A Dynamic Generator for Machine Learning Training for Traffic Management Systems

Florin ANDREESCU
Bucharest University of Economic Studies, Romania
andreescuflorin19@stud.ase.ro

This study introduces the concept of a Dynamic Generator for Machine Learning (ML) training. The paper presents the use of this key concept in order to obtain valuable data about the traffic in a crowded city like Bucharest according to the city habits and to the real-time data collected. The trained ML models may be used in the architecture of a modern Traffic Management System (TMS) based on Artificial Intelligence (AI) for Bucharest or any other crowded city. The Generator consists of a Simulator and a Collector of real data from the observed environment. In this way, the obtained set of data is hybrid because it may contain real data obtained from sensors together with synthetic data obtained from the Simulator. The Generator is dynamic because the synthetic data are smoothly replaced with real data as time goes by and as more and more sensors are put into operation in the city. The Simulator results are produced according to a simplified city model and some socio-compartmental parameters which try to describe at the macro level the behavior of the city based on historical data collected in time by the authorities and other nongovernmental organizations. These initial parameters could be improved, and more than that, the city's behavior is changing over time. That's why the city model and the parameters have to be refined in time. In this way, each new data generation will serve to re-train more accurately the set of ML models needed in the Traffic Management System.

Keywords: *Informatics management system, Simulation, Artificial Intelligence, Machine Learning, Traffic*
DOI: 10.24818/issn14531305/26.4.2022.05

1 Introduction

The concept exposed here is part of a complex research project whose general objective is to define an Information System for Urban Traffic Management based on AI.

Like stated in [1], this type of management is a complex objective that aims at both the optimization of flows and the optimal response to disturbances and crisis situations. This involves the cooperation of several intelligent systems such as Real-time data collection systems, IoT, Prediction systems, Decision/Execution systems, and Systems for interaction with actuators in intersections.

An important component of this ensemble is the Predictive Core of traffic levels at city intersections.

Book [2] describes a concept similar to the General Objective of this project and details the role of the components involved, especially the prediction part and the relationship between prediction, AI, and traffic actuators. The main idea highlighted is that smart systems need to solve traffic problems by simulating the current behavior of the Traffic Police.

At the current stage, in near the total absence of concrete real-time data from the real world related to traffic, it was necessary to create the Historical Data Dynamic Generator (HDDG) to generate the sufficient data needed to train the ML components of the main system. In the beginning, Prediction Core will be the main beneficiary of the generated data.

HDDG consists of two components: General Traffic Simulator (GTRSIM) and Real Data Integrator (RDI)

HDDG aims at this moment to reproduce with a reasonable degree of approximation the situation in a city according to the "Mirror" concept presented in [3].

HDDG is used in this project to generate traffic data over time intervals of more than five years.

In general, the generator HDDG can in turn be improved for higher levels of accuracy both by integrating new sources of reliable real data and by increasing the degree of refinement of the simulator GTRSIM.

The advantage of this approach is that everything can be developed gradually: very few sources of real data are usually available during the launch phase (behavioral data from the authorities and archival meteorological data). The rest of the necessary data must be provided by GTRSIM.

Due to the inherent difficulties (organizational and technological) related to the integration of real data sources, the GTRSIM simulator will have a very important role, especially in the first years of operation for the Prediction Core.

HDDG will also be able to be used by institutions that have roles in city development through scenario analyzes, prior to the approval of urban planning projects.

Compared to the conceptual level of the initial project, the real implementation of HDDG will inherently lead to higher consumption of processing resources with an impact on simulation costs. The institutions that will use this generator will be able to adjust from the general parameters the degree of refinement and implicitly the costs for the simulation to the desired acceptable cost levels, in balance with the desired degree of accuracy.

2 Content details

GTRSIM presentation

GTRSIM is a general-purpose traffic simulator based on city behavior. It fits into the class of "computational tools to obtain simulations for equations which need stochastic, rather than deterministic, methods" [4].

For the **Static Simulation Scenario** (one step for the whole city), **t**he city is divided into sections according to a reasonable grid. A district can contain one or more sections. The sections inherit the coefficients of population density (PD), breaking (BR), and magnetism (MG) characteristics of the districts they belong to. These coefficients for districts are input data taken from the statistics collected by the authorities.

From the total population of the city, a part of the inhabitants is selected who become "candidates" for the respective day according to the p Candidates system parameter (default = 90%). The rest of the "non-candidates" are residents who will not leave the house on that day.

General **GTRSIM parameters** are:

1. City population (POP)
2. Average number of entries/exits in/out of the city per day (NMIO)
3. City map. The map is superimposed on a grid of equal-sized sections that constitute the simulator's internal coordinate system.

The defined size of the sections for a given simulation has a direct impact on the computational power required for the GTRSIM simulator.

