Semi-synthetic Data and Testbed for Long-Distance E-Vehicle Routing

Andrius Barauskas, Agnė Brilingaitė, Linas Bukauskas, Vaida Čeikutė, Alminas Čivilis, and Simonas Šaltenis(✉)

Institute of Computer Science, Vilnius University, Vilnius, Lithuania
{andrius.barauskas,agne.brilingaite,linas.bukauskas,vaida.ceikute, alminas.civilis,simonas.saltenis}@mif.vu.lt

Abstract. Electric and autonomous mobility will increasingly rely on advanced route planning algorithms. Robust testing of these algorithms is dependent on the availability of large realistic data sets. Such data sets should capture realistic time-varying traffic patterns and corresponding travel-time and energy-use predictions. Ideally, time-varying availability of charging infrastructure and vehicle-specific charging-power curves should be included in the data to support advanced planning.

We contribute with a modular testbed architecture including a semi-synthetic data generator that uses a state-of-the-art traffic simulator, real traffic distribution patterns, EV-specific data, and elevation data to generate time-dependent travel-time and energy-use weights in a road-network graph. The experimental study demonstrates that the testbed can reproduce travel-time and energy-use patterns for long-distance trips similar to commercially available services.

Keywords: Semi-synthetic data · Data generation · Testbed · Electric vehicle · Long-distance EV routing · Time-dependent road network

1 Introduction

Transportation is currently undergoing a profound transformation. This is driven by the emergence of new automotive technologies, such as electric (EV) and autonomous vehicles, new business models such as ridesharing, and the continued digitalization of all aspects of transportation. For example, the efficiency of a fleet of autonomous electric vehicles will be highly dependent on effective routing and scheduling algorithms and these will, in turn, depend on data-driven predictions of travel time and energy use. Furthermore, as real-world routing problems are often formulated as multi-objective optimization involving multiple constraints, the optimal algorithms are intractable; thus, only heuristic algorithms are possible [2]. The efficiency and the efficacy of such algorithms can only be tested through extensive experimental studies on large datasets and workloads.

L. Bellatreche et al. (Eds.): ADBIS 2021 Short Papers, Workshops and Doctoral Consortium, CCIS 1450, pp. 61–71, 2021.
https://doi.org/10.1007/978-3-030-85082-1_6

To understand the complexity of the data required by real-world routing algorithms, consider a long-distance EV routing query. It has to take into account the predicted traffic to estimate both the expected travel time and the expected energy use. To plan charging stops, this information is combined with the information about the availability and the power of chargers. Both the traffic and the availability of chargers are *time-dependent* (TD). Furthermore, we argue that any realistic long-distance routing system has to work with the inherent uncertainties of predictions. Thus, the travel time, the used energy, and the time waiting for charging are all modeled as *intervals* of expected values.

Research studies that explore advanced routing problems expend much effort to prepare their experiments. For example, to implement Eur-PTV and Ger-PTV benchmarks [2], road network data, elevation information, energy consumption data, traffic data, and charging station data are preprocessed and integrated. Åkerblom et al. [1] extend the simulation framework of Russo et al. [15] taking the traffic patterns from the LuST Scenario data [5]. Several studies apply statistical and machine learning methods to forecast travel time and future congestion along the route using data collected from Google Maps Platform API. Traffic conditions can be identified by capturing traffic layer image and identifying *color* data on monitored road segments [14,18], or using Estimated Time of Arrival [17]. Suggested methodologies are time consuming—to apply machine learning algorithms, a substantial amount of data has to be collected during an extended period of time. Brinkhoff [3] pioneered a framework to generate moving objects on a road network. The framework did not consider traffic models and resulting vehicle movements were not very realistic. In contrast, the open-source microscopic traffic simulation tools, such as SUMO [12], used in this work, and GeoSparkSim [16], were designed to handle realistic traffic simulation on large-scale road networks.

This paper aims to do the necessary legwork for the road-network algorithms community. While there are a few traffic simulators and general-purpose spatial and graph data generators, we provide, to the best of our knowledge, the first testbed for experimentation with advanced routing algorithms, in particular, algorithms for EVs.

