert a new one. For the TRAFAIR scope, we verified that OSM contains data by the Regional Administration of Emilia Romagna provided by Geoportale E-R.

It is common to find some mistakes or missing information working on geographical data obtained from OSM. The best thing to do in these cases is to integrate OSM data with the information that is needed. It is possible both to change existing features or to add tags and properties as described in [27]. Checking information of Modena in OSM, an analysis of the road network was done to understand the completeness of OSM data in the area of interest. The most important information for our scope is number of lanes, street name, max speed allowed and roadway directions. In our case of study, the number of lanes needs to be known because induction loop sensors are located in a precise lane. We discovered that only 36.28% of OSM streets have information about lane number. Since the relevance of that information, it was manually integrated directly in OSM by looking at the pictures showing traffic light junctions and sensors position.

OSM data are XML based and are structured around three main elements:nodes, ways and relations.

Nodes are points in the space represented by their latitude and longitude and a unique identifier. Each node can contain several tags that describe the content and the meaning of the node. Tags elements describe specific features of map elements. Every tag is composed of a key and a value. Keys are employed to describe topics, categories, or types of feature and values are the corresponding numeric or textual information.

Ways elements are not always a representation of a roadway, they are an ordered list of nodes. Indeed also buildings and parking areas can be represented using the way element. Moreover, ways can have tag sub-elements which give more information about what they represent in reality.

A relation is a group of elements that are referred to as members and it is used to define a geographical relationship between them. In a relation, every member can also have a role that specifies the part it plays in it. Restrictions are a specific type of relation that represent traffic signals and obligations.

For creating the Modena road network, we make use of OSM data. OSM data can be filtered, for example, to include only one type of ways or to exclude a type of nodes. We select the polygon of interest that covers the urban area of Modena, then, we filter:

- all the ways with the highway tag (this tag ensures that the way is a road) excluding some values: 'proposed', 'pedestrian', 'cycleway', 'path', 'track', 'footway', 'steps', 'raceway' and 'service', because these roads are not affected by vehicular traffic flows,
- all the nodes in the polygon connecting these ways extracted,
- all the relations of restriction type referring to the ways and nodes above.

This information was included in the TRAFAIR DB and in the SUMO model.

Several tables in the TRAFAIR DB are used to store the Modena road network. A Python script using Overpy[8] was realized. Overpy is a Python Wrapper to access the Overpass API[9]. It allows sending a query to obtain OSM data contained in a specified polygon defined by a sequence of latitude and longitude pairs. We select the polygon of interest retrieving latitudes and longitudes and we query the Overpass API to filter the data of interest and load this data in the DB.

The SUMO road network has been created by using osmfilter[10] and Netconverter, a SUMO tool to convert the OSM map into the SUMO map. Indeed, SUMO requires maps with a structure a little different from the OSM map. SUMO map does not have ways, but edges that connect two nodes in a given direction. Edges can have one or more lanes, each lane has a proper direction, its restrictions and traffic signs that could be different from other lanes of the same edge. In OSM, there is only the information about the number of lanes in ev-

[7] www.openstreetmap.org

[8] https://pypi.org/project/overpy/

[9] http://overpass-api.de/

[10] https://wiki.openstreetmap.org/wiki/Osmfilter.

ery way, lanes are not described as a specific object. Therefore, also restrictions and traffic signs are defined between two ways and not between two lanes. Netconverter allows to specify some parameters to define several aspects of the conversion: whether to keep or not the geometry of polygons representing buildings included in OSM map, automatically join lanes that have the same permissions and do not admit lane changing or keep them separated, enable roundabout guessing where traffic signs and street geometry suggest their presence. The obtained SUMO map is a connected graph which contains all traffic signs, traffic lights and restrictions included in OSM. A lane will be connected with all the other lanes where it is allowed to drive starting from it. A vehicle can only move across connected lanes.

V. Traffic model

In this section, we describe how we have implemented a micro-simulation dynamic traffic model based on the real traffic data for the city of Modena. Traffic data is required for every point of the city. Thus, to estimate vehicle flows, a dynamic traffic model, based on induction loop sensors real measurements, was realized. Traffic models simulate the appearance of a traffic system integrating mathematical models as described in [28]. The simulations are realized by using open source resources: SUMO (Simulation of Urban Mobility) as the simulation tool and OSM as the source of geographical data.

An overview of the model structure is depicted in Figure 4.

