Connected Vehicle Simulation Framework for Parking Occupancy Prediction (Demo Paper)

PIERPAOLO RESCE*, LUKAS VORWERK*, and DANIEL WEIMER, Volkswagen AG, Germany
ZHIWEI HAN and YUANTING LIU, fortiss GmbH, Germany
GIULIANO CORNACCHIA and OMID ISFAHANIALAMDARI, University of Pisa, ISTI-CNR, Italy
MIRCO NANNI and LUCA PAPPALARDO, ISTI-CNR, Italy

This paper demonstrates a simulation framework that collects data about connected vehicles' locations and surroundings in a realistic traffic scenario. Our focus lies on the capability to detect parking spots and their occupancy status. We use this data to train machine learning models that predict parking occupancy levels of specific areas in the city center of San Francisco. By comparing their performance to a given ground truth, our results show that it is possible to use simulated connected vehicle data as a base for prototyping meaningful AI-based applications.

CCS Concepts: • **Information systems** → *Location based services.*

Additional Key Words and Phrases: traffic simulation, parking occupancy prediction, connected car, machine learning

ACM Reference Format:
Pierpaolo Resce, Lukas Vorwerk, Daniel Weimer, Zhiwei Han, Yuanting Liu, Giuliano Cornacchia, Omid Isfahanialamdari, Mirco Nanni, and Luca Pappalardo. 2023. Connected Vehicle Simulation Framework for Parking Occupancy Prediction (Demo Paper). 1, 1 (February 2023), 4 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Nowadays, connected vehicles around the world continuously collect vast amounts of data from LIDAR systems, cameras or other built-in sensors, providing information about their location and their surroundings. This data is a valuable asset for industry, e.g. when developing new, data-driven business models as well as for researchers aiming to advance the future mobility.

However, acquiring such data poses numerous challenges for research as it is usually not made available to the public. Additionally, in its non-anonymized form, it is considered personal data and can only be processed within the legal boundaries that are given in regulations such as the GDPR [Otonomo 2020]. Often, these regulations do not allow for it being used for research or for prototyping new use cases and applications. To overcome this hurdle, we propose a framework that includes a realistic simulation of connected vehicles and traffic in urban areas gathering connected vehicle data. The collected data can serve as foundation for further empirical studies and new data-driven applications.

To demonstrate how our simulation framework is utilized, we apply it to a use case that many car drivers can relate to: parking space search. It not only leads to a dissatisfactory driving experience, it also causes traffic congestion as well as atmospheric and acoustic pollution. In our demo application, we train a set of machine learning models on the generated data and predict on-street parking occupancy levels. In particular, we present a showcase in central San Francisco, exploiting a mix of public datasets and simple data reconstruction heuristics to feed the simulation framework. The ideal result of our empirical study is a proof-of-concept that shows a potential for optimizing the last mile mobility around trip destinations and may facilitate parking space search in the future.

2 RELATED WORKS

2.1 Occupancy prediction

Parking occupancy prediction (POP) can be boiled down to a multivariable time-series prediction problem [Chen 2014]. A major research line of POP relies on static on-street sensors. They provide robust input for supervised POP methods. [Origlia et al. 2019] utilize sensors in parking slots to detect the number of occupied parking slots in real-time and train a supervised regressor for POP. Alternatively, multi-modality data fusion is proved to be beneficial for POP in recent works. [Fiez and Ratliff 2017; Yang et al. 2019] fuse the data from multiple sources, e.g., parking meter transaction data, traffic data and mobility location data, and achieve considerable improvements. [Gong et al. 2021] combine the spatial and temporal information by reformulating a traffic net as a spatial graph. Compared to traditional statistical methods [Fiez and Ratliff 2017], [Gong et al. 2021; Yang et al. 2019] propose to deploy novel neural network based methods for solving POP problems.

Although the static sensor based methods are efficient in terms of performance, they can not be easily scaled out to larger areas due to the high cost and effort of new sensor deployments. An alternative is to exploit the streaming event data, e.g., from LIDAR systems, sensors or cameras of moving vehicles. The research potential is much less explored in this direction compared to using static sensors. To our best knowledge, [Bock et al. 2019] is the only work that uses the data from vehicle sensors, however, their data source is limited to only taxis. Propelled by the development of Car2X and 5G technologies, data collected by vehicles can be used when proper anonymization and encryption methods are applied for

*Both authors contributed equally to this research.