The paper contributes with a modular architecture and a data preparation workflow to generate realistic semi-synthetic EV-specific TD traffic data that captures uncertainty. We provide a layer of services on top of the generated data to be used as building blocks of future advanced routing algorithms. The experiments indicate that the proposed environment provides data patterns similar to commercial ones, and it can be used to test the TD routing of EVs.

The work is structured as follows. Section 2 introduces the testbed architecture. Sections 3 and 4 present implementation details and the experimental evaluation of the testbed, respectively. Section 5 concludes the paper.

2 Semi-synthetic Data Generation and Testbed API

2.1 Testbed Architecture and Functionality

Generating and managing the test data introduced above calls for a multi-component architecture (see Fig. 1). First, driving speed depends on the traffic at a particular time. Therefore, the TD Traffic Information component requires Traffic Simulation data and TD Traffic Statistics to define parameters of road edges. Second, the Energy Consumption component is dependent on elevation data and the consumption function that uses the EV properties as its parameters. Finally, long-distance EV routing requires charging stops along the road. Hence, the component of Charging Stations is supported by TD availability data of charging stations and charging function that uses the parameters of EV type.

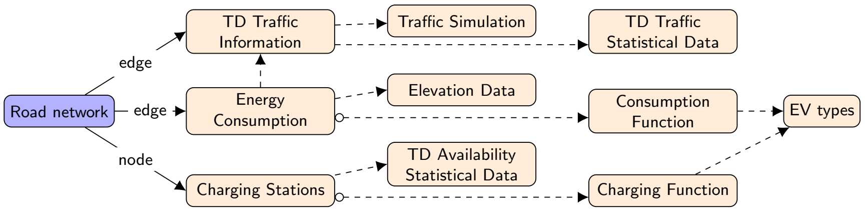


Fig. 1. Components of the testbed architecture

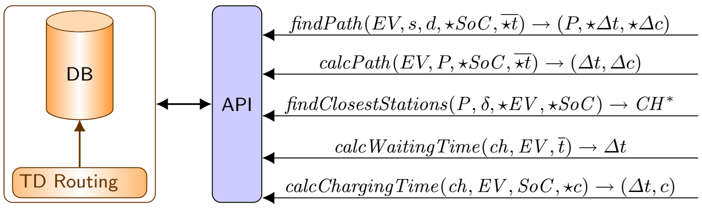

DB
TD Routing
API
$findPath(EV, s, d, \star SoC, \overline{\star t}) \rightarrow (P, \star\Delta t, \star\Delta c)$
$calcPath(EV, P, \star SoC, \overline{\star t}) \rightarrow (\Delta t, \Delta c)$
$findClosestStations(P, \delta, \star EV, \star SoC) \rightarrow CH^*$
$calcWaitingTime(ch, EV, \overline{t}) \rightarrow \Delta t$
$calcChargingTime(ch, EV, SoC, \star c) \rightarrow (\Delta t, c)$

Fig. 2. Testbed API

While the main contribution and focus of this work is the generation of semi-synthetic data, a thin layer of services is proposed as well. Such services query and aggregate the data and can be used as the building elements of advanced routing algorithms. Figure 2 presents five API functions with optional parameters marked by $\star$. Function *findPath* uses a TD router to construct a path P and to estimate the expected trip duration interval Δt and the expected energy consumption interval Δc when traveling from start s to destination d and starting the trip some time during $\overline{t}$ time interval. The starting time is given as an interval, which is useful if the function computes a leg of a longer route. If the initial state of charge of the EV battery SoC is given, the returned Δc is the final expected interval of the state of charge of the battery, rather than the consumed energy.

Function *calcPath* is used to calculate the same travel estimates on an already known path P. Function *findClosestStations* returns a set of charging stations CH^* containing the stations within a Euclidean buffer δ around path P and reachable by EV when starting on the path with SoC. Finally, functions *calcWaitingTime* and *calcChargingTime* return waiting-time and charging-time intervals Δt at charging station ch for EV. A waiting time interval depends on the daytime interval when the EV reaches ch. Also, a charging time interval depends on the SoC before starting the charging process. The required SoC c can be provided and the reached SoC c is returned.