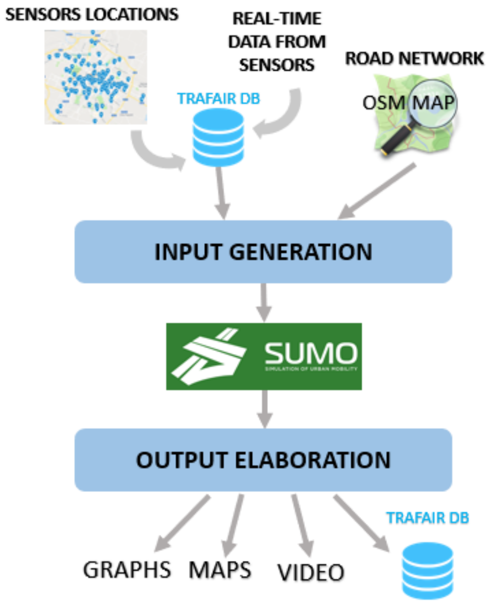

Fig. 4. Overview of the model structure.

SUMO (Simulation of Urban Mobility)[11] is an open-source, highly portable, microscopic and continuous road traffic simulation package and is designed to handle large road networks. This simulation suite has been continuously developed for more than 15 years and has been extensively successfully applied in different projects related to urban traffic management, traffic emission and other traffic issues [29]. In SUMO time is discrete and every time step of the simulation corresponds to one second in reality. SUMO gives the user the possibility to configure it in several ways including different objects and tools to adapt it to the necessities of use. As a micro-traffic model, every vehicle that moves within the simulated network is modelled individually. Instead, the macro-traffic model simulates vehicles as groups, or more precisely, flows, moving all in the same way and with the same route as a fluid in a tube. The speed and the acceleration of the vehicle are updated depending on the vehicle ahead and the traffic state of the road network. When simulating traffic, the street restrictions, such as maximum velocity and right of way rules, are regarded as described in [30]. SUMO supports a lot of different inputs: routes derived from sensor data, routes derived from an origin-destination matrix, GPS routes and also random routes. All of them can help to enrich the simulation and adapt it to the case scenario.

In order to set up the model, it is important to define some SUMO objects: calibrators and virtual induction loop detectors were inserted to compare real measurements with values obtained in the simulation. Calibrators are used to redirect vehicle routes [31]. The positions of these objects need to be the real position induction loop sensors in the streets. A calibrator acts by monitoring the number of vehicles passing through them: removing vehicles or inserting new vehicles if the flow value measured by the real traffic sensor is different from vehicle counted in the simulation. Calibrators can also modify the speed of vehicles passing if their detected speed is different from the average speed measured in reality. Every calibrator has several flow definitions, one for every fifteen minutes of simulation indicating the total number of vehicles measured in the interval and their average speed. A calibrator needs at least one vehicle flow definition, i.e. the value of vehicle counts and speed that the calibrator aspires to reach in a defined time interval. Flows can be employed to program a calibrator, they assign the aspired number of vehicles in a time interval. Calibrators have to be associated with a ”route probe” object which is used to determine the route distribution for all vehicles that passed an edge in a given time interval. As soon as the first vehicle has passed the route probe detector, the calibrator will be able to use the route of that vehicle to create new vehicles or assign a new route to the existing ones. In [32], a detailed description of the employed model can be found.

In our implementation, SUMO directly queries the TRAFAIR DB tables in which measurements from induction loop sensors are collected in real-time, then the configuration file of SUMO is produced and the SUMO simulation can start. Psycopg[12] Python library was used to query the database to obtain the sensors data in order to organize them in the

[11] https://sumo.dlr.de

[12] http://initd.org/psycopg/

required XML files. Therefore the creation of the input file for SUMO is automatized. The input files are created inside Trafair server and then, since the traffic model is executed in a remote HPC resource, these files are copied in a remote directory and the simulation is configured and started in one of the remote resource nodes. Once the simulation is ended, the obtained output files are generated in a specific directory on the remote resource. These files are then copied and synchronized on the Trafair server through an SSH procedure.

The main output produced by SUMO is an XML file that includes information about vehicle count, lane density and average speed for every road in the map and every minute of simulation. The generated output provides a possible, complete representation of the traffic in the urban area of Modena and it can be directly stored in the TRAFAIR DB. Inserting virtual induction loop detectors in the simulation, another output file is produced, containing the exact counts and speed of vehicles for every minute of the simulation and every virtual induction loop detector inserted, thus potentially in every road of Modena. Besides, other types of output can be generated, such as video, graphs and images, managing the XML file.