Each section contains a 10x10 grid. Its purpose is to randomly disperse the coordinates of the roads inside the section.

4. List of Sectors - Percentage of the city's population on each sector (Density_population_sector matrix)

5. List of Districts

Several districts make up a sector (according to data provided by the authorities)

The default population density in the district is the population/area ratio (Population_ density_district matrix). The matrix of densities by districts is generated in such a way that it fully respects the matrix of densities by sectors. The latter is given by official data.

6. Districts are grouped into District Classes depending on their specifics (residential and business, predominantly residential, predominantly business, congestion, GATE entry-exit point). A district can occupy one or more sections in the grid.

All sections of a district inherit from it the following parameters: Population_density_default, Breaking default, Magnetic intensity default (PD_default, BR_default, MG_default)

To describe the parameters that influence the stochastic generation, matrices, and vectors were used in which notes from 1 to 10 with the generic meaning "weight" were entered on each cell. In most GTRSIM components, the granulation from 1 to 10 allows the generation of random events after this grade by using parallelized weighted_random_generators [5].

The **Population_density**_section_default matrix (PD) allows the calculation of the "present" population at the beginning of day zero in each section.

The actual_section_population_density is variable and depends on the number of departures or arrivals during a day, as well as the population transiting that section. From one day to the next, this actual density can also be influenced by the following phenomena: people who do not return home until midnight, people who entered the city via a GATE and remain in a section of the city at the end of the daily route.

The Population_density_section_coefficient matrix (TPD) amplifies the PD_default by sections depending on the hours and the class of the neighborhood in which the section is located.

The **Braking_**default matrix (BR) influences the roads that pass through the sections at a given time. The Braking Coefficient matrix (TBR) amplifies the BR_default on sections depending on the hours and the district class in which the section is located.

The **Magnetic_intensity** matrix (MG) is the intensity with which the sections "attract" the roads towards them. The actual_magnetic_intensity is strongly influenced by the time of day. (eg. Business centers attract strongly in the morning and residential districts attract more in the afternoon and evening). The Magnetic_intensity_coefficient matrix (TMG) amplifies the MG_default on sections depending on the hours and the class of the district in which the section is located.

Six **types of transport** were defined: AUTO, Metro, STB, TAXI, Bicycles, Pedestrians

For each of these, the following parameters are defined: average speed default, probability of generation default, and cost default.

They are also other computed hourly dependent coefficients used during the simulation, like:

- Sensitivity to braking. It is composed of the braking coefficient in each section that a road crosses and influences the real average speed in that section for each type of transport at a certain time. It is important for AUTO, STB, and TAXI. The metro is not affected at all. Bicycles and pedestrians are influenced to the same extent.
- Sensitivity to direction. It is composed of the braking coefficient in each section and influences the real average speed in that section for certain types of transport at a certain time. It becomes important for pedestrians and bicycles (the length and frequency of pedestrian and bicycle roads decrease in proportion to the headwind).

The main correlations are "artificially injected" generation principles in the GTRSIM simulator.

The initial objective of the generation is to establish the degree of accuracy with which the ML models trained after the simulation manage to adapt to these correlations in order to make predictions as correct as possible.

The main correlations of GTRSIM are:

- Day/night cycle (dependency on the hourly interval of the day) by section classes: population densities, braking, and magnetism in certain time intervals.
- The cycle of days of the week (dependency on the day of the week): magnetism amplification in InOut Points on Friday, population density amplification in InOut points on Sunday.
- Legal holidays (dependency on the day of the year): magnetism amplification in InOut points one day before legal holidays, density amplification of InOut Point on the last day of legal holidays.
- Vacation periods (dependency on the day of the year): magnetism amplification in InOut points one day (or week) before the vacation period, density amplification of InOut points on the day (or week) before the end of the vacation period.
- The cycle of years - the share of AUTO increases by 1% per year.
- Temperature, precipitation, wind speed, and wind direction (dependency on the hourly interval of the day [6]): the

weather-temperature table is applied over the distribution of types of transport.

The direction of the wind only affects bicycles and pedestrians according to the following principle: if the road being taken is in the opposite direction of the wind, then the cost of the road increases, and above a certain value that road is abandoned. In this case, GTRSIM will allocate another road with another type of transport for the respective person.

