SUMMER INTERNSHIP REPORT

1 July 2021 - 31 September 2021, 3 months

Analysis of Real and Simulation Traffic Data

Student
Vincent LABARRE
Second year, Data science specialization at EURECOM, from Télécom SudParis

Project Supervisors
Dr Motonobu KANAGAWA
Dr Paolo PAPOTTI

PhD Student
Mr Kensuke MITSUZAWA

Company supervisors
Dr Stefano BORTOLI
Dr Daniele FORONI
Dr Margherita GROSSI

September 22, 2021

ABSTRACT

To understand and improve road traffic flows, one needs to collect a large amount data the vehicles' behaviors in the road network. This data collection can be accomplished by setting sensors (such as cameras) in appropriate locations such as intersections. However, to use such traffic data in subsequent tasks (such as simulator calibration), one first needs to assess the reliability of the data, as the data may contain anomaly or bias for many reasons (such as the failure of a camera).

This reports explain our preliminary analysis made for both real and simulated traffic data. As a traffic simulator we used SUMO (Simulation for Urban MObility) version 1.9.21. Afterwards, it deals with the analysis of the real data. Then, simulator outputs are examined. Finally, the problem of collisions, and more generally the problem of ill-defined routes, are addressed.

ACKNOWLEDGEMENTS

First of all, I would like to express my special thanks to my project supervisor Dr **Motonobu KANAGAWA** for giving me the opportunity to work on this project, and for his time and guidance throughout this three-month internship.

Then, I would like to express my appreciation to Dr **Paolo PAPOTTI** for his availability and regular feedback.

I would also like to thank PhD student Mr **Kensuke MITSUZAWA** for his great and helpful collaboration throughout this work.

My grateful thanks are extended to the company supervisors Dr **Stefano BORTOLI**, Dr **Daniele FORONI** and Dr **Margherita GROSSI** for their valuable remarks.

Finally, I would like to thank **EURECOM** for accepting the internship.

Contents

List of Figures

1 Study Area

The studied area is the Bantian Station in Shenzhen (BSS), in China, for which the map is shown in Figure 1.

Figure 1: Maps of the BSS in SUMO (left) and in Google Maps (right)

The allowed speeds on the roads and the traffic light systems (tls) are two major features of the network for traffic simulation. A specific file named "*shenzhen_bantian.net.xml*" is used to define the shape of the network (roads, intersections) and the speed limits. The map can be seen as a representation of a directed graph with intersections ("junctions" in SUMO) as nodes and roads or streets as edges[1]. The tls and the allowed speeds of the BSS are displayed in the diagram in Figure 2.

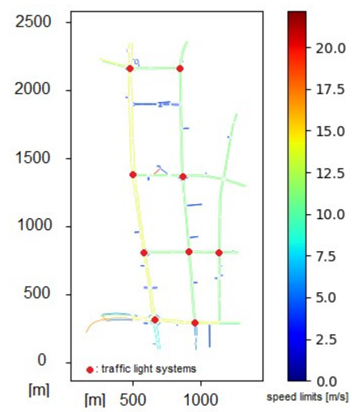

Figure 2: The speed limits and tls positions at the BSS.

The state information of the tls is programmed[2] with a file called "*shenzhen_bantian.waut.xml*". In SUMO, a vehicle is composed of three parts[3]: its type which describes its physical properties, a route that it shall take, and a vehicle ID. These properties are defined in a file entitled "*shenzhen_bantian_date.routes.xml*". Finally, a special file named "*shenzhen_bantian.det.xml*" describes the inductive-loop traffic detectors on the network that collect specific measures that we will discuss later.

In the rest of the report, two route files will be considered. The first one is the "*shenzhen_bantian_2020_03.routes.xml*" file, which describes the routes of a day in March 2020. The second one is the "*shenzhen_bantian_2020_12_22.routes.xml*" file, which describes the routes for the 22th of December, 2020.

[1] https://sumo.dlr.de/docs/Networks/SUMO_Road_Networks.html
[2] https://sumo.dlr.de/docs/Simulation/Traffic_Lights.html
[3] https://sumo.dlr.de/docs/Definition_of_Vehicles%2C_Vehicle_Types%2C_and_Routes.html

2 Analysis of the real data

To obtain the route files mentioned above, it is necessary to analyze the data collected with cameras. Some cameras are placed at 6 intersections indicated in Figure 3. These cameras are reproduced in SUMO as induction loop detectors with the "*shenzhen_bantian.det.xml*", by replacing each camera into a couple of detectors, one before the intersection and one right after. Indeed, there are no primitives to model cameras in the SUMO framework.

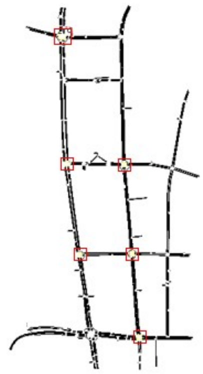

Figure 3: Position of tls with cameras (framed in red) at the BSS.

Data is reorganized in a csv file. For the rest of this section, the real data analysed is from 2020/12/22. The attributes (columns) of this data are the following: the identifier of the camera, or **stream_id**, associated with a tls (there may be multiple cameras for a single edge); the **timestamp** of the record; the monitored lane **lane_id**; the counts of the vehicles for each **direction** (left/straight/right) as detected by the camera at a specific timestamp; the counts of the **plates** recognised for the detected vehicles; and the number of cars that are placed between the camera and the stopping line of the intersection, namely **queue_num**.