VI. Analysis of Traffic Flows

Through the collection of several simulation outputs, the average trend for every day of the week had been evaluated as described in [33]. Then these average trends were compared with daily simulations to identify unusual trends and detect the location of the city where road traffic was irregular. Several outputs obtained simulating November weekly days were considered to evaluate a mean value of vehicles count for every one of the 297 points, where traffic sensors are located, every fifteen minutes. As can be seen in Figure 5, the average curve shows a peak in the morning, then an increment during lunchtime, then a growth during the afternoon and evening hours followed by a strong reduction of vehicles at night.

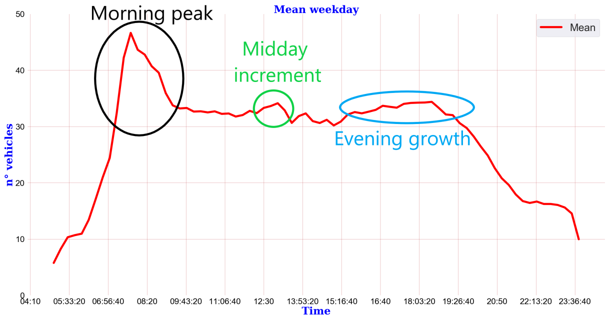

Fig. 5. The curve represents the average weekday obtained calculating an average of all weekly days in November 2018.

In order to communicate the traffic model results obtained and identify what are the areas of the city where jams are frequent during the performed simulation, additional data elaborations are needed. Output data obtained from the simulation are spatio-temporal data, their spatial dimension is important to analyse too. The more effective way to compare different edges is to show them on a map with colours indicating the value of vehicles counted. The visualization tool allowed to generate an image for every minute of simulation. This image was a reproduction of the same map used by SUMO where edges colours depend on the number of vehicles passing in the minute of simulation the image refers to, as in Figure 6. Once all the images have been saved in a directory, an Audio Video Interleave file will be generated using a Python script and OpenCv library [34]. In the video[13] is possible to see the evolution of the traffic situation during the simulated interval.

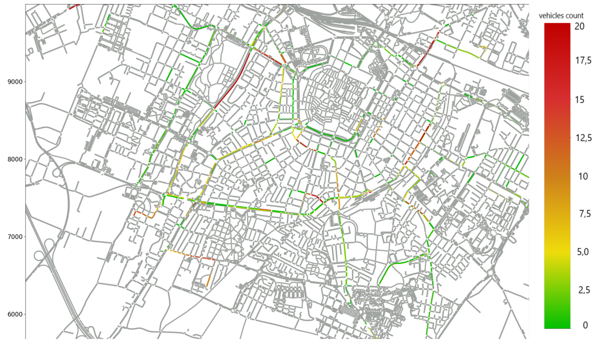

Fig. 6. Map of Modena in a November Monday at 8:00 in the morning. Colours indicate the number of vehicles in each edge and the red ones are the more populated.

The output of the traffic model includes the value of traffic density: the number of vehicles per kilometres driving through the lane every minute in the simulation. When density grows over a critical value, the flow starts to decrease and vehicles are congested. Traffic density is a good indicator of traffic estate. Since the number of vehicles that could be over a one-kilometre section of road to avoid traffic congestion is similar in different types of roads. There is a cut off value of lane density which can be interpreted as a signal of congestion. This value was estimated to be around 100 vehicles per km. A Python script was realised to return the name of the congested street, given a lane-based simulation output. This script also shows the percentage of minutes of the simulation in which the roadway was congested and lists the time instants in which the density was above the cut off value for the designed lane. Then it generates a word cloud in which name of the streets appears with a font size proportional of times in which their traffic density was above the cut-off. Examples are given in Figure 7.

VII. Towards Real-Time Traffic Modelling

In order to generate the real-time traffic model of Modena, performance analysis of SUMO has been made and the configuration to run the model has been improved to make the whole implementation fast. Although in Modena the collection of traffic data is made in real-time, the reconstruction of traffic flows within the streets is not immediate because it requires time to be computed by the simulation model. Daily simulations usually require more than 12 hours.

[13] An example of a video is available at http://trafair.eu/sumo-simulation/.