Authors' addresses: Pierpaolo Resce, pierpaolo.resce@volkswagen.de; Lukas Vorwerk, lukas.vorwerk1@volkswagen.de; Daniel Weimer, daniel.weimer1@volkswagen.de, Volkswagen AG, Munich, Germany; Zhiwei Han, han@fortiss.org; Yuanting Liu, liu@fortiss.org, fortiss GmbH, Munich, Germany; Giuliano Cornacchia, giuliano.cornacchia@phd.unipi.it; Omid Isfahanialamdari, omid.isfahanialamdari@di.unipi.it, University of Pisa, ISTI-CNR, Pisa, Italy; Mirco Nanni, mirco.nanni@isti.cnr.it; Luca Pappalardo, luca.pappalardo@isti.cnr.it, ISTI-CNR, Pisa, Italy.

XXXX-XXXX/2023/2-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

data transmission. Therefore, we propose a simulation framework and a proof-of-concept to boost the future research of POP using streaming event data.

2.2 Parking simulation

Works like [Leclercq et al. 2017] use real on-street parking data and stochastic numerical experiments to simulate and investigate how different parking strategies affect the parking search time and distance. [Codecá et al. 2018] investigates on how to monitor parking areas by integrating them in a SUMO simulation environment. It also provides some methods to aggregate and compute the intention of a vehicle of occupying a given parking area. [Vo et al. 2016] investigates the modelling of a system that simulates vehicles moving between parking lots in a NetLogo environment. [Balmer et al. 2006] describes a modelling framework for large scale scenarios. Previous works regarding connected vehicle simulation scenarios were mainly done in the field of communication infrastructure simulation [Kim et al. 2017] and to spot anomalies in the communication of connected vehicles due to cyber-attacks [Levi et al. 2018]. To the best of our knowledge, this is the first work that presents a simulation framework able to simulate connected vehicles that send event data to a backend system, that describes directly the surrounding area of a simulated agent.

3 SIMULATION

In this section, we describe the simulation framework we propose. While it is general and can be applied to any desired area, we will consider a reference case study located in the San Francisco downtown area. The main driver for the selection of this area is the availability of public datasets for traffic, parking slot locations and parking occupancy patterns.

3.1 Parking slots reconstruction

To simulate a realistic parking situation, the locations and occupancy levels of parking slots are required. When creating the simulation for a specific area, we identify three possible situations: fully available data, partially available data and no available data.

In case, the location and occupancy data is fully available (e.g. data from parking meters/sensors), we can import this data into our simulation framework and let the simulation run accordingly. In case of partially available data, the missing areas can be filled using interpolation and mapping distributions from known areas within the same city. If no data is available at all, it might be an option to map known parking spot distributions and occupancy levels from one city to the target city. However, this is the least desired case with the most drawbacks. In the given case of San Francisco, we are faced with the second option - we used publicly available datasets for parking slots locations and occupancy levels of parts of the San Francisco city center and the framework fills the missing areas.

As base layer, we use OpenStreetMap data to include the descriptions of roads and traffic lights and insert on-street parking spots. As part of the SFPark project by the San Francisco Municipal Transporation Agency, occupancy rates of stationary on-street parking sensors were collected. Datasets containing the locations and occupancy levels of these sensors are publicly available and serve as the real-world data source for parking spot locations and occupancy information [1]. After inserting the real-world parking spots, we applied the following steps to fill gaps with synthetic parking spots where we do not have real-world information:

(1) Compute the distribution of "spots per meter"-values for roads where sensors are deployed
(2) Find roads without sensors and use the distribution of step 1 to assign a "spots per meter" value
(3) Create new parking spots along these roads according to their assigned "spots per meter"-value

To fill missing parking occupancy information, we divided the selected area in hexagons (see Figure 1), and computed the percentage of occupied parking spot at each hour per hexagon for which the measurement is available. To have meaningful data for hexagons without parking occupancy information, we spatially interpolated the data. The resulting occupancy patterns at midnight and 7 PM can be observed in Figure 1.

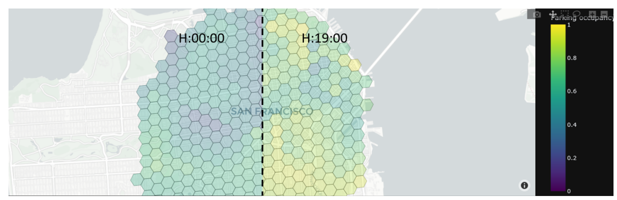

Fig. 1. Park occupancy pattern at midnight and 7 PM

3.2 Mobility demand reconstruction

To model real-world traffic demand, we used a dataset proposed by Piorkowski et al. in [Piorkowski et al. 2009], describing the GPS coordinates of 536 taxis collected over 25 days in San Francisco (USA)[2]. Outliers and noise GPS points were removed, and then the GPS traces were segmented into trajectories (i.e. trips).