2.2 Traffic Data Simulation and Calibration

Traffic data preparation process is shown in Fig. 3. To prepare the road network (RN), first, map data is filtered leaving only car roads. Next, the road network graph is made routable (a directed graph), and finally routable network segments are augmented with length data and free-flow speed data (speed-limit data).

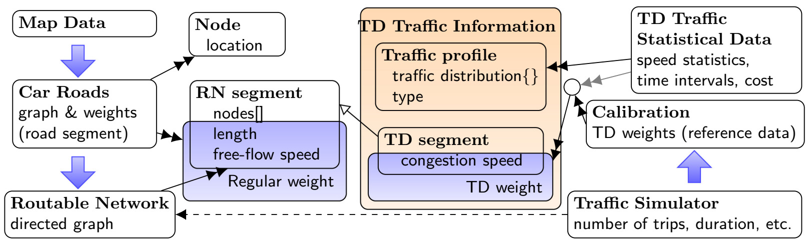


Fig. 3. Traffic data preparation

Two main sources are used to generate the semi-synthetic TD weights of road-network segments. TD traffic statistical data for a given region describes how traffic at large changes relatively to the time of day. This is used to derive a traffic profile. Then, network segments are augmented with congestion speed data—either real statistical data, if available, or synthesized data generated by traffic simulators. Finally, the results of simulations are calibrated using commercial traffic data providers.

Semi-synthetic TD segment weights are composed of edge-specific minimum traffic speed, edge-specific maximum traffic speed, and region-wide TD traffic distribution. Given a time of day, they are used to calculate edge-specific traffic speed as a weighted average of the minimum and the maximum traffic speeds of a segment. We use maps from the OpenStreetMap project (OSM, [7]). Thus, the maximum speed is the free-flow speed from OSM, the minimum speed is derived from congestion modeling using open-source traffic simulator SUMO [8], and the TD traffic distribution is sourced from TomTom's (TT) Traffic Index.

SUMO takes a routable network as data input for traffic simulation and augments it with simulated traffic data. The testbed's routable network is fed to

SUMO using the `netconvert` tool. The output of the simulation is a congestion-hour travel time for each segment on the routable network. To perform a simulation, the whole map is divided into regions and each region is simulated separately. Random traffic generation method of the SUMO tool *randomTrips* is used. This method allows choosing different weights affecting the probability of selecting a segment for routing. Segment length is used as a weight; thus, dense regions like city centers get more traffic. Finally, the number of trips is calculated proportionally to the population size of the region and distributed evenly in an interval from 0 to 3600 s.

Assuring realistic generated data requires *calibration* of both the free-flow and congestion travel times. The calibration is implemented via two coefficients for the congestion speed and the free-flow speed. The coefficients are calculated by comparing simulated travel times with Google Maps travel times. First, two sets of routes are generated—inside cities and out of cities—for congestion and free-flow travel time calibration, respectively. Then, travel times are calculated at peak hours for inside-cities set and off-peak hours for out-of-cities set.

2.3 Data for Energy Consumption Estimation

Energy consumption (EC) along a given route is estimated by adapting the Vehicle Energy Model (VEM) as introduced in the SUMO simulator [11]. In addition, the EC model considers traffic information to estimate TD energy use along the route.

Energy consumption calculation uses two types of parameters—vehicle specific and road-network dependent. The following EV characteristics are employed: battery, vehicle mass, front surface area, air drag coefficient, internal moment of inertia, radial drag coefficient, roll drag coefficient, propulsion efficiency, recuperation efficiency, and constant power intake. While we are currently using a predefined set of these values, the constant power intake parameter could be extended and vary based on weather conditions for more precise modeling. Such EV data can be collected from various sources, including car manufacturers and EV enthusiasts that try to measure various parameters of their vehicles under specific conditions.