Table 1. Example of definition for transport types: table TT

	Transport type	weight	resulted generation probabiliy	average speed-default (km/h)	cost
1	AUTO	10	26%	40	6
2	Metro	8	21%	50	4
3	STB	9	23%	30	3
4	TAXI	7	18%	35	8
5	Bicycles	1	3%	10	1
6	Pedestrians	4	10%	4	1
	Total:	**39**	**100%**		

Table 2. Legal holidays for one year

Day of the year	Date
1	1.01
2	2.01
24	24.01
116	26.04
118	28.04
119	29.04
121	1.05
152	1.06
167	16.06
168	17.06
227	15.08
273	30.11
335	1.12
359	25.12
360	26.12

Table 3. Days of the year for the vacancy intervals

Interval	Day of the year	
01.01 - 13.01	1	13
02.02 - 10.02	33	41
20.04 - 05.05	110	125
15.06 - 15.09	166	258
27.10 - 04.11	300	101
22.12 - 31.12	356	365

Table 4. Influence of temperature on the choice of the type of transport

	TEMPERATURE	-30	-20	-10	0	10	20	30	40
1	AUTO	10	10	9	8	7	6	9	10
2	METRO	10	10	8	6	5	4	8	10
3	STB	7	7	6	8	9	8	6	5
4	TAXI	10	10	6	5	5	6	8	10
5	Bicycles	1	1	1	2	5	10	10	8
6	Pedestrians	1	1	7	10	10	10	8	5

Table 5. Influence of precipitation level on the choice of the type of transport

	PRECIPITATION	0	1	2	3	4	5	6	7	8	9	10	11	12
1	AUTO	1	1	1	3	5	7	9	10	10	10	10	10	10
2	METRO	1	1	1	2	4	6	8	9	9	10	10	10	10
3	STB	1	1	2	2	3	3	4	4	7	8	9	10	10
4	TAXI	1	1	1	4	5	6	7	8	9	10	10	10	10
5	Bicycles	10	10	10	8	6	4	2	1	1	1	1	1	1
6	Pedestrians	10	10	10	10	8	6	5	4	3	2	1	1	1
	MG	10	10	10	10	9	8	7	6	5	4	3	2	1

Table 6. Influence of wind speed on the choice of the type of transport

	WIND SPEED	0	10	20	30	40	50	60	70	80	90	100
1	AUTO		1	2	4	7	10	8	6	4	3	2
2	METRO		1	2	4	6	8	10	10	10	10	10
3	STB		2	3	5	8	10	10	9	4	2	1
4	TAXI		5	5	6	8	10	10	10	6	3	2
5	Bicycles		10	10	8	5	3	1	1	1	1	1
6	Pedestrians		10	10	10	7	5	3	2	1	1	1
	MG		10	10	10	10	9	8	6	5	2	1

3 Static road generator

By iterating over the entire "present" population (POP) in the city at 00:00, all simulated routes for that day are generated in a single operation.

For this, GTRSIM has a recording and tracking system of the population present in the city at a given time, based on the id, for each person. This system also considers new entrants through GATEs.

Static generation creates for each person considered the entire route of the day composed of several roads and pauses between them.

The generation process takes place only once at 00:00 of each simulated day.

A typical route has the following structure:

Road 1 (departure, arrival, transport_type),
Pause 1 (number of seconds)
Road 2 (departure, arrival, transport_type),
Pause 2 (number of seconds)
...
Road N (departure, arrival, transport_type),
Pause N (number of seconds) = 0

For each Road**, Departure** is a structure that contains: Departure time and the Start coordinate. Departure time characteristics are:

- If it is Road 1, it is randomly allocated according to statistical criteria related to Population Density
- For the following roads in a candidate's route, the departure time is: Time_Arrival_previous_road + Pause_ previous _road

Start coordinate characteristics:

- If it is Road 1, it is randomly allocated according to statistical criteria related to Density
- For the other roads it is: Arrival Coordinate of the previous road

Arrival is a structure that contains:

- The stop coordinate - is randomly assigned according to statistical criteria related to the magnetism of the sections at the initial estimated time.
- Arrival time = Departure_time + estimated_average_time
- Where initially: estimated_average_time = euclidean_distance_departure_arrival / speed_of_transport_type

4 Principles of the GTRSIM simulator

At the beginning of each day, GTRSIM performs a static generation of each candidate's route, which proceeds as follows: the route is composed of "roads" and "pauses" between them.

The data generation mode by the simulator is configured through a set of socio-behavioral matrices completed with data resulting from statistics collected by the authorities.

For each **road**, the type of transport is chosen randomly. The choice is influenced by weather factors (e.g. if it rains, depending on the intensity of the rain, the probability is lower for bicycles and pedestrians)

The "start" of a road is made from the section in which the candidate is located, but the departure time is selected randomly and is influenced by the PD density on sections corrected with the coefficient of PD per hour (TPD). In this way, it is more likely that a candidate from a denser section will leave at an hour with a higher density coefficient.

The "arrival" of a road is chosen randomly but is influenced by the Magnetism (MG) of the sections, which in turn is dependent on the time.

For each generated road, the trajectory is analyzed with a sweep algorithm inspired from [7] and a list of the sections through which it passes from the start to the finish is generated. The journey time is calculated according to the average speed for the type of transport chosen. The duration of the road is subsequently "penalized" by the braking coefficients (BR) of the sections through which the road passes, corrected with the corresponding BR coefficients (TBR) for the respective time.