We now analyze the most important information for the route data in more detail.

2.1 Timestamp

Every camera is supposed to record data every second. However, there are some timestamps for which there are no records. For example, in the "2020/12/22" real data, there is a 2-minute break in the recording data from 03:00:16 to 03:02:15, meaning that there are no records in this period of time[4].

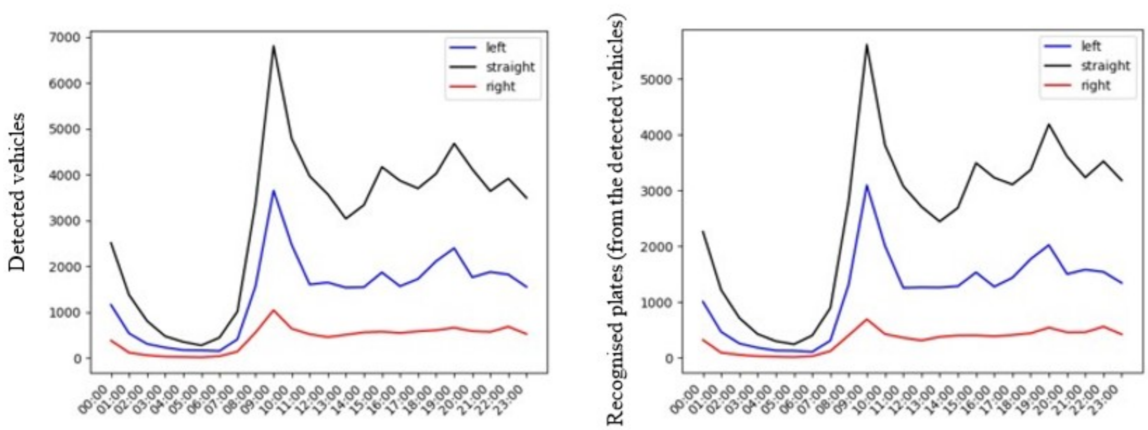

Figure 4: Number of detected vehicles (left) and recognised plates from these vehicles (right) for left, straight and right directions.

[4] To see all the missing timestamp and reproduce all the following graphs of this section, use the "analysis_raw_data_code.py" script.

2.2 Plates

We distinguish between the vehicles **detected** by the cameras and the vehicles (among the detected ones) for which the license plate has been **recognised**. In Figure 4, both the detected vehicles and recognised plates have similar frequency plots.

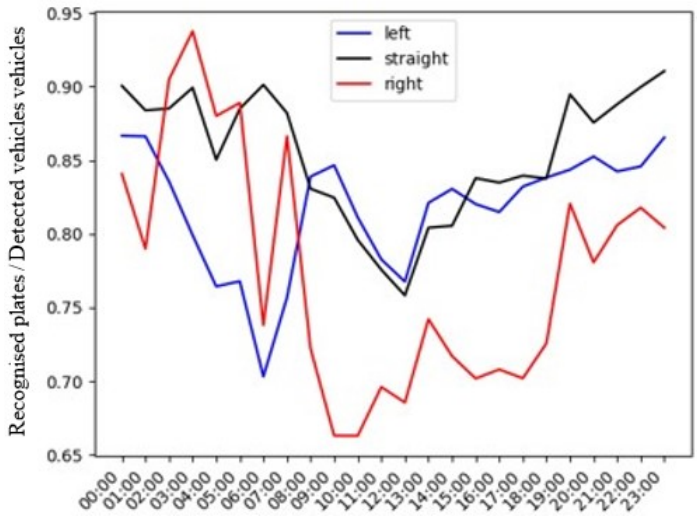

Figure 5: Ratio of recognised plates over detected vehicles for left, straight and right directions.

We report the ratio of the recognised plates over the detected vehicles over time in Figure 5. As expected, the ratio is lower than one for all cases. First, we observe that two curves are similar if their ratio curve is horizontal. In these cases they are all between 0.66 and 0.95, which is quite stable over time. Second, the ratio shows the efficiency of the recognition of plates. At the beginning and at the end of the day the recognition ratio is very good in all cases. This can be due to many reasons: the efficiency in itself of the plate recognition algorithm, the sun during the day that creates a shadow on the camera, etc).

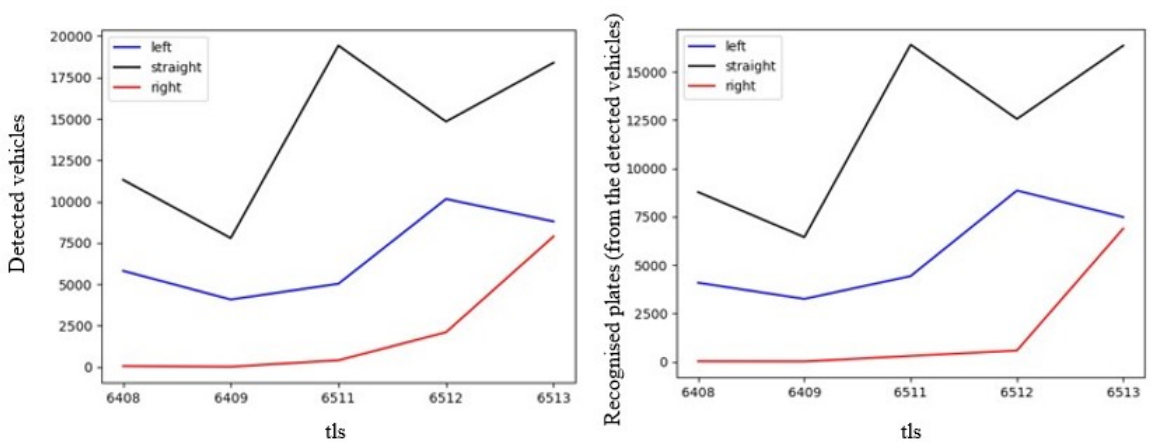

Figure 6: Detected vehicles (left) and recognised plates from these vehicles (right) for the left, straight and right directions at each tls.