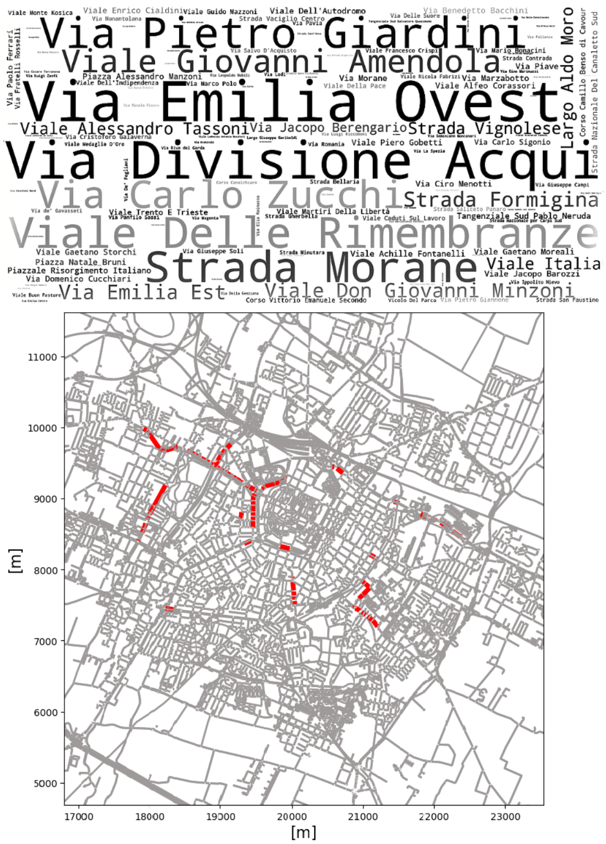

Fig. 7. Word cloud image of the streets whose lane density was above the value of 100 vehicles/km at least one time in the 19th November 2018 simulation between 05:00 and 23:59. The font dimension is proportional to the times in which the lane density value overcome the cut off during the simulation. The red coloured segments in the map are the 100 most congested roads.

Several tests show that the simulated interval is an important parameter influencing the performance and the execution time of the simulation. Where congestion was observed, calibrators were not able to follow the real flow values due to the creation of traffic jams that did not allow to insert new vehicles. In real measurements the flow was very different from the simulated flow in these points. Reducing the duration of the simulation and simulating every hour independently will help to avoid the creation of fake jams. Further studies show that splitting the daily simulation into smaller intervals and running all the obtained sub-simulations in parallel would reduce strongly the time to perform the simulation and would also improve the results that appear more similar to real measurements. These simulations have been performed on the Finis Terrae II supercomputer: a Linux based heterogeneous cluster, with an Infiniband FDR low latency network interconnecting 317 computing nodes based on Intel Xeon Haswell processors. Sub-simulations of subsequent time intervals are ran in parallel, each one in a different core. The contemporary execution of all the sub-simulations allows us to reduce the execution time required to perform a daily simulation. When the simulation is restarted traffic jams that are not caused by real flows are removed, thus shorter simulation interval improve performance. In Figure 8, daily simulations obtained dividing the same day into sub-simulations of different duration were compared. All the simulations show a similar trend coherent with real measurements and the smaller the simulated interval is, the more similar to measured values the output is.

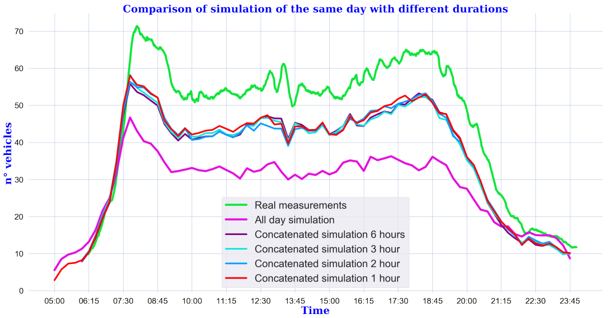

Fig. 8. A Monday simulated all in one simulation or with different simulation dividing the day in blocks of the same duration and concatenating the outputs.

As we can see in Table I, the reduction in the time required to run the simulation would also allow to obtain results more frequently and update data and maps in semi-real-time.

TABLE I
AVERAGE EXECUTION TIMES OF THE SIMULATIONS REPORTED IN FIGURE 8.