Pauses are generated randomly but are influenced by the candidate's situation (he or she is employed).

At the same time, on weekends and legal holidays, the rules for "employees" are suspended.

The GATE (the city entry-exit point) behavior is defined by:

a. Magnetic_Intensity_default: "attracts" roads to leave the city

b. Population_Density_default: behaves the same as a district that has a population with a certain density. This population will randomly enter the city on that day through that point.

On weekends, legal holidays, and vacancies, corrections are applied to the density coefficients (entrances to the city) and magnetism (departures from the city) at the entry-exit points of the city, according to the rules presented in the Main Correlations sub-chapter.

Finally, the real_Magnetic_Intensity and the real_Population_Density of a GATE are influenced by various factors: the day of the week (Friday/Sunday), legal holidays, school and university vacation periods, and the weather.

The number of entries into the city per day (NMI). Candidates who enter the city in one

day through GATEs have the same regime for generating roads through the city that day, just like those from POP. They can possibly get out of town the same day if any of their roads are hit by a GATE.
The number of exits in the city per day (NMO). Candidates who leave the city in one day can be from the city or entered earlier from outside the city through a certain GATE.
The population in the POO city is defined as:

POP + NMI - NMO

Through the tracking system of the city's population, people who have left the city at some points are entered into the GATE record and are priority candidates for "returning home".

5 The Generation Algorithm

1. Each section is divided into an ns x ns size grid. For these experiments, the default value ns = 10 was used. By increasing ns, a finer resolution of the simulation is obtained, but the consumption of computing resources increases exponentially. The values of the pair x and y are chosen randomly within each section for each generated road. For each generated road, the start and stop coordinates contain both the address of the section and the x and y coordinates chosen from within the section.
2. Each person will have a route generated for the day that contains between 0 and 10 roads. The total number of roads in a route is determined randomly at the start of the route. For this, the vector of weights (probabilities) is used to generate the number of roads within the route.
3. Within a daily route, the roads are chained
4. After each road, a variable pause is randomly allocated according to the following rules:

- The population of employees (EMPOP) is a percentage of 60% from POP.
For EMPOP, the rule of generation for days 1-5 of the week is: the first road that ends in the interval 7-10 (service entrance) is followed by a break. For this, a vector of weights (probabilities) is used to generate the pause duration. The first "break" can be 8 hours (the duration of the service) or it can be 4 hours followed by "a road to meal", "meal break", "a road to return from the meal" and a new break of 4 hours, i.e. the second part of the service. In both cases, the durations of 8 or 4 hours involving "service" can be variable. At the end of the "service time", GTRSIM will continue the generation of other roads according to the above rules.
- The part of the population that does not behave like an "Employee" is: NEMPOP = POP – EMPOP. For NEMPOP, pauses are generated according to the general vector of weights (probabilities) for pauses.
For days 0 (Sunday) and 6 (Saturday), EMPOP behaves similarly to NEMPOP.
For the rest of the roads for both EMPOP and NEMPOP, the following rule applies: the pause is randomly allocated between 10 min and 12 h.
For each generated route, GTRSIM has to execute the following **steps**:

1. Set the total number of roads in the route. (max 10)
2. Generate the sequence of roads and pauses

For each newly generated road in the route, GTRSIM has to execute the following steps:

1. Generate the type of transport
2. Generate the start time for the first road in the route, or set the start time = stop time of the previous road plus the previous pause duration
3. Generate the start section and the internal coordinates x and y
4. Generate the stop section and the internal coordinates x and y
5. Generate the pause duration
6. Compute the consumed time between start and stop according to the break coefficients of each traversed section at that time and finally set the stop time.

The generation of the necessary values is carried out based on algorithms that use the established vectors populated with values computed in the preprocessing phase of the simulator parameters described above and which ensure a properly weighted distribution at the output.

Example: Using the GTRSIM simulator for the city of Bucharest.
The parameters are:

- city population (POP): 2,000,000
- the average number of entries into the city (NMIO): 150,000
- maximum number of roads per day allowed for one inhabitant (max_roads): 10
- list of Bucharest Sectors - Percentage of the population in each sector
- list of Bucharest neighborhoods
- approximate location matrix of Bucharest districts (Cartesian coordinates on the grid)
- default Densities, Braking, and Magnetism for Bucharest districts
- matrix of Bucharest GATEs
- matrix of Coefficients for Population Density (PD) by districts and Bucharest GATEs by hours.
- matrix of Coefficients for Braking by districts and Bucharest GATEs by hours
- matrix of Coefficients for Magnetism by districts and Bucharest GATEs by hours

Data of interest provided by various institutions and companies:

Distribution of types of transport:

- The annual increase in the number of vehicles: 5.07%
- Metro statistical data: 500K trips per day
- STB statistical data: 2M trips per day [29]
- Taxi: 17,000 (including UBER and BOLT)
- Bicycles/scooters: 1%
- Pedestrians: average walking distance: 3 km/day

The weather data for Bucharest for the last 5 years from [30] have been loaded with the following structure: Temperature, Precipitation, Wind Speed, Wind Direction

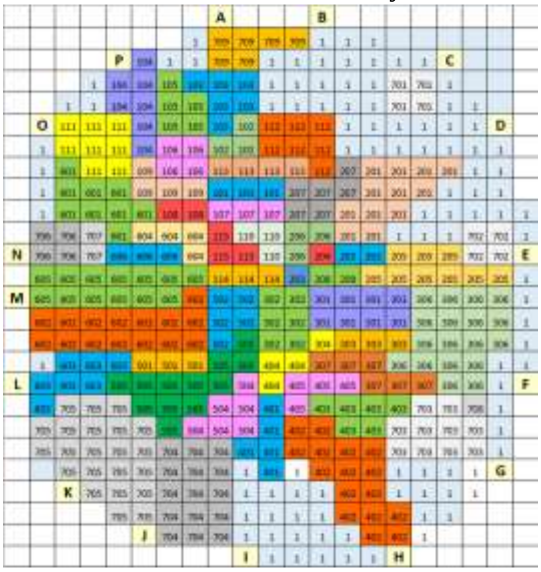

Fig. 1. The grid defined for Bucharest

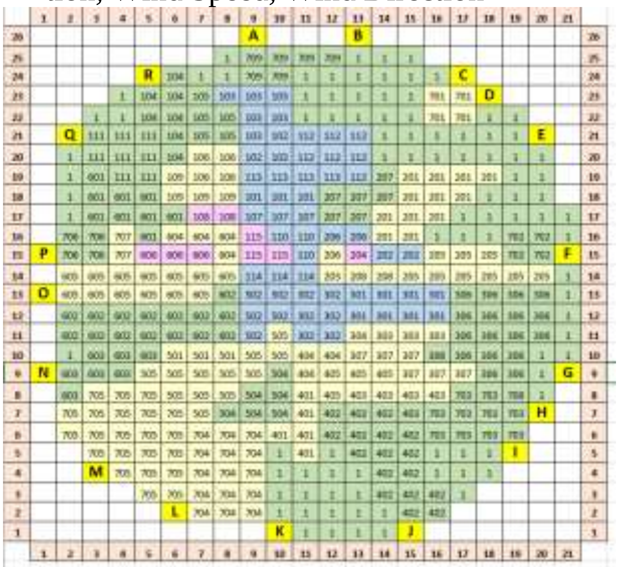

Fig. 2. Distribution by classes of sections in Bucharest

Table 7. The districts codification

Sector 1		Sector 2		Sector 3		Sector 4		Sector 5		Sector 7		neighboring		GATES		
Aviatorilor	101	Colentina	201	Downtown	301	Giurgiului	401	13 Septembrie	501	Crângasi	601	Voluntari	701	1	A	Otopeni - Ploiesti
Aviatiei	102	Iancului	202	Old Town	302	Berceni	402	Cotroceni	502	Drumul Taberei	602	Dobroiesti	702	2	B	Tunari - Balotesti
Baneasa	103	Mosilor	203	Dristor	303	Oltenitei	403	Giurgiului	503	Ghencea	603	Popesti-Leordeni	703	3	C	A Ploiesti
Bucurestii Noi	104	Obor	204	Dudesti	304	Tineretului	404	Ferentari	504	Giulesti	604	Magurele	704	4	D	Afumati - Urziceni
Damaroaia	105	Pantelimon	205	Vacaresti	305	Vacaresti	405	Rahova	505	Militari	605	Bragadiru	705	5	E	Branesti-Lehliu
Domenii	106	Stefan cel Mare	206	Titan	306					Regie	606	Chiajna	706	6	F	A Soarelui
Dorobanti	107	Tei	207	Vitan	307							Rosu	707	7	G	Pasarea - Oltenita
Gara de Nord	108	Vatra Luminoasa	208									Catelu	708	8	H	Berceni - Vidra
Grivita	109											Greenfield Res	709	9	I	Jilava - Giurgiu
Floreasca	110													10	J	Magurele - Darasti
Pajura	111													11	K	Bragadiru - Alexandria
Pipera	112													12	L	Domnesti - Bolintin
Primaverii	113													13	M	A Pitesti
Romana	114													14	N	Chiajna - Dragomiresti
Victoriei	115													15	O	Chitila - Tragoviste
														16	P	Mogosoaia - Ploiesti

Table 8. Legend for districts Categories

districts Categories (Classes)		color
homes and businesses	EQ	
mainly houses	Sender	
mainly business	Receiver	
frequent congestion	Braker	
PINOUT	EQ	