The number of detected vehicles and recognized plates for each tls is reported in Figure 6. The shapes (especially for the right direction) look less similar than with the timestamp.

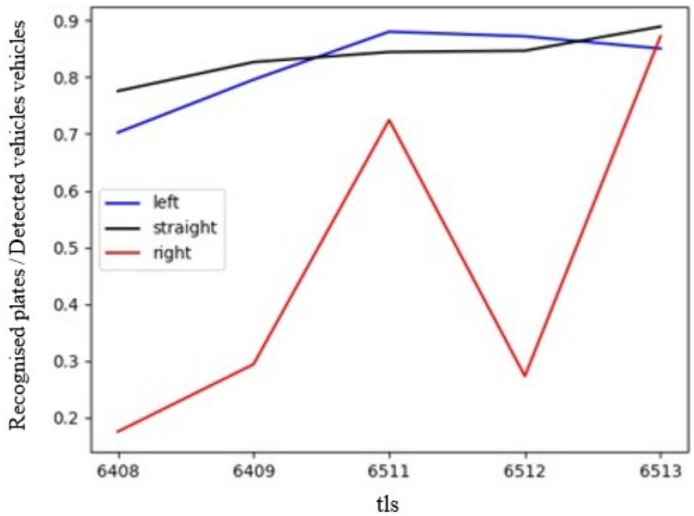

Figure 7: Ratio of recognised plates over detected vehicles for the left, straight, and right directions at each tls.

The ratio reported in Figure 7 confirms the impression: the ratios for the left and straight directions are quite similar, while the ratios for the right direction are quite different.

2.3 Queue number

The **queue_num** is the number of vehicles present between the camera and the intersection stopping line. Data is not perfect in these cases, as the same vehicle can be detected several times by the same camera. However, the sum of the queue number values at each timestamp can be used as a proxy to know how much vehicles are in the BSS at a given timestamp. This number enables an estimate of the number of vehicles running in the BSS during the simulation.

Queue numbers are also useful to enable comparisons between real and simulated data. As for the previous discussion, it is possible to observe if frequency curves are about the same shape or not, meaning if the simulation is consistent with the real data.

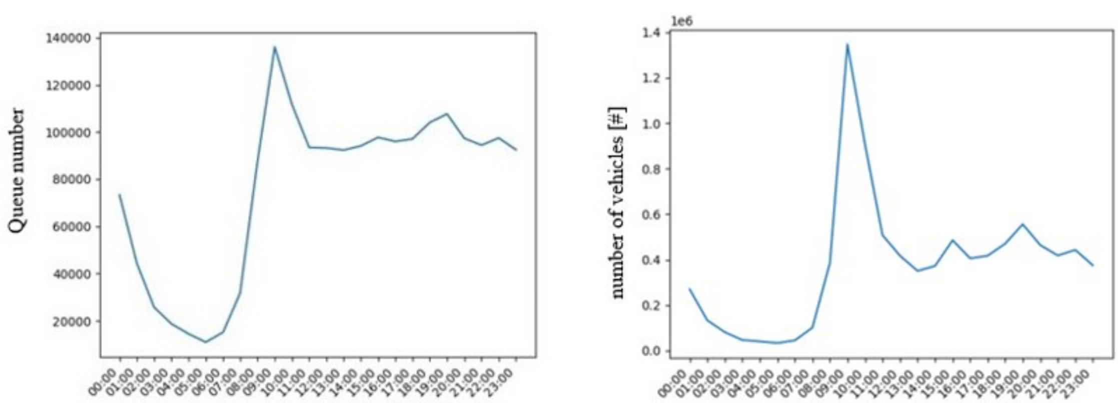

Figure 8: The queue number from the real data (left) and the number of vehicles running in the simulation (right).

Consider the plots in Figure 8. The numbers for the y-axis are not the same. In fact, there is a factor 10 between the aggregated values per hour of both figures. But as we said above, there are not the same metric but the queue number can be used as a proxy for the number of running vehicles in the BSS.

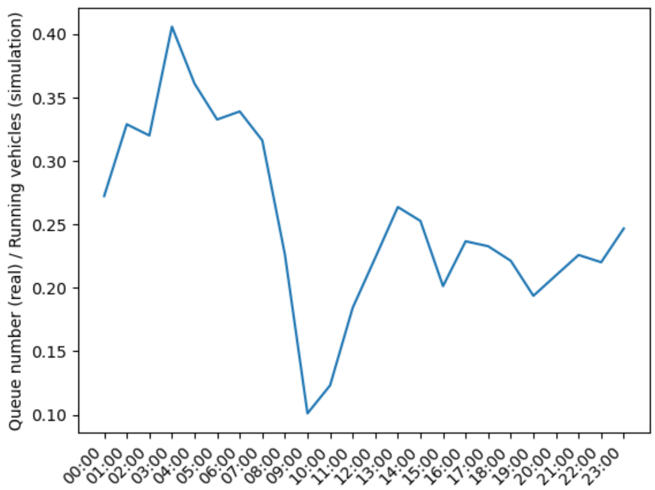

Figure 9: Ratio of the queue number from the real data over the number of vehicles running in the simulation.