The core road-dependent parameters, the slope and the radius, are precomputed for each EC segment and stored in the database. In addition, a segment inherits free-flow speed, length, congestion speed, and contains node coordinates, as it extends the TD segment. Segment geometry is used to compute the length and the radius of each segment. We deem the slope and the radius as the essential terrain approximation parameters.

2.4 Charging and Waiting Times at Charging Stations

Each charging station contains a set of chargers (see Fig. 4), a TD availability profile, and geographic location to be mapped to the road network. The model could be extended with other features, e.g. connection fee or charging price. Each

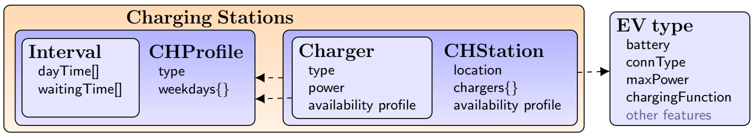


Fig. 4. Domain model of charging stations

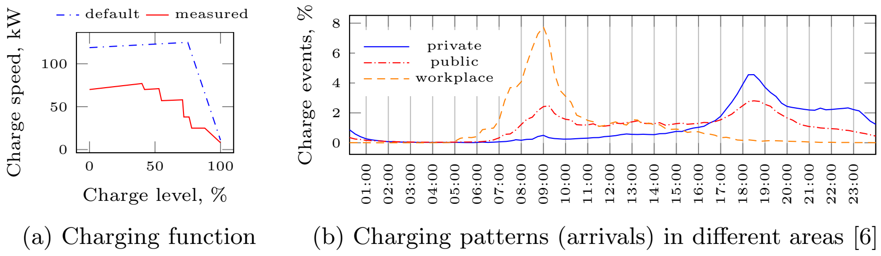


Fig. 5. Charging functions of two different 65 kWh batteries [4] and charging demands

charger of the station is described by a connector type, power, and its own availability profile. Figure 5a presents two charging functions for 65 kWh battery—the default one and one measured by observation. The charging function depends on EV features and charger properties. First, some EV types are limited by their own maximum charging power. Second, charging process is slower when the battery's SoC is below 20% and above 80%, especially in the case of rapid charging. Charging functions are retrieved from open data available on the internet, e.g. [4], to get EV maximum power and power at different points of charging (piece-wise linear function) for different chargers. Availability of a charging station or an individual charger can be represented by a piece-wise linear function of time. Features like type, e.g. rural areas, and weekdays, e.g. Sunday and Saturday, define a particular availability profile. Figure 5b shows percentage of charging cases throughout a 24-h period for private, public, and workplace charging points on work days [6]. For example, at 9:00 the need for power is very high at workplaces. At night a number of charging cases is low in all cases. Therefore, the availability profile can be constructed based on observation data with a high probability of a waiting time that depends on the charger power.

3 Testbed Implementation

Figure 6 summarizes the process of semi-synthetic data preparation using open tools and data sources. Germany map was retrieved from OSM and filtered for vehicle roads using `osmfilter`. The `osm2po` tool generated a routable network and it was stored in PostgreSQL with PostGIS extension. SUMO tool was used to simulate traffic flow. Congestion index was obtained from TomTom. Network-segment slopes were calculated based on CGIAR-CSI SRTM 90 m Digital Ele-

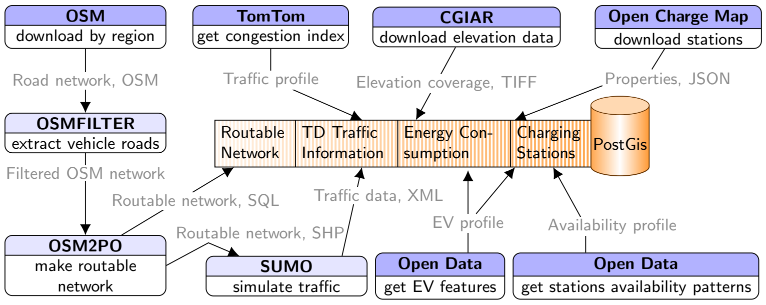


Fig. 6. Data sources and processing

vation Data [10]. EV features and charging stations [13] with their availability patterns were set up using publicly available data.