Real simulation interval	Medium time required to obtain results
19 hours	almost 13 hours
6 hours	almost 2 hours
3 hours	45 minutes
2 hours	36 minutes
1 hour	12 minutes

VIII. CONCLUSION

In this paper, we have described how we generate an integrated data platform collecting geographical and sensors data from different sources to monitor the traffic flow in the city of Modena. The traffic model we implemented uses real data from 400 traffic sensors. It is able to deduce the traffic flows within all the roads of the city, starting from punctual information. The output of the model has been used to identify which are the most congested roads and to discover roads that have a similar trend of vehicles circulating during the day.

Some important limitations and possible solutions have also been highlighted in this paper. Firstly, the simulation strongly depends on the position of the detectors. Vehicles that do not have a route which involves roads where these detectors are placed would not be counted and simulated. Even if some roadways are very crowded in reality, if no detectors are present on these roads, their traffic density would appear low. Adding known routes (for example, by exploiting Origin-Destination matrices) can increase the reliability of the simulation. Secondly, simulations of long time intervals (more than 6 hours) take a long time to generate results and, also, might create fake traffic congestion. We evaluated that if the simulated time interval is less than 3 hours, the results are produced in less than 1 hour and no fake congestion are created in the traffic flows.

As future work, we intend to integrate our platform with open crowdsourced data in order to add traffic information where no traffic sensors are available. Besides, we are exploring how to make SUMO more efficient and to parallelize some calculations on traffic flows to improve the simulation execution time.

Acknowledgment

This research has been supported by the TRAFAIR project 2017-EU-IA-0167, co-financed by the Connecting Europe Facility of the European Union. The contents of this publication are the sole responsibility of its authors and do not necessarily reflect the opinion of the European Union. The authors would like to thank in particular all the partners that contribute in the collection and management of traffic sensor data: the City of Modena and Lepida S.c.p.A..

References

[1] E. E. Agency, "Progress of eu transport sector towards its environment and climate objectives," European Environmental Agency, Tech. Rep., 2018.

[2] T. Kurczveil and P. lvarez Lpez, "eneteditor: Rapid prototyping urban traffic scenarios for sumo and evaluating their energy consumption," in *SUMO 2015 Intermodal Simulation for Intermodal Transport*, 05 2015.

[3] G. Fltterd, D. Krajzewicz, Y.-P. Wang, M. Behrisch, and P. Wagner, "Disaggregate route choice estimation for the sumo traffic microsimulator with the cadyts calibration tool for improved traffic management," *2nd NEARCTIS workshop 2009*, 2009. [Online]. Available: http://infoscience.epfl.ch/record/154995

[4] L. Bieker-Walz, D. Krajzewicz, A. Morra, C. Michelacci, and F. Cartolano, "Traffic simulation for all: A real world traffic scenario from the city of bologna," *Lecture Notes in Control and Information Sciences*, vol. 13, pp. 47–60, 01 2015.

[5] "Trafair: Understanding traffic flow to improve air quality." [Online]. Available: http://www.trafair.eu

[6] L. Po, F. Rollo, J. R. R. Viqueira, R. T. Lado, A. Bigi, J. C. Lopez, and P. Nesi, "Trafair: Understanding traffic flow to improve air quality," in *The 1st IEEE African Workshop on Smart Sustainable Cities and Communities (IEEE ASC2 2019) - In conjunction with the 5th IEEE International Smart Cities Conference, ISC2 2019, Casablanca, Morocco, October 14-17, 2019*, 2019, to appear.

[7] C. Benevolo, R. Dameri, and B. DAuria, *Smart Mobility in Smart City. Action taxonomy, ICT intensity and public benefits*. Springer, 01 2016, vol. 11, pp. 13–28.

[8] E. J. Tomaszewska and A. Florea, "Urban smart mobility in the scientific literature bibliometric analysis," *Engineering Management in Production and Services*, vol. 10, pp. 41–56, 2018.

[9] H. Jeekel, "Social sustainability and smart mobility : Exploring the relationship," *Transportation Research Procedia*, vol. 25, pp. 4296 – 4310, 2017, world Conference on Transport Research - WCTR 2016 Shanghai. 10-15 July 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2352146517305616

[10] J. Tang, F. Liu, Y. Wang, and H. Wang, "Uncovering urban human mobility from large scale taxi gps data," *Physica A: Statistical Mechanics and its Applications*, vol. 438, pp. 140 – 153, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0378437115005853

[11] J. Yuan, M. Abdel-Aty, L. Wang, J. Lee, R. Yu, and X. Wang, "Utilizing bluetooth and adaptive signal control data for real-time safety analysis on urban arterials," *Transportation Research Part C: Emerging Technologies*, vol. 97, pp. 114 – 127, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0968090X18314669

[12] M. Blogg, C. Semler, M. Hingorani, and R. Troutbeck, "Travel time and origin-destination data collection using bluetooth mac address readers," *ATRF 2010: 33rd Australasian Transport Research Forum*, 01 2010.