We compare these 2 curves by making the ratio as we did in section 3.2 for the plates to see if the curves are proportional (horizontal line) or not. Here, all the values except two are between 0.20 and 0.35, which is fairly stable over time.

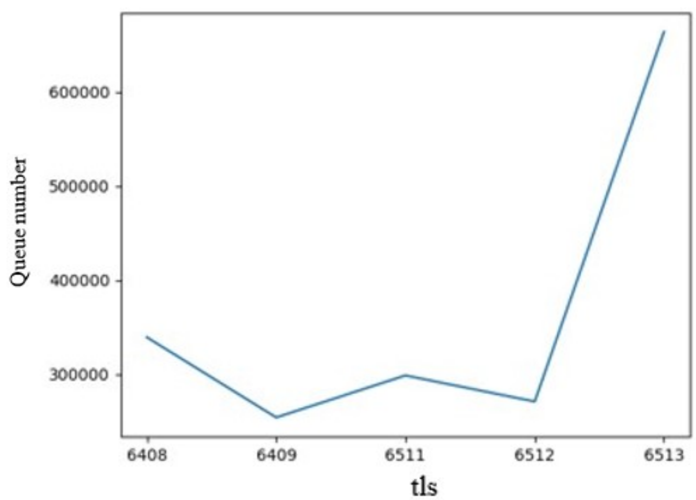

Figure 10: The queue number from the real data at each tls.

We compare the values for the queue numbers to observe whether or not there are big differences in the traffic on these intersections, as reported in Figure 10. They have about the same values except for the tls "6513", which has the double of their number. Indeed, this tls is the one at the top left of the BSS (see Figure 1 in Section 2), which is a popular highway.

3 Observation for the simulation data

To analyse the quality and the efficiency of the simulations, there are several output XML files with different metrics[5].

Categories	Attributes	Numbers
Vehicles		
	Loaded	105055
	Inserted	105055
	Running	0
	Waiting	0
Teleports		
	Total	8
	Jam	0
	Wrong Lane	0
	Yield	0
Safety		
	Collisions	8
	Emergency Stops	0
Vehicle Trip Statistics		
avg	Route Length (m)	639.25
avg	Speed (m/s)	7.14
avg	Duration	0:01:38
avg	Waiting Time	0:00:32
avg	Time Loss	0:00:46
avg	Depart Delay	0:01:11
avg	Depart Delay Waiting	-1.00
	Total Depart Delay	86 days, 10:02:46
	Total Travel Time	119 days, 9:37:48

Figure 11: Results in the statistic file ("2020/03/??" route file).

3.1 Statistic file

The *statistic file*[6] is a small file ($\approx 5KB$) containing the overall statistics of the simulation. An example of the results is reported in Figure 11[7].

For the "**Vehicles**" category, the **loaded** attribute reports the number of vehicles that were loaded from input files up to the end of the simulation; in other words, it is the theoretical number of vehicles that should be running into the network (according to the input file)[8]. The **Inserted** attribute is the number of vehicles actually inserted into the network at the end of the simulation. But some constraints must be respected so that the vehicle can be inserted into the network: it must not intersect with other vehicles, must be at a safe distance of them and must be able to brake for all situations that require it (upcoming non-prioritized intersection, scheduled stops)[9].

Running is the number of vehicles that are still running at the end of the simulation and **Waiting** attribute is the number of vehicles that could not be inserted into the network at the end of the simulation because of the constraints mentioned above.

The **Teleports** category indicates the total number of vehicles that were teleported. These teleportations can occur for the following causes:

- **collisions**, meaning the minGap[10] of the vehicle is not respected;

[5] https://sumo.dlr.de/docs/Simulation/Output/index.html
[6] https://sumo.dlr.de/docs/Simulation/Output/StatisticOutput.html
[7] This table is produced with the "statistic_file_output_table" script.
[8] https://sumo.dlr.de/docs/Simulation/Output/Summary.htmlgenerated_output.
[9] https://sumo.dlr.de/docs/Simulation/VehicleInsertion.html.
[10] **minGap**: Empty space after the "leader" vehicle (the vehicle ahead). In other words, the minimum distance allowed to the vehicle in front (2.5 m by default).

- **timeout**, that is the number of seconds[11] during which the car could not move because of a **jam** (no space on the lane to keep driving), a **wrong lane** (cannot continue to drive on the next correct lane), or a **yield** (an intersection that cannot be crossed because of the lack of priority).

Emergency Stops refers to certain sudden and unexpected events such as very abrupt braking. In most cases it is a sign of bugs or issues in the simulation settings (more precisely in the car-following model)[12].