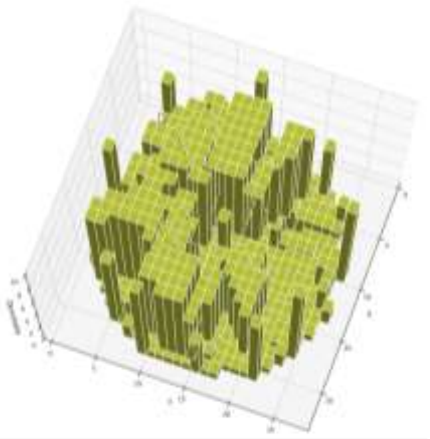

Fig. 3. Density distribution by sections at 7 AM (view from north to south)

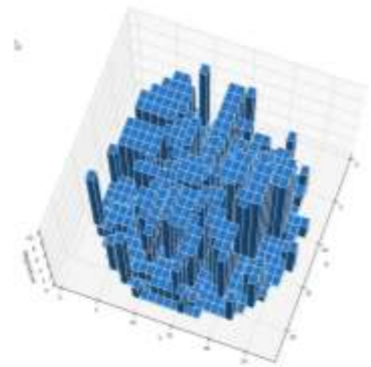

Fig. 4. Magnetism distribution by sections at 7 AM (view from north to south)

GTRSIM implementation

GTRSIM was executed on a scalable server in Google Cloud to be able to benefit from a variable number of processors for intensive parallelization.

The launch parameter tables are persistent in a Cloud SQL database.

The aggregated results could be stored on a persistent disk in the cloud, in cloud storage or they could be transferred to PubSub or Dataflow for different purposes.

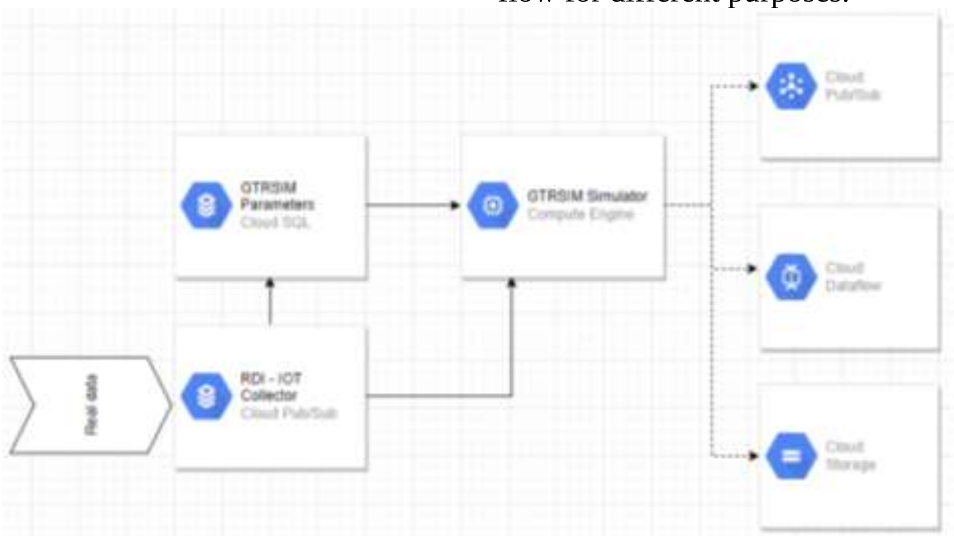

Fig. 5. GTRSIM simulator architecture in the cloud

The simulator was developed with execution speed as the main performance criterion.

The volume of unaggregated raw data generated is over 3 petabytes.

For economic reasons, unaggregated data was not stored in the preliminary testing processes.

For speed reasons, all existing data in the database is loaded into the memory at the beginning of the execution. Comparative speed tests were done for several cache variants and it was demonstrated that the maximum performance is reached if the data is kept in memory in a dictionary-type structure.

The internal simulation coordinator uses the generator for a day to generate sets of roads for all the inhabitants of the city.

The generated data is sent to an internal aggregation processor which in turn processes per time intervals the aggregates considered significant for the prediction process.

Aggregated data is either stored on Google Storage or transmitted directly to Google PubSub.

6 IoT data sources

Thanks to the use of the IoT Collector in the cloud, GDI can receive data from a lot of heterogeneous sensors and sources.

Fixed data sources:

a. Traffic counters. These can be inductive or optical counters that transmit data about the number of cars that pass through a point and possibly their speeds.

b. Simple cameras (with or without artificial intelligence) existing or to be installed at intersections or on road sections.

They will transmit either video streams that require processing in the cloud to extract the data of interest, or they are equipped with their own AI and then directly transmit the number of vehicles or pedestrians that pass in front of them, possibly the registration numbers.