To generate a plausible mobility demand reflecting real traffic patterns in the selected area of San Francisco (surface of $\approx$ 16 km^2), we computed an origin-destination (OD) matrix M from real mobility traces, based on a regular grid that splits the region of interest into squared cells with a size of 400 meters. The element $m_{i,j} \in M$ contains the number of trajectories having origin in tile i and destination in tile j. Trips are then generated by first randomly selecting a pair of tiles (i, j) with probability $p(i, j) = m_{i,j} / \sum m_{i,j}$; then a road edge is randomly selected in tiles i and j, which are used as starting and ending edges.

3.3 Simulation framework

The developed framework consists of a set of Python classes that interact with the traffic simulation package SUMO by means of its APIs. The code extends SUMO functionalities to replicate the behavior of a connected fleet in the selected area. Implementing full

[1] The datasets are available at https://data.sfgov.org/Transportation/Parking-Meters/8vzz-qzz9/data and https://www.sfmta.com/getting-around/drive-park/demand-responsive-pricing/sfpark-evaluation
[2] The dataset is available at https://crawdad.org/epfl/mobility/20090224/

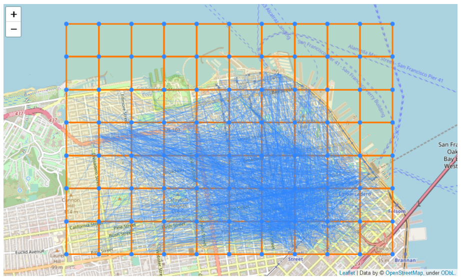

Fig. 2. OD trips in tessellated map

connected vehicle capabilities (e.g. vehicle-to-vehicle communication) is beyond the scope of our research, therefore we focused on the specific functionality of detecting the occupancy status of parking spots along a traveled road and transmitting this information to a simulated backend system.

The interaction between the framework modules can be described as follows: the datasets described in 3.1 and 3.2 can be considered as target traffic and parking occupancy patterns in our simulation. These datasets are used to feed the developed Python objects, that take care of controlling the simulation, by actively inserting cars, or letting cars park inside the SUMO simulator. The simulator in this case can be considered as a semi-supervised environment: we impose certain traffic and parking operations inside simulation, but the simulated vehicles will have sufficient freedom to interact with each-other.

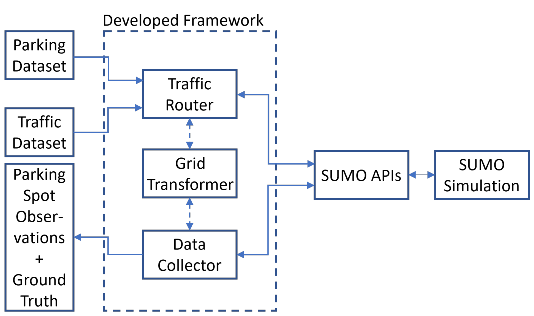

Fig. 3. Modules interaction diagram

The key elements of the developed framework are summarized below:

Data Collector: collects information regarding the status of each vehicle and their observations of parking spot occupancy levels. In line with the real-world capability of modern cars to detect empty or occupied parking spots with cameras and sensors, cars in our simulation gather this information by passing on-street parking spots and determining whether a spot is occupied. Every observed parking spot status along the traveled edges is collected and stored for further processing. The data collector additionally collects the ground truth for parking spot occupancy at each step to prepare the dataset for model training.

Traffic Router: orchestrates traffic demand and parking status, making the simulation replicate desired mobility patterns, represented as "traffic dataset" and "parking dataset" blocks in Fig. 3. This is done by inserting vehicles into the simulation and routing them according to the dataset as well as synchronizing the parking occupancy levels with the given data.

Grid Transformer: maps geographic coordinates to simulation coordinates and vice-versa. It is responsible for bringing a given map into a grid form with addressable grid cells and performs spatial aggregations where needed.

At each simulation step, the Data Collector stores simulated observations and ground truth, and at the end of the simulation it stores all the collected data to the generated dataset, represented as "Parking Spot Observations + Ground Truth" in Fig. 3. This represents also the entire set of data required to feed the machine learning pipeline.