Route Length, **Speed**, **Duration**, **Waiting Time** (the time spent by the vehicle to stand involuntarily), **Time Loss** (the time lost by driving/walking slower than desired; it includes the Waiting Time), **Depart Delay** (the time the departure of the vehicle has been delayed), and **Depart Delay Waiting** (the waiting time for the vehicle to be inserted at the end of the simulation; -1 means that there were no vehicles waiting for) are given as averages. The **Total Depart Delay** and the **Total Travel Time** are respectively the sum of the depart delay and the duration of all the vehicles that were running in the simulation.

3.2 Summary file

The *summary file*[13] is bigger than the previous output file ($\approx 25MB$). It contains the number of vehicles **loaded**, **inserted**, **running**, **waiting** to be inserted, or that have **reached** their destination with their **mean waiting time**, **travel time**, and **speed** for each second of the simulation time.

There is a difference for the vehicles that have actually reached their destination ("**arrived**)" and those that have "**ended**" their route either because they have reached their destination ("**arrived**") or because they have been removed from the simulation due to traffic problems. These 2 notions are related by the formula :

$$Ended_{vehicle\_route} = Arrived_{vehicle\_route} + Deleted_{vehicle\_route}$$

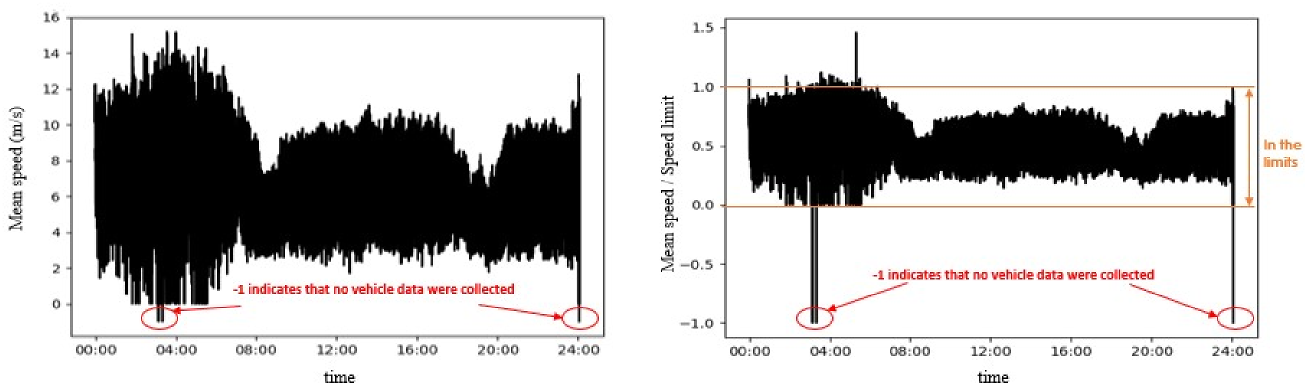

Figure 12: The mean speed (left) and the ratio mean speed over speed limit (right) of all the vehicles during the whole simulation (for the "2020/03/??" route file)

We visualize is the mean speed of all the vehicles during the simulation and the ratio between this mean speed and the authorized speed limit in Figure 12. The ratio allows us to observe if the vehicles globally respect the speed limits or not and in which proportions. When the value of this ratio is between 0 and 1, it means that the users respect the speed limits. When the value is above 1, they tend to drive faster than the authorized speed limit. A negative value for this ratio is meaningless: the only possible negative value is -1 when vehicle data is not available because of problems in the simulation or simply because no vehicles are driving during the given period. Here, in the case of the simulation with the "2020/03/??" route file, the speed limits are generally respected, with only one big peak at around 5 a.m, possibly because there are much less vehicles on the roads at this time, encouraging drivers to accelerate.

[11] 300 seconds by default.
[12] https://sourceforge.net/p/sumo/mailman/message/30762126/
[13] https://sumo.dlr.de/docs/Simulation/Output/Summary.html

3.3 Trip-Info file

The *trip info file*[14] is of the same magnitude in terms of size as the summary file ($\approx 60MB$). It contains the information about the **departure** and **arrival** of each vehicle (planned[15] and actual date, location, speed). These data is given when the vehicles arrive at their destination.

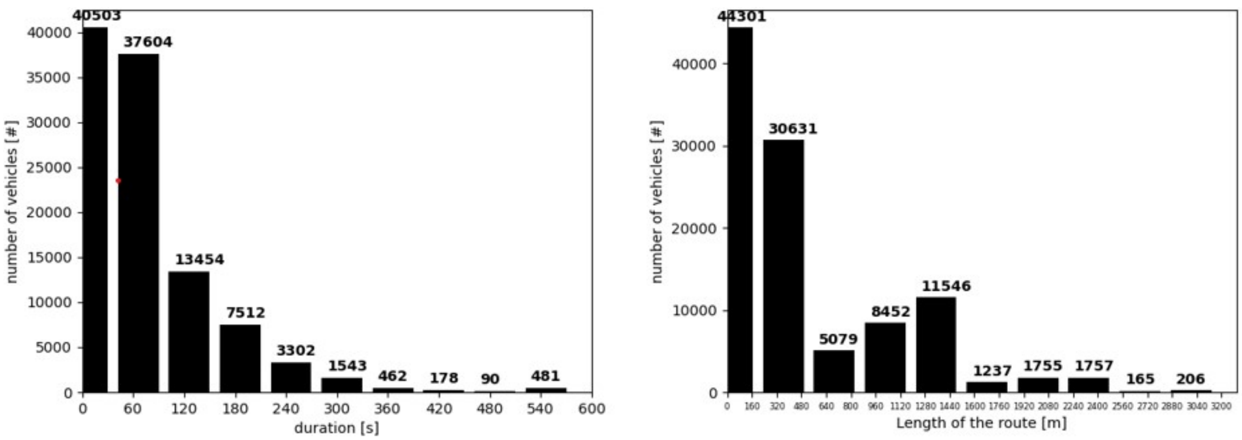

Figure 13: The duration (left) and route length (right) distribution of the car trips ("2020/03/??" route file).