Example: counting the cars waiting to enter an intersection. Two variants were used for the tests: implementation for autonomous workers in the cloud in Python for parallel processing of the video streams and implementation using Google Cloud Vision

Fig. 6. Counting the cars waiting to enter an intersection

Although it has the advantage of low costs per camera, the solution leads to a high processing power requirement in the cloud with an impact on total costs.

c. Cameras with artificial intelligence existing or to be installed in intersections or on road sections.

They can process the video stream locally and transmit to the cloud only the data of interest (number of cars detected in the scene, number of passing cars, number of blocked cars, number of pedestrians, etc.)

For automatic traffic management, it is very important to count the cars detected in the scene, which can provide information about the waiting queues at the entrances to the intersections.

d. Mobile data sources:

- Mobile telephony:
- Voluntary variant (contribs): applications through which traffic participants voluntarily provide data to the system like Waze.
- Automatic variant: statistical and impersonal collection of the data provided by the GPS of mobile phones.
- GPS monitoring systems already exist on cars.
- Telemetry data transmitted by most cars produced in the last 5 years.
- Data from traffic operators and existing dispatchers.

7 Conclusions

GTRSIM is a general-purpose traffic simulator based on city behavior. Even though the initial use was dedicated to Bucharest, the simulator can be used for any other busy city.

GTRSIM was implemented using cloud technologies and was used to generate massive sets of data at the petabyte level. These sets were mixed with data provided by the Real Data Integrator in order to prove the Historical Data Generator (HDDG) concept. The aggregate outputs of HDDG were used later for diverse ML components training.

The simulator was developed with execution speed as the main performance criterion.

The GTRSIM cloud implementation can scale according to the needs because it intensively uses process parallelization.

Compared with other approaches [8] [9] [10] [11] [12] whose main focus is on specific details and aspects related to traffic, GTRSIM is capable to generate very fast large amounts of data needed to train and experiment the ML models.

From playing with the set of injected correlations, it becomes possible to study in a reasonable time the varied behaviors of the trained ML models, both for ordinary traffic conditions and for exceptional situations (ex. irrational behaviors, Heterogeneity, and Nonlinear interactions [4]**)**.

The promising results obtained from the studies on the generation and simulation side allow the transition to the next stage, namely the study and training of ML models for prediction, decision, and control.

References

[1] F. Andreescu, "The Current Scientific Stage of The Instruments and Methods Needed for an Efficient Traffic Management System Based on AI," *Informatica Economică,* vol. 26, no. 1/2022, pp. 46-56, 03 2022.

[2] B. L. Sun, Application and Study on Artificial Intelligence Technoligy in Traffic Singal Control System, Springer, Berlin, Heidelberg ISBN: 978-3-642-23222-0, 2011.

[3] C. M. Tomas Ambra, "Agent-Based Digital Twins (ABM-DT) in Synchromodal Transport," in *Proceedings of the 2020 Winter Simulation Conference*, Brussels, 2020.

[4] N. B. L. F. G. Albi, "Vehicular traffic, crowds, and swarms," *Mathematical Models and Methods in Applied Sciences,* vol. 29, no. 10 (2019), p. 1901, 2019.

[5] P. S. Lorenz Hübschle-Schneider, "Parallel Weighted Random Sampling," *ACM Transactions on Mathematical Software,* vol. 48, no. 3, pp. 1-40, 2022.

[6] Meteoblue, "Descărcare istoric meteo București," Meteoblue, 2023. [Online]. Available: https://www.meteoblue.com/ro/vreme/archive/export/bucure%C8%99ti_rom%C3%A2nia_683506. [Accessed 23 01 2023].

[7] M. Ohdan, "An algorithm for counting line segments intersections using planar partitions," in *ISCCSP 2014 - 2014 6th International Symposium on Communications, Control and Signal Processing, Proceedings*, 2014.

[8] C. A. B. R. S. Y. E. A. D. E. R. M. L. A. V. Allan M de Souza, "Traffic management systems: A classification, review, challenges, and future perspectives," *International Journal of Distributed Sensor Networks* , vol. 13, no. 10.1177/1550147716683612, p. 04, 2017.

[9] DLR - Institute of Transportation Systems, "Eclipse SUMO – Simulation of Urban MObility," DLR - Institute of Transportation Systems, 2022. [Online]. Available: https://www.dlr.de/ts/en/desktopdefault.aspx/tabid-9883/16931_read-41000/. [Accessed 21 09 2022].

[10] TELEGRA, "Complete AI Based Advanced Traffic Management System," 2019. [Online]. Available: https://www.telegra-europe.com/solutions/solution-664-complete-ai-based-advanced-traffic-management-system. [Accessed 10 03 2020].