3.4 Results

The realism of the simulation was evaluated by taking into account the following criteria:

Duration of trips: the distribution of the simulated trip duration must match the real distribution of trips observed in SF taxi dataset.

Traffic management: the Traffic Router module schedules the trips of simulated cars, which once inserted in simulation are able to follow properly the desired behavior of traffic demand described in 3.2. During the trips, the vehicles behave like expected: they move in the city and collect parking spots observations, but due to SUMO limitations it was not possible to manage more than 500 active (non-parked) cars simultaneously without facing the so-called "grid lock" issue of the SUMO simulator.

Management of parking demand: the Traffic Router module also takes care of managing parking occupancy in each position at each instant. According to the parking reference described in section 3.2 it was possible to replicate the desired parking schedule in the in simulation.

4 PARKING OCCUPANCY PREDICTION

In this section, we conduct a comparative study to show the viability of using simulated streaming event data for POP. We compare five prediction models that were trained on the SFPark occupancy dataset from static sensors as well as on the generated dataset from simulated, moving vehicles and their sensors. To our best knowledge, this study is the first proof-of-concept that aims to conduct POP with synthetic streaming event data.

4.1 Experimental settings

4.1.1 Dataset. For the static sensor setup, the ground truth parking data is simply adapted into the input data while only the parking slots observed by an actively moving vehicle will be used as input data in the vehicle sensor setup. The final datasets include 24 hours of data collected within a selected area in the city center of San Francisco and is sorted in chronological order.

4.1.2 Preprocessing. For each simulation step (2 steps/second of the simulated time), the received event data (the observed parking slot status and their locations) is aggregated into two observation matrices (occupied and free parking slot number matrices) by their locations according to the tessellation in Fig. 2. This is followed by a temporal aggregation, which sums up the sparse observation matrices of 5 minute time intervals to a dense observation matrix. The dense aggregated matrices are then Gaussian smoothed with a window size of 10 and normalized by the total parking slot number and the fleet size in the simulation. In the experiment, the prediction models are trained to predict the parking occupancy level of each grid cell in the next 30 minutes from the preprocessed parking slot observation matrices.

4.1.3 Prediction Models and Implementation Details. In the comparative study, we implemented experiments with five prediction models. We adapted three statistical prediction models, (linear regressor, random forest regressor and KNN-regressor) from sklearn. Moreover, we implemented two deep learning models (convolutional neural network and multilayer perceptron) using PyTorch. The deep learning models are optimized by minimizing the mean squared error (MSE) of predicting the grid cell parking occupancy using the Adam optimizer, where the batch size is 256 and the learning rate is 0.001. All model parameters are initialized by a default Gaussian initializer. We applied a bottleneck architecture to the multilayer perceptron and did a grid search for the layer number amongst $\{3, 4, 5\}$. For the convolutional neural network, we adapted the encoder-decoder architecture with matrix input and did a hyperparameter grid search for the encoder layer amongst $\{1, 2, 3\}$ and number of channels amongst $\{32, 64\}$.

4.1.4 Evaluation protocols. The performance of the implemented POP models is evaluated by the mean absolute error (MAE) of the predicted grid parking occupancy to the ground truth grid parking occupancy. The MAE is defined as following,

$$\mathrm{MAE} = \frac{1}{T \times N} \sum_{t=1}^{T} \sum_{k=1}^{N} |\tilde{y}_{tk} - y_{tk}|, \tag{1}$$

where $\tilde{y}_{tk}$ and y_{tk} are the parking occupancy prediction and ground truth of the kth grid at the time t, N is the number of total grid cells in the prediction area and T is the prediction time horizon.

4.2 Experimental Results

Table 1 shows the POP performance comparison of 5 implemented models using static and vehicle sensor setups. The neural network based models demonstrate their superior performance compared to other statistical models using both sensor setups. Three models perform better under the static sensor setup, two with considerable margins (4.5% using MLP and 5.3% using KNN-regressor). Two models are better using the vehicle sensor setup (-0.7% using DT and -1.2% using RF). Nevertheless, the performance gaps measured by MAE on all models have a maximum of 5.3% (KNN-regressor) and are in general not large. The observations indicate that, although the best POP model for the static sensor setup outperforms the best model in the vehicle sensor setup, the vehicle sensor setup is able to offer a comparable performance in POP tasks using various prediction models. Our study suggests that the vehicle sensor setup enabled by connected vehicle functionalities is an ideal replacement for full static sensor setup in terms of flexibility and deployment.