With these information, it is possible to visualize the duration and route length distribution of the vehicle journey during the entire simulation, as displayed in Figure 13[16]. The graphs show that the majority of trips are short in time and distance.

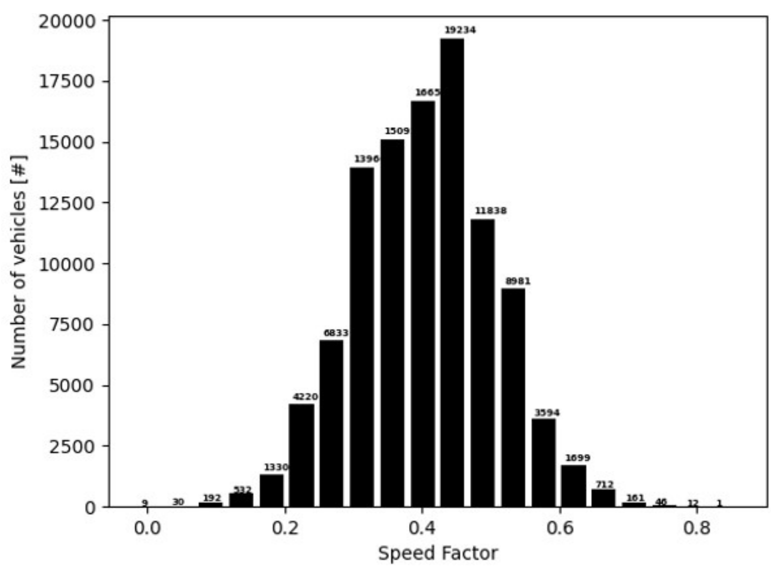

Figure 14: The SpeedFactor distribution of the vehicles ("2020/03/??" route file).

An important parameter in the vehicle trip is the **speedFactor**[17]. This individual speed multiplier is drawn randomly from a Gaussian distribution with, by default, a mean of 1.0 and a deviation of 0.1, as shown in Figure 14. It enables the vehicle to exceed the speed limit of a road, which is otherwise very rare to occur because vehicles in SUMO always

[14] https://sumo.dlr.de/docs/Simulation/Output/TripInfo.html
[15] Only the departure is planned, but it can be delayed.
[16] These figures are obtained with a modified version of the "plot_tripinfo_distributions.py" built-in function of SUMO. This modified script is visible in the "plot_tripinfo_distributions_modified.py" file.
[17] https://sumo.dlr.de/docs/Simulation/VehicleSpeed.html

tend to slow down to stay within these speed limits as much as possible. In fact, when they can ride without constraint (by other vehicles, for example), they accelerate until reaching the speed:

$$min(maxSpeed, speedFactor * speedLimit)$$

The **maxSpeed** parameter models the maximum speed at which a vehicle can circulate. It can be thought of as the maximum speed of the engine or the maximum speed desired by the driver under any circumstances.

3.4 Detector Output file

The *detector output file*[18] is bigger than the previous output files ($\approx 180MB$). It contains all the detectors with their position (on which lane and where on this lane) and their data collection frequency. They can be generated automatically with a SUMO command line[19], but in the current case they are obtained with an ad-hoc algorithm. These output files are useful to apply algorithms to measure the dissimilarity between different simulations[20].

3.5 Floating Car Data (FCD) file

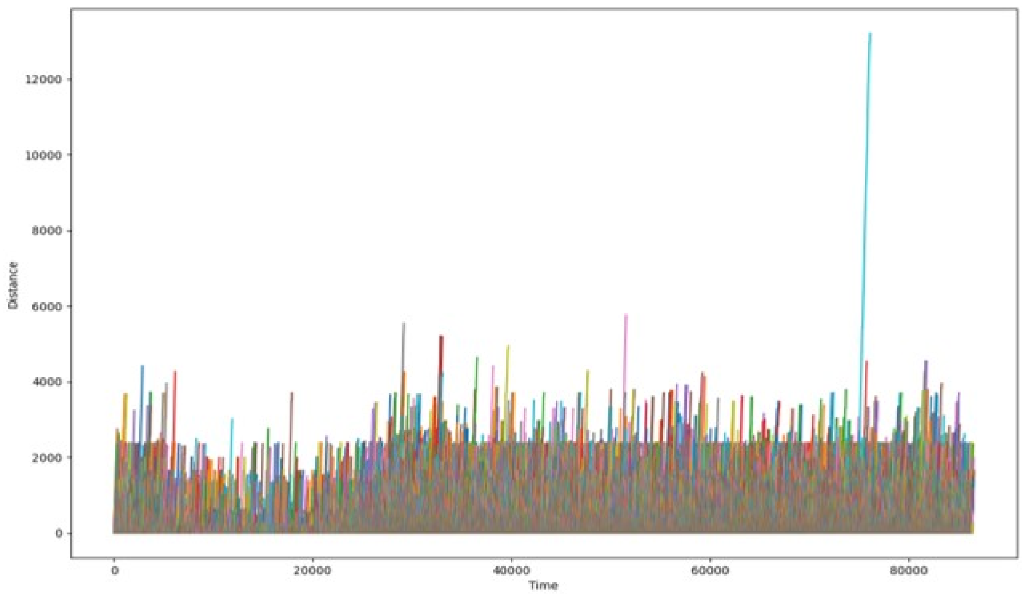

Figure 15: All vehicle distances traveled during the simulation ("2020/03/??" route file).