At the core of *findPath* and *calcPath* functions (see Fig. 2) is the computation of the total energy and the total driving time on a route. For each segment, the EC segment properties at a given time are used to calculate the travel time and energy required to traverse the segment. For some segments, it might result in a negative value e.g. going downhill. If so, the calculation has to make sure that the battery is not charged more than its capacity. Finally, the estimations for each route segment are added up to get the total energy and travel time.

Travel speed $v(t, seg)$ along the segment seg is a time function as it varies based on traffic conditions. Each TD segment has an estimated free-flow speed, $v_{freeflow}(seg)$, and congestion speed, $v_{congestion}(seg)$. The speed along the segment varies between the two extremes. This is modeled via the $cost(t)$ function defined by the traffic profile:

$$v(seg, t) = v_{freeflow}(seg) - (v_{freeflow}(seg) - v_{congestion}(seg)) \cdot cost(t).$$

The testbed simulates the uncertainty of prediction by assuming that the timing of peaks in the traffic profile might slightly shift from day to day. The testbed calculates the minimal cost value and the maximal cost value for each segment using the time window defined by t and uncertainty ϵ:

$$_{min}cost(t) = \min(cost([t - \epsilon, t + \epsilon])),\ _{max}cost(t) = \max(cost([t - \epsilon, t + \epsilon])).$$

The default ϵ value is set to 30 minutes, but can be adjusted. Note that t is a time when an EV reaches a given segment seg_i along the route. Thus, it depends on the travel speed and departure time of previous $i-1$ segments. Let us assume the trip start time is t_{start} and Δt_i is the time required to pass segment i, then seg_i entrance time t_i is $t_i = t_{start} + \sum_{n=1}^{i-1} \Delta t_n$. For the whole route, the bounds of the estimated energy and time intervals are calculated in two iterations. The first iteration uses $_{min}cost(t)$ as the cost function for the lower bound and the second iteration—$_{max}cost(t)$ for the upper bound.

The testbed contains synthetic data to estimate waiting times at charging stations at different times of the day. Various profiles, e.g. business premises, were integrated to follow statistical data on charging patterns [6]. Then, waiting time intervals were constructed as $wt = (wt_{min}, wt_{max}) = (0, \Delta ch)$ where Δch is the time needed to charge from 10% to 100% SoC and 80% at AC and DC chargers, respectively. For example, $\Delta ch = 15$min in the case of rapid charging. Afterwards, the intervals were shifted based on the number of chargers of the same type, location of the charging station in relation to the highway, and charging patterns (similar to TD travel speed). The calculation and tuning details are left out of the scope of this paper.

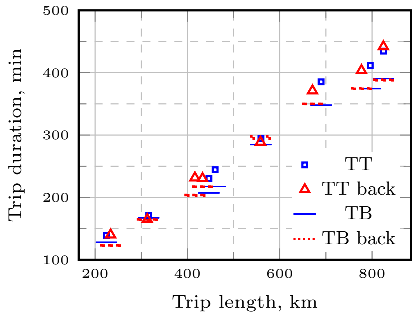
4 Experiments and Results

KaTCH [9] implementation of time-dependent contraction hierarchies was integrated into the testbed as a routing engine, and several tests were run to illustrate realistic results and appropriate scalability within the testbed.