Models	static	vehicle	performance attenuation (percentage)
MLP (5 layers)	**0.072***	**0.117**	0.045 (4.5%)
CNN (1 layer, 16 dim.)	**0.105**	**0.110***	0.005 (0.5%)
DT	0.189	0.182	-0.007(-0.7%)
RF	0.243	0.231	-0.012(-1.2%)
KNN-regressor	0.177	0.230	**0.053(5.3%)**

Table 1. Performance comparison of implemented POP models with two sensor setups: the star and bold values in the second and third columns indicate the best and second-best performance in MAE. The fourth column illustrates the performance attenuation under the vehicle sensor setup.

5 CONCLUSION

We presented a simulation framework for connected vehicles and their capabilities to collect data about their surroundings. Using the simulated data, we trained machine learning models that predict parking occupancy levels in urban areas. By comparing these models with models trained on real-world data from static sensors, we demonstrated that simulated vehicle data can serve as basis for data-driven applications and use cases. This proof-of-concept should encourage researchers and industry to consider simulated data when exploring new applications for connected vehicle data.

REFERENCES

Michael Balmer, Kay W Axhausen, and Kai Nagel. 2006. Agent-based demand-modeling framework for large-scale microsimulations. *Transportation Research Record* 1985, 1 (2006), 125–134.

Fabian Bock, Sergio Di Martino, and Antonio Origlia. 2019. Smart parking: Using a crowd of taxis to sense on-street parking space availability. *IEEE Transactions on Intelligent Transportation Systems* 21, 2 (2019), 496–508.

Xiao Chen. 2014. Parking occupancy prediction and pattern analysis. *Dept. Comput. Sci., Stanford Univ., Stanford, CA, USA, Tech. Rep. CS229-2014* (2014).

Lara Codecá, Jakob Erdmann, and Jérôme Härri. 2018. A sumo-based parking management framework for large-scale smart cities simulations. In *2018 IEEE Vehicular Networking Conference (VNC)*. IEEE, 1–8.

Tanner Fiez and Lillian Ratliff. 2017. Data-driven spatio-temporal analysis of curbside parking demand: A case-study in seattle. *arXiv preprint arXiv:1712.01263* (2017).

Shuhui Gong, Xiaopeng Mo, Rui Cao, Yu Liu, Wei Tu, and Ruibin Bai. 2021. Spatio-temporal Parking Behaviour Forecasting and Analysis Before and During COVID-19. *arXiv preprint arXiv:2108.07731* (2021).

Heejae Kim, Jiyong Han, Seong-Hwan Kim, Jisoo Choi, Dongsik Yoon, Minsu Jeon, Eunjoo Yang, Nhat Pham, Sungpil Woo, Jeongkyu Park, et al. 2017. IsV2C: An integrated road traffic-network-cloud simulator for V2C connected car services. In *2017 IEEE International Conference on Services Computing (SCC)*. IEEE, 434–441.

Ludovic Leclercq, Alméria Sénécat, and Guilhem Mariotte. 2017. Dynamic macroscopic simulation of on-street parking search: A trip-based approach. *Transportation Research Part B: Methodological* 101 (2017), 268–282.

Matan Levi, Yair Allouche, and Aryeh Kontorovich. 2018. Advanced analytics for connected car cybersecurity. In *2018 IEEE 87th Vehicular Technology Conference (VTC Spring)*. IEEE, 1–7.

Antonio Origlia, Sergio Di Martino, and Yuri Attanasio. 2019. On-Line Filtering of On-Street Parking Data to Improve Availability Predictions. In *2019 6th International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)*. IEEE, 1–7.

Otonomo. 2020. *A Privacy Playbook for Connected Car Data*. Technical Report. Otonomo.

Michal Piorkowski, Natasa Sarafijanovic-Djukic, and Matthias Grossglauser. 2009. CRAWDAD dataset epfl/mobility (v. 2009-02-24). Downloaded from https://crawdad.org/epfl/mobility/20090224. https://doi.org/10.15783/C7J010

Thi Thuy An Vo, Peter van der Waerden, and Geert Wets. 2016. Micro-simulation of car drivers' movements at parking lots. *Procedia engineering* 142 (2016), 100–107.

Shuguan Yang, Wei Ma, Xidong Pi, and Sean Qian. 2019. A deep learning approach to real-time parking occupancy prediction in transportation networks incorporating multiple spatio-temporal data sources. *Transportation Research Part C: Emerging Technologies* 107 (2019), 248–265.