The *FCD file*[21] is much bigger than the previous files ($\approx 2GB$). It records the outputs (location, speed) as an accurate GPS device for every vehicle at every time step. It can be used to display the total distance traveled by all cars at a given time as in Figure 15. A python library (SumoNetVis) produces Sumo network files or trajectories on the network, but it seems to be deprecated[22].

3.6 Netstate Dump file

The *netstate dump file*[23] is of the same size as the FCD file ($\approx 2GB$). It contains detailed information for every edge, vehicle, and simulation step. In fact, it is very similar to the FCD Output, but the latter does not group vehicles on edges.

There were some SUMO embedded functions to make visualizations with this output file, but they are now outdated[24].

[18] https://sumo.dlr.de/docs/Simulation/Output/Induction_Loops_Detectors_%28E1%29.html
[19] https://sumo.dlr.de/docs/Tools/Output.htmlgeneratetlse1detectorspy.
[20] See our report *Discrepancy of Traffic Simulations* for more details.
[21] https://sumo.dlr.de/docs/Simulation/Output/FCDOutput.html
[22] https://pypi.org/project/SumoNetVis/
[23] https://sumo.dlr.de/docs/Simulation/Output/RawDump.html
[24] https://sumo.dlr.de/docs/Tools/Visualization.htmloutdated

3.7 Full Output file

The *full output file*[25] is a large file ($\approx 50GB$), containing the information about every edge, lane, vehicle, and traffic light for each time step. It gathers in a single file all the information related to the simulation; it allows analysis of the data without using many commands to obtain and combine the other output files. However, its very large size remains a hindrance to its use.

4 Discovery of incorrect routes by collisions

collider	collider_Speed (m/s)	collider_Type
flow___300_45630_28035_71704355_42318296.5	8.14	DEFAULT_VEHTYPE
flow___43500_41530_22244_41611559_42318296.2	7.27	DEFAULT_VEHTYPE
flow___51900_56148_10021_19016355_42318296.2	6.61	DEFAULT_VEHTYPE
flow___53700_39280_16279_41611559_45006190.4	8.52	DEFAULT_VEHTYPE
flow___58500_82170_51962_19016355_45006190.5	5.7	DEFAULT_VEHTYPE
flow___61800_21051_41519_41611559_15724472.25	8.98	DEFAULT_VEHTYPE
flow___62400_66107_33091_19016355_45006190.0	7.62	DEFAULT_VEHTYPE
flow___70800_28249_13027_41611559_45006190.0	6.22	DEFAULT_VEHTYPE

Figure 16: Summary of the collision information for the colliders.

As reported in Figure 11 (Section 4.1), there are object collisions during the simulation. They can be observed in the terminal as warnings during the simulation or they can be listed in an output file called "*collisionOutput.xml*". Examples of collision tables are shown in Figures 16–18[26]. The first table summarizes the name of the vehicle (**collider**) responsible for the collision, its type (**collider_Type**) and its speed (**collider_Speed**).

[25] https://sumo.dlr.de/docs/Simulation/Output/FullOutput.html

[26] Tables produced from the "collision_output_table.py" script.

victim	victim_Speed (m/s)	victim_Type
flow__300_82167_94070_19016355_42318296.5	9.98	DEFAULT_VEHTYPE
flow__43800_38329_11322_19016355_42318296.2	10.14	DEFAULT_VEHTYPE
flow__51900_97154_33550_19016355_45006190.1	9.5	DEFAULT_VEHTYPE
flow__54000_93905_60630_19016355_42318296.0	11.0	DEFAULT_VEHTYPE
flow__58500_52575_13425_19016355_42318296.2	9.78	DEFAULT_VEHTYPE
flow__61800_21051_41519_41611559_15724472.24	12.54	DEFAULT_VEHTYPE
flow__62400_40790_80615_19016355_42318296.8	9.35	DEFAULT_VEHTYPE
flow__70800_45270_57294_19016355_42318296.4	9.37	DEFAULT_VEHTYPE

Figure 17: Summary of the collision information for the victims (vehicle hits).

The second table summarizes the name of the vehicle hit (**victim**), its type (**victim_Type**), and its speed (**victim_Speed**). We notice that the speed of the collider is slower than that of the victim. However, one would expect that the collision is happening because the collider was driving too rapidly and therefore did not have the time to brake[27]. However, in the simulation the collisions are happening at a junction: the typical collision situation is that the victim went across the junction and the collider did not remark it. Because of the (relative) high speed of the victim, the collider could not brake or stop at time[28].

lane	pos	time	type
:763080_4_0	11.16	0:10:34	junction
:763080_4_0	9.8	12:13:34	junction
:763080_4_0	7.26	14:28:34	junction
:763080_4_0	6.41	15:03:06	junction
:763080_4_0	5.28	16:19:34	junction
:761467_0_0	2.52	17:13:40	junction
:763080_4_0	9.27	17:25:34	junction
:763080_4_0	7.07	19:43:34	junction

Figure 18: Summary of the overall collision information.

