Massive-Scale Models of Urban Infrastructure and Populations

Daniel Baeder, Eric Christensen, Anhvinh Doanvo, Andrew Han, Ben F. M. Intoy(✉), Steven Hardy, Zachary Humayun, Melissa Kain, Kevin Liberman, Adrian Myers, Meera Patel, William J. Porter III, Lenny Ramos, Michelle Shen, Lance Sparks, Allan Toriel, and Benjamin Wu

Deloitte Consulting LLP, Arlington, Virginia, USA
bintoy@deloitte.com

Abstract. As the world becomes more dense, connected, and complex, it is increasingly difficult to answer "what-if" questions about our cities and populations. Most modeling and simulation tools struggle with scale and connectivity. We present a new method for creating digital twin simulations of city infrastructure and populations from open source and commercial data. We transform cellular location data into activity patterns for synthetic agents and use geospatial data to create the infrastructure and world in which these agents interact. We then leverage technologies and techniques intended for massive online gaming to create 1:1 scale simulations to answer these "what-if" questions about the future.

Keywords: Simulation · Urban mobility · Pattern mining

1 Introduction

There is tremendous value to studying complex systems, but these systems cannot be understood simply by analyzing their individual components [2]. If a complex system cannot be studied analytically, an alternative option is to explore the system experimentally. However, these experiments may be difficult to perform due to cost or infeasibility. An easier, more rapid, and more scalable solution would be to instead model and then simulate the system.

As used in this document, "Deloitte" means Deloitte Consulting LLP, a subsidiary of Deloitte LLP. Please see www.deloitte.com/us/about for a detailed description of our legal structure. Certain services may not be available to attest clients under the rules and regulations of public accounting. This publication contains general information only and Deloitte is not, by means of this publication, rendering accounting, business, financial, investment, legal, tax, or other professional advice or services. This publication is not a substitute for such professional advice or services, nor should it be used as a basis for any decision or action that may affect your business. Before making any decision or taking any action that may affect your business, you should consult a qualified professional advisor. Deloitte shall not be responsible for any loss sustained by any person who relies on this publication.

R. Thomson et al. (Eds.): SBP-BRiMS 2019, LNCS 11549, pp. 113–122, 2019.
https://doi.org/10.1007/978-3-030-21741-9_12

Modeling and simulation are commonly used techniques for gathering data from, and making predictions inside of, complex systems. This is especially important if the data are sparse, private, or difficult to collect.

There are many cases where we can use simulation to model scenarios and their possible outcomes [1,4,11]. However, these simulations use static or random data to drive entity behavior. With data on the movements of individuals becoming more readily available, implementing realistic and anonymized human travel behavior into models is now a possibility [5].

Currently, it is neither efficient nor feasible to know the position of every single person in a large population. However, by using a subset of the population, it is possible to generate general travel schedules [8,15]. This is advantageous because it allows a user to simulate large populations without using personally identifiable information. It also gives a user the ability to generate different, but statistically similar, data sets in order to test the robustness of the system.

A city is more than its population. Cities also include infrastructure such as power, roads, telecommunications, water, etc. that are interconnected and are controlled by people or autonomous systems, that, in turn, are influenced by infrastructure. This high degree of connectivity combined with uncorrelated dynamics means that changes cascade through and across the layers in the system. We are able to simulate these interactions and visualize all of these different layers in a format that allows for user interactivity. To make the simulation tractable at scale, we distribute the micro-simulations in a cloud environment [3].

2 From Geolocations to User Travel Schedules

In order to simulate realistic human behavior, we aim to construct synthetic traffic patterns for individuals in a metropolitan area using geolocation data sourced from cellular phones. These patterns take the form of generated individual schedules of stationary locations with associated start and end time values. These schedules are then used in a cloud-based simulation environment to model traffic behavior. Our work builds off of a process that uses cellular call detail records to train an input-output hidden Markov model (IOHMM) to generate models of individual activity [8,15]. Following the methodology in [8], we do not make any native assumptions that user behavior will adhere to a standard pattern (e.g., home-work-leisure-home). Instead, we intentionally capture irregularities in travel patterns across individuals to model aggregate behavior more completely. However, our approach diverges from earlier work in two important ways. First, we use individual position data derived from a mobile location service that provides more frequent and precise information. Second, we import our generated individual travel schedules to a simulated urban environment through which we can capture effects on behavior stemming from environmental changes (e.g., adding or removing roads on which users can travel). Figure 1 illustrates the process of transforming raw data into synthetic individual user schedules.