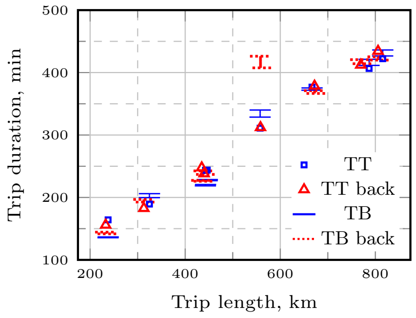
As a case study, sources and destinations were chosen for 8 representative trips in Germany. Then, the travel-time and energy-consumption intervals were calculated for all of them when traveling from a source to a destination and back—16 individual trips in total. Also, the departure times were set to 00:00 and 16:00 as non-congestion and congestion-time representatives. To estimate energy consumption in both testing environments, the energy use curve was constructed as a sequence of pairs (kWh per 100 km, km/h)—$(24.94, 10)$, $(15.91, 20)$, $(12.73, 30)$, $(11.65, 40)$, $(12.06, 60)$, $(15.25, 80)$, $(19.36, 100)$, $(22.50, 120)$. The prototype vehicle had the following characteristics: mass 1785 kg, battery 62 kWh, 1.5 kW constant power consumption, 0.28 air drag coefficient and 2.44 m^2 front surface area, 0.8 propulsion efficiency and 0.8 recuperation efficiency.

Figure 7 plots estimated travel time and energy consumption for different departure times on the testbed with the results from TomTom shown for reference. The results show that the testbed is more conservative regarding travel time and energy consumption when leaving at non-congestion time. For the congestion hour, the testbed is more optimistic regarding travel time and energy consumption, and the generated uncertainty intervals are longer—for long trips interval length is approximately half an hour. The testbed provides different results for forward and backward trips as the model considers elevation details and recuperation.

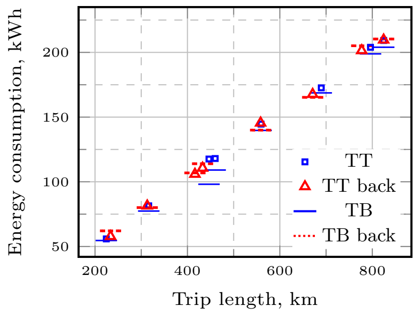
Figure 8 plots results of scalability tests, which were run on a Linux workstation with Intel(R) 16 Core(TM), i9-9880H CPU @ 2.30 GHz, 32 GB RAM with an equivalent remote database server. For each different trip length (air distance), 1000 source-destination pairs were generated, their routing was executed, and routes were saved in the database. The region was loaded into the main-memory KaTCH data structure, with approx. 21GB RAM used. The difference in the air distance and route length is app. 30%. The results show that the query cost without energy consumption calculation is almost constant, whereas energy computation grows linearly to the length of the path.



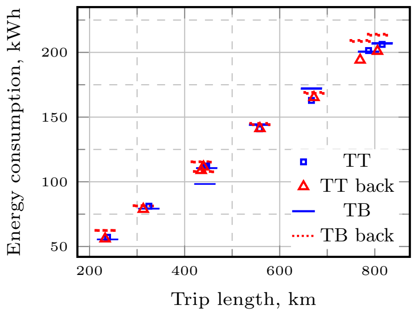
(a) departure at 00:00



(b) departure at 16:00



(c) departure at 00:00



(d) departure at 16:00

Fig. 7. Travel time and energy consumption for different departure times

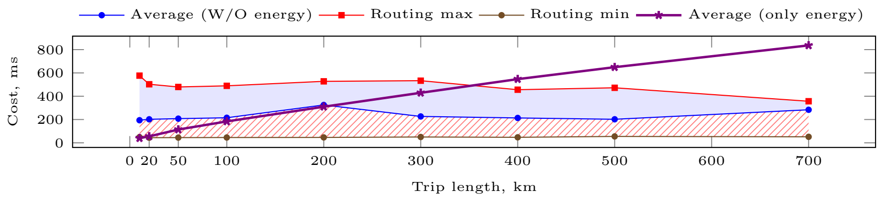


Fig. 8. Scalability tests

5 Conclusions and Future Work

Motivated by the inherent complexity of testing advanced routing algorithms, the paper proposes a testbed that integrates state-of-the-art tools and provides a systematic approach to available open-source data. We believe the provided insights and the testbed itself will shorten the preparation phase of future experimental studies. The scalability and reference-based tests demonstrate the merits of the testbed. The work can be extended in several directions, e.g. to enable functions to switch among EV energy consumption and life-cycle profiles or to enrich the environment with a flexible setup for experiments.