The third table reports the **lane** on which the collision took place, the position of the collision along the lane (**pos**), the **time** of the collision, and its **type**. In the table, collisions occurred almost exclusively on the "763080_4_0" lane. Thus, this junction has to be visualized in order to be inspected.

[27] An example of this kind of collisions: `https://www.youtube.com/watch?v=uBZOdEXY5ec&list=WL&index=53&t=20s`

[28] An example of this kind of collisions: `https://www.youtube.com/watch?v=-Q7YM8llivU&t=55s`

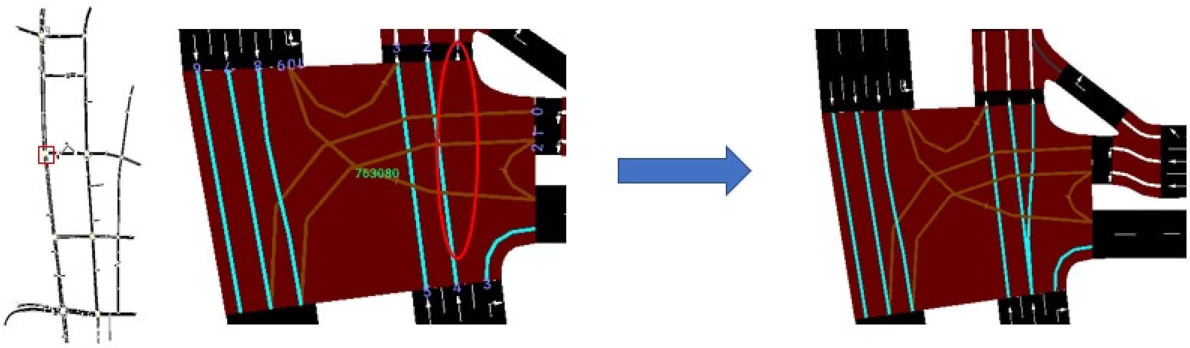

Figure 19: Position and map of the "763080" junction (left) and its map after the correction circled in red (right).

After visualization, a problem becomes apparent. A road is missing to link the lane n°4 (at the bottom right of Figure 19) to the lane n°1 (at the top right of Figure 19). Adding a road to connect these 2 lanes solves the collisions issue at this junction.

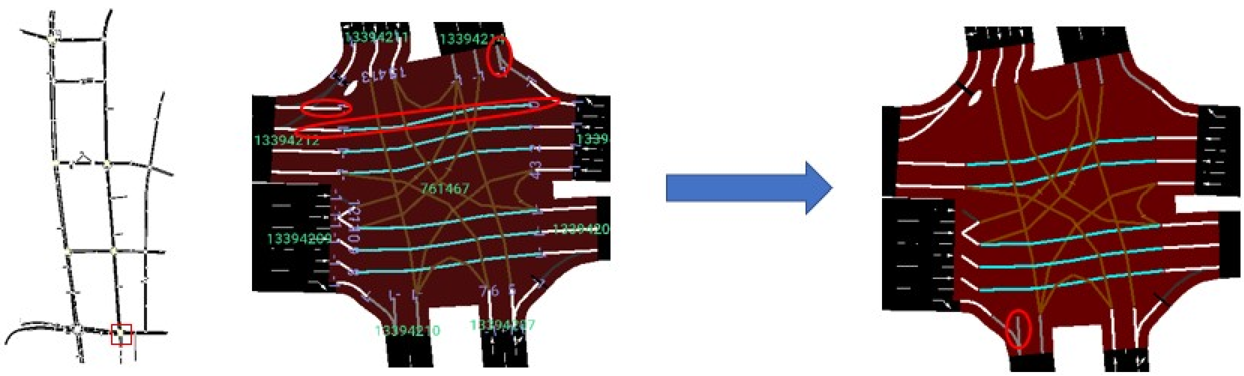

Figure 20: Position and map of the "761467" junction (left) and its map after the correction circled in red (right).

More collisions occurred at junction 761467. More precisely, it happened on its internal lane "0" (blue lane in Figure 20). This is located towards the top of the junction and it has one of its ends (the right) that is not connected to any road and cannot be connected to the nearest one because they do not go in the same direction. It is then necessary to remove it, as well as the other two roads that are not connected to any other road at the top of the junction. By doing this, there are no more collisions at this junction, nor at any other. However, it still has routes that are not connected to others (e.g. bottom left of Figure 20).

The analysis of the collisions seems to be a valid tool to detect issues on the road network, such as unconnected routes.

5 Conclusion

The analyses of the real and simulation data from the Bantian Station in Shenzhen (China) show the difficulty of collecting good data in a real road traffic environment and the complexity of modelling it in a traffic simulator, SUMO in our case. We used the real measure "queue number" as a proxy to assess the consistency of real vehicle traffic with that of the simulator. We detailed the different possible outputs of the simulator which enable us to have a good understanding of the simulation. We investigated the problems of poor road definition and collisions by finding a way to correct them by hand.
The next step would be to try to solve these problems automatically, as well as to estimate the risks of their occurrence.