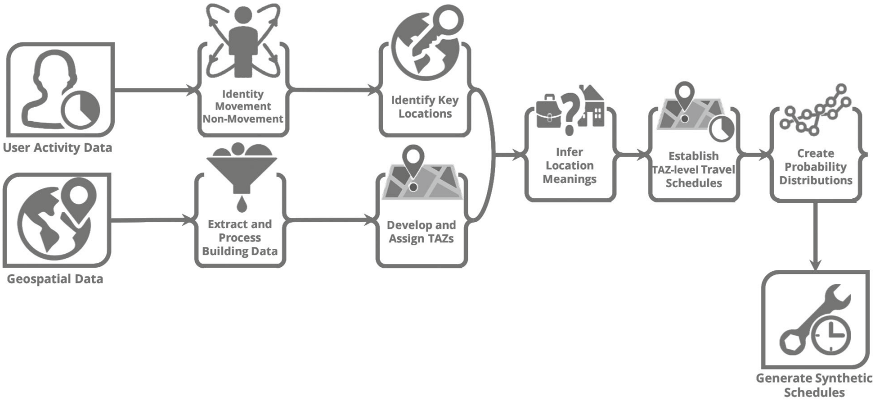
User Activity Data
Geospatial Data
Identity Movement Non-Movement
Extract and Process Building Data
Identify Key Locations
Develop and Assign TAZs
Infer Location Meanings
Establish TAZ-level Travel Schedules
Create Probability Distributions
Generate Synthetic Schedules

Fig. 1. Process of constructing synthetic user schedules

2.1 Identifying Stationary Periods

The raw geolocation data cover approximately 100,000 unique users per day. In their raw form, these spatio-temporal data points do not capture whether an individual is moving or stationary at any given point, nor do they record activity over time. We transform the data into time-ordered lists of start and end positions and times from which we can extract distances traveled and average velocities achieved.

To build user schedules from point-to-point data, we identify discrete periods of movement and non-movement. The simulation environment uses pathfinding algorithms that require defined start and end points, so we focus on isolating periods of non-movement, known as stay locations. A user's schedule of activities contains these stay locations and the amount of time spent at each. This process has the added benefit of significantly reducing the data size, as many data points can exist during a single movement or non-movement period.

Each user data point has an associated latitude and longitude position, but the positioning methodology used to create the raw data introduces some error. To account for this, we set a minimum threshold for detecting movement between data points at 0.05 miles. Travel distance less than that threshold over a period of time is classified as stationary behavior. We also control for several contingencies, including (1) instances where data points are so frequent that travel distance fails to exceed our threshold, and (2) short breaks in travel between periods of clear movement (i.e., an individual stopped at a traffic light), by classifying these periods as movement. There were 2,695,624 entries in our raw sample, which we reduced by 89% to 293,759 entries after completing this process. Even so, this initial data reduction is intended to be conservative in order to retain as much user activity data as possible.

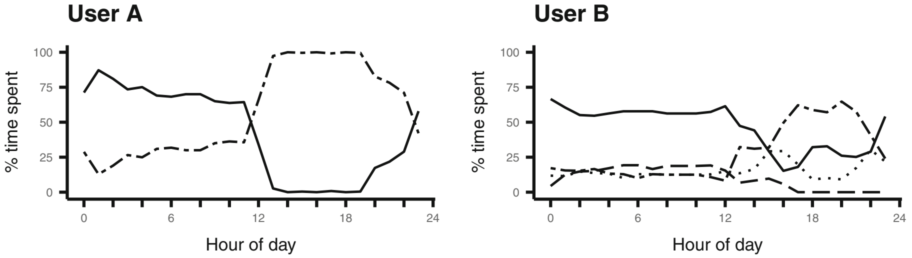

Fig. 2. Behavioral profiles of weekday stay locations for two users. User A has a clear two-location profile, with possible home and work locations. User B has a complex profile, with possible home and three work/leisure locations.