[13] "Google traffic." [Online]. Available: https://cloud.google.com/maps-platform/routes/

[14] "Inrix." [Online]. Available: http://inrix.com/

[15] A. Sharma, V. Ahsani, S. Rawat, I. S. U. C. for Transportation Research, Education, and N. D. of Transportation, *Evaluation of Opportunities and Challenges of Using INRIX Data for Real-time Performance Monitoring and Historical Trend Assessment: Final Report*. Nebraska Department of Transportation, 2017. [Online]. Available: https://books.google.it/books?id=-E-2wgEACAAJ

[16] "Here traffic api." [Online]. Available: https://developer.here.com/documentation/traffic/topics/what-is.html

[17] K. Jedlika, P. Hjek, V. Cada, J. Martolos, J. astn, D. Beran, F. Kolovsk, and D. Kozhukh, "Open transport map routable openstreetmap," in *2016 IST-Africa Week Conference*, May 2016, pp. 1–11.

[18] "Tomtom." [Online]. Available: https://developer.tomtom.com/traffic-api/traffic-api-documentation/traffic-flow

[19] "Bing traffic api." [Online]. Available: https://docs.microsoft.com/en-us/bingmaps/rest-services/traffic/

[20] L. Bieker-Walz, D. Krajzewicz, A. Morra, C. Michelacci, and F. Cartolano, "Traffic simulation for all: A real world traffic scenario from the city of bologna," *Lecture Notes in Control and Information Sciences*, vol. 13, pp. 47–60, 01 2015.

[21] P. Pau, K.-H. Kastner, R. Keber, and M. Samal, "Real-time traffic conditions with sumo for its austria west," in *SUMO 2015 Intermodal Simulation for Intermodal Transport*, 05 2013, pp. 63–68.

[22] ITS-teknik, "Inductive loops," https://www.its-teknik.dk/CustomerData/Files/Folders/13-datablade/183_datablad-inductive-loops-cs6.pdf, its-teknik, Tech. Rep., 2018.

[23] F. H. Administration, *Traffic Detector Handbook: Third EditionVolume I*. U.S. Department of Transportation, 2006.

[24] X.-Y. Lu, P. Varaiya, R. Horowitz, Z. Guo, and J. Palen, "Estimating traffic speed with single inductive loop event data," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2308, pp. 157–166, 12 2012.

[25] G. Maps, "Google maps my maps creation," https://www.google.com/maps/, 2019.

[26] OpenStreetMap contributors, *OSM user documentation*. [Online]. Available: https://www.openstreetmap.org/about

[27] M. Haklay and P. Weber, "Openstreetmap: User-generated street maps," *IEEE Pervasive Computing*, vol. 7, no. 4, pp. 12–18, Oct 2008.

[28] N. Nasuha and M. Rohani, "Overview of application of traffic simulation model," *MATEC Web of Conferences*, vol. 150, p. 03006, 01 2018.

[29] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker, "Recent development and applications of sumo - simulation of urban mobility," *Int. Journal on Advances in Systems and Measurements*, vol. 5, 2012.

[30] D. Krajzewicz, G. Hertkorn, C. Feld, and P. Wagner, "Sumo (simulation of urban mobility); an open-source traffic simulation," *4th Middle East Symposium on Simulation and Modelling (MESM2002)*, pp. 183–187, 01 2002.

[31] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. Wießner, "Microscopic traffic simulation using sumo," in *The 21st IEEE International Conference on Intelligent Transportation Systems*. IEEE, 2018. [Online]. Available: https://elib.dlr.de/124092/

[32] C. Bachechi and L. Po, "Implementing an urban dynamic traffic model," in *IEEE/WIC/ACM International Conference on Web Intelligence, WI'19, Thessaloniki, Greece, Oct. 14-17, 2019*, 2019, to appear.

[33] ——, "Traffic analysis in a smart city," in *Web4City, International IEEE/WIC/ACM Smart City Workshop: Web for Smart Cities - In conjunction with IEEE/WIC/ACM International Conference on Web Intelligence, WI'19, Thessaloniki, Greece, Oct. 14-17, 2019*, 2019, to appear.

[34] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.