[11] Ericsson, "Avoiding gridlock using cognitive architecture and smart traffic management systems," Ericsson, 2019. [Online]. Available: https://www.ericsson.com/en/blog/2019/1/avoiding-gridlock-using-cognitive-architecture-and-smart-traffic-management-systems. [Accessed 19 12 2020].

[12] UTI, "Bucharest, Romania – Adaptive traffic management system," UTI, 2019. [Online]. Available: https://www.uti.eu.com/business-lines/intelligent-transportation-solutions/urban-traffic-management/portfolio/bucharest-romania-adaptive-traffic-management-system/. [Accessed 05 06 2019].

[13] V. Kumar, "Artificial Intelligence: Application in traffic management system," 11 Oct 2018. [Online]. Available: https://medium.com/@wrandomwriter/artificial-intelligence-application-in-traffic-management-system-9b5e3d7e620c. [Accessed 20 06 2019].

[14] R. K. M. P. Rndr. Šusteková d., "How Is the Artificial Intelligence Used in Applications for Trafic Management," in *Scientific Proceedings XXIII International Scientific-Technical Conference "Trans & Motauto '15" Year XXIII, Volume 3*, 2015.

[15] D. G. M. Shahgholian, "Advanced Traffic Management Systems: An Overview and A Development Strategy," arXiv funded by Cornell University, 2018.

[16] M. V. S. A. K. N. R. A. S. J. P. Senthil Kumar Janahan, "IoT based smart traffic signal monitoring system," *International Journal of Engineering & Technology,* vol. 7, no. (2.21) (2018) 309-312, 2018.

[17] A. K. M. Treiber, Traffic Flow Dynamics, Springer, 2013.

[18] L. J. X. Z. V.E. Balas, "Information Technology and Intelligent Transportation Systems," in *Proceedings of the 2nd International Conference on Information Technology and Intelligent Transportation Systems*, Xi'an, China, 2017.

[19] Z. Y. R. G. a. F. D. Christoph Sommer, "Simulating the Influence of IVC on Road Traffic," *IEEEE,* vol. 08, no. 978-1-4244-2219-7, p. 1, 2008.

[20] A. M. O. S. Andreas Pell, "Trends in Real-time Traffic Simulation," in *Transportation Research Procedia Volume 25, 2017, Pages 1477-1484*, Shanghai, 2016.

[21] PTV Vissim, "The world's leading multi-modal simulation software," PTV VISSIM, 2019. [Online]. Available: https://www.ptvgroup.com/en/solutions/products/ptv-vissim/areas-of-application/traffic-flow-simulation/. [Accessed 01 06 2019].

[22] SimWalk, "Make the Best of Your Public Transport Infrastructure," SimWalk, 2019. [Online]. Available: https://www.simwalk.com/simwalk_transport/index.html. [Accessed 10 05 2019].

[23] MATSim, "MATSim is an open-source framework...," MATSim, 2019. [Online]. Available: https://www.matsim.org/. [Accessed 03 06 2019].

[24] S. F. D. B. Alberto Pozanco, "Urban Traffic Control Assisted by AI Planning and Relational Learning," in *ATT@IJCAI*, 2016.

[25] S. P. a. B. L. Hariom Tatsat, Machine Learning and Data Science Blueprints for Finance, Sebastopol, CA 95472: O'Reilly Media, Inc., 2021.

[26] J. W. Michelle Vaccaro, "The Effects of Mixing Machine Learning and Human Judgment," *acmqueue,* p. 22, july-august 2019.

[27] L. Moroney, AI and Machine Learning for Coders, Sebastopol, CA 95472: O'Reilly Media, Inc., 2021.

[28] A. F. Vermeulen, Industrial Machine Learning, West Kilbride, UK, 2020.

[29] J.-A. Goulet, Probabilistic machine learning for civil engineers, Cambridge, Massachusetts: The MIT Press, 2020.

[30] TomTom, "Bucharest in the Traffic Index," TomTom, 2023. [Online]. Available: https://www.tomtom.com/en_gb/traffic-index/bucharest-traffic#statistics. [Accessed 14 01 2023].

Florin ANDREESCU has graduated from the Faculty of Computer Science and Automation at The Polytechnic Institute Bucharest (UPB), Domain: Computers/Software Engineering. Currently, he is a Ph.D. student at The Economic Informatics Doctoral School - Bucharest University of Economic Studies (ASE), Domains: Management Information Systems, Artificial Intelligence, and Simulators. He works at Totalsoft as a senior developer and is in charge of an important project in Orange Romania. He has an extensive experience in complex projects including projects for research, industry, governmental, and private companies. He was trained in Google and Azure cloud platforms and in recent projects, he worked in domains like Scalable Machine Learning and Video Streaming in Google cloud, Pub-Sub, Data Flow, Big Query, Auto ML, Vision ML integration, ERP/ORM integration, Data Warehouse, ETL and ML pipelines.