2.2 Identifying Key User Locations

We use a density-based clustering algorithm (DBSCAN) to cluster multiple data points that reflect a single user location but appear as multiple points due to geolocation error. The DBSCAN algorithm is well-suited for this problem because it operates independently of any preconceived notion of a system's number of clusters and the shapes of those clusters [12]. We parameterize the algorithm by specifying the maximum allowable radius (r) from each seed point (s) for points (p) to be considered part of the same cluster (C) defined in Eq. 1. We also specify the minimum percentage of the user's time (t), defined in Eq. 2, that is captured in the data for a group to be considered a cluster. Thus, points which meet the following criteria are combined into clusters:

$$C_r(s) : \{p \mid d(s,p) \leq r\}, \tag{1}$$

$$t_C \geq 0.1 * t \quad \forall_C, \tag{2}$$

where t_C is the time spent in cluster C and $d(s,p)$ is the great-circle distance between s and p.

The clustering process provides us with two classes of stationary behavior: clustered locations where user data points appear repeatedly and irregular travel locations where users spent time, but not with enough frequency to be captured as a cluster. From this output, we are able to link key user locations to time periods and generate behavioral profiles of users based on the percentage of each hour of the day that a user spends in each of their stationary locations. We are able to generate these behavioral profiles for any user in the dataset, as exemplified in Fig. 2, and, at this point, can derive user schedules without location labels for use in the simulation environment.

2.3 Inferring Cluster Meaning

User stay location clusters are coded numerically to indicate their importance to the user in terms of total time spent in each cluster, but the meaning of each cluster might differ across users due to high variability in the dataset. We specify a Gaussian mixture model (GMM) to group clusters across all users by temporal characteristics. A GMM is an appropriate choice in situations where there is a strong assumption of underlying groupings in the data, but those groups are unlabeled and unobserved. The algorithm iterates over an expectation-maximization process to assign unlabeled observations to clusters based on the probability of each point belonging to a cluster given its variance-adjusted distance from that cluster centroid. We conservatively estimate to expect two clusters, home and not-home, opting to handle irregular travel locations separately.

We link each stay location cluster to an individual building to provide further context to user activities. We use open source data on tax parcels to extract centroid positions for each one [13]. We then estimate building type as being "residential" or "non-residential" based on the percentage of total residential square footage indicated for each parcel. Lastly, we link each user stay location, both clustered and irregular travel points, to a building using a nearest neighbor approach, with distances between points calculated based on the World Geodetic System Ellipsoid. Thus, if a user's primary stay location shares temporal characteristics with locations we label as home, and is also linked to a residential building, we can more confidently say that location is the user's home. Accurate labeling of primary user locations is critical to properly train the IOHMM. A rules-based approach to home and work location labeling would make sense in regions where tax parcel data is unavailable

2.4 Assigning User Stay Locations to Traffic Analysis Zones

To simultaneously abstract from specific user locations and reduce our risk of overfitting, we establish traffic analysis zones (TAZ), opting to use Census geographies in order to simplify our overall methodology [14]. The selection criteria for appropriate TAZs is an objective function defined by a minimal overall number of potential user locations and a maximal retention of user transition information (i.e., avoiding recursive transitions whenever possible). We find that Census blocks are the optimal choice given these criteria (see Table 1). The result is a reduction in the location feature space by 95% with a small loss in user transition detail, as measured by the percentage of time captured in each user's top ten stay locations or zones. Census blocks are also highly homogeneous in terms of building type included within each block, which protects against user transition information loss, since individuals often travel between residential and non-residential locations. Lastly, the use of Census blocks could allow for a more in-depth investigation of the relationship between the travel patterns and socioeconomic characteristics of users since Census demographic and housing data is easily attached to block polygons.

We link each user stay location to a TAZ based on the position of the stay location in the case of irregular travel, or based on the position of the cluster centroid in the case of clustered stay locations returned from the DBSCAN algorithm. This results in the aggregation of close by irregular travel locations for each user to a single TAZ, capturing broader areas of frequent activity. We

Table 1. Traffic analysis zones. The third column is the percentage of user time captured in the top ten locations or zones. The fourth column is the percentage of zones with high building type concentration (95% or greater of one type).