Acknowledgement. This project has received funding from European Regional Development Fund (project No 01.2.2-LMT-K-718-02-0018) under a grant agreement with the Research Council of Lithuania (LMTLT).

References

1. Åkerblom, N., Chen, Y., Chehreghani, M.H.: An online learning framework for energy-efficient navigation of electric vehicles. In: IJCAI, pp. 2051–2057. ijcai.org (2020). https://doi.org/10.24963/ijcai.2020/284
2. Baum, M., Dibbelt, J., Wagner, D., Zündorf, T.: Modeling and engineering constrained shortest path algorithms for battery electric vehicles. Transp. Sci. **54**(6), 1571–1600 (2020). https://doi.org/10.1287/trsc.2020.0981
3. Brinkhoff, T.: A framework for generating network-based moving objects. GeoInformatica **6**(2), 153–180 (2002). https://doi.org/10.1023/A:1015231126594
4. Chargeprice: Open EV Data. https://github.com/chargeprice/open-ev-data. Accessed 7 Mar 2021
5. Codeca, L., Frank, R., Faye, S., Engel, T.: Luxembourg SUMO traffic (lust) scenario: traffic demand evaluation. IEEE Intell. Transp. Syst. Mag. **9**(2), 52–63 (2017). https://doi.org/10.1109/MITS.2017.2666585
6. ElaadNL: Open data sets. https://platform.elaad.io. Accessed 12 Mar 2021
7. Geofabrik GmbH: Openstreetmap data extracts (2020). http://download.geofabrik.de/. Accessed 5 Sept 2020
8. German Aerospace Center (DLR) and others.: Sumo – simulation of urban mobility (2020). https://sumo.dlr.de/docs/. Accessed 24 Feb 2020
9. Institut fuer Theroretische Informatik, Karlsruher Institut fuer Technology (KIT): KaTCH - Karlsruhe Time-Dependent Contraction Hierarchies (2016). https://github.com/GVeitBatz/KaTCH. Accessed 16 Mar 2021
10. Jarvis, A., Reuter, H.I., Nelson, A., Guevara, E.: Hole-filled seamless SRTM data v4 (2008). http://srtm.csi.cgiar.org. Accessed 24 Feb 2020
11. Kurczveil, T., López, P.Á., Schnieder, E.: Implementation of an energy model and a charging infrastructure in SUMO. In: Behrisch, M., Krajzewicz, D., Weber, M. (eds.) SUMO 2013. LNCS, vol. 8594, pp. 33–43. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-45079-6_3
12. López, P.Á., et al.: Microscopic traffic simulation using SUMO. In: ITSC, pp. 2575–2582. IEEE (2018). https://doi.org/10.1109/ITSC.2018.8569938
13. Open Charge Map: The Open Charge Map API. https://openchargemap.org/site/develop/api. Accessed 7 Mar 2021
14. Pramanik, A., Rahman, M., Anam, I., Ali, A.A., Amin, A., Rahman, M.: Modeling traffic congestion in developing countries using google maps data (2020). preprint at arXiv:2011.02359
15. Russo, D., Roy, B.V., Kazerouni, A., Osband, I., Wen, Z.: A tutorial on Thompson sampling. Found. Trends Mach. Learn. **11**(1), 1–96 (2018). https://doi.org/10.1561/2200000070
16. Yu, J., Fu, Z., Sarwat, M.: Dissecting GeoSparkSim: a scalable microscopic road network traffic simulator in apache spark. Distrib. Parallel Databases **38**(4), 963–994 (2020). https://doi.org/10.1007/s10619-020-07306-x

17. Zafar, N., Haq, I.U.: Traffic congestion prediction based on estimated time of arrival. PLoS ONE **15**(12) (2020). https://doi.org/10.1371/journal.pone.0238200
18. Zhao, X., Spall, J.C.: Modeling traffic networks using integrated route and link data (2018). preprint at arXiv:1811.01314