Geography type	Feature size	User time captured	High building concentration
Building layer	1,373,831	0.847	1
Census blocks	63,305	0.925	0.734
Census block groups	3,238	0.934	0.485
Census-defined TAZs	1,114	0.935	0.376

then condense the data to a list of distinct locations with start and end times for each user, along with labels of estimated location meaning. From this matrix, we can calculate the distribution of times during which users transition between stay locations, both at the individual and sample population levels. We also calculate the geographic distribution of users' primary cluster locations. We will extend this methodology to a generative approach utilizing an IOHMM architecture as in [15].

3 Simulation Architecture

In order to achieve the scale, density, and interconnectivity present in modern cities in the simulation, we must distribute it over multiple compute nodes. For each simulation layer, we assign several computers to process and implement updates to the entity states at each time step. The world is partitioned into disjoint regions that are assigned to a computer and are identified for exclusive *write* access. A computer can change the state of all the entities, which may or may not interact with each other, within their *write* partition.

Areas on and near the boundary between *write* partitions contain information that is potentially relevant to state changes of entities that may not be within the same *write* partition. We identify these areas as areas with *read* access, where computers send information about state changes to other computers that have *read* access over those entities. When interactions and updates only depend on local information, a cloud distribution of this sort will scale easily to larger geographical areas.

To understand how our simulated population effects, and is affected by, the rest of its environment, we create a digital twin and model cars, traffic signals, telecommunications, and power at a microscopic level. Each car is a simulation agent that follows a set of logic either dictated by rules or by artificial intelligence (AI). By using the travel schedules generated in the previous section, the cars travel from destination to destination, routing between activity points via the road network.

We use OpenStreetMap (OSM) [10], an open- and crowd-sourced database, to generate the network of roads, buildings, and traffic lights in which the entities exist in the simulation. Each road in the network is assigned a weight that

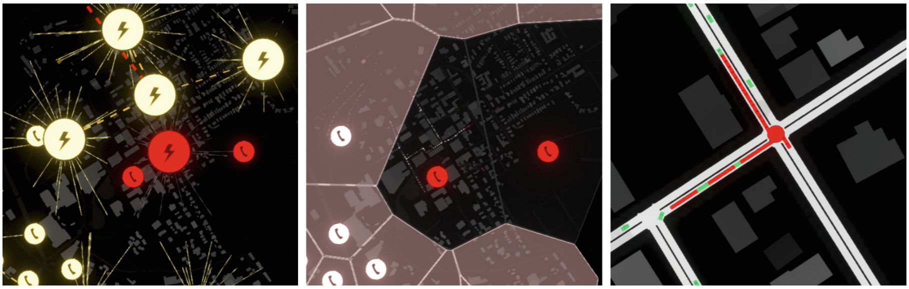

Fig. 3. Example images of power failure cascade. The leftmost image contains power and telecommunication nodes, with cartoon thunderbolt symbols representing power nodes, the phone symbols representing telecommunication nodes, and red denoting a node that is not working. The center image contains telecommunication nodes, with Voronoi areas with service in a lighter color and areas without in a darker color. The rightmost image is an intersection with cars (red and green rectangles along the roads) backing up due to a powered down traffic signal and loss of navigation. (Color figure online)

can take the form of the length or the expected time to traverse. A hierarchical bi-directional A-Star algorithm [6] provides the optimal path subject to either metric. As these cars traverse the road network, they stop at traffic lights and stop signs which are parsed from OSM. They are also restricted by a desired follow distance to the car in front of them and follow empirically derived kinematics [7]. We take the location of telecommunication towers from the OpenCellID database [9], and infer the power network structure from the building distribution given by OSM.

We can add dynamic, human-in-the-loop interaction with power and telecommunication entities, and changes to the road network. When users disable power network entities, the dependent traffic lights and telecommunication antennas are disabled (see example images in Fig. 3). In our simulation, cars connected to disabled telecommunication antennas change their routing behavior to prioritize high throughput roads and the disabled traffic lights toggle to behave as inefficient stop signs.

4 Using Data to Drive Entity Behaviors

We can use the empirically derived data to drive the entity behaviors. We achieve this by importing user schedules generated from geolocation data (detailed in Sect. 2) to dictate the agents within the simulation (detailed in Sect. 3). By using these generated data, the number of agents within the simulation can exhibit realistic behavior at any scale. This can lead to non-trivial, complex, and emergent behavior, such as rush-hour traffic, that may not be seen in simpler models.

The user can then perturb the system. An obvious example is to alter the road network and observe the differences of the traffic flows and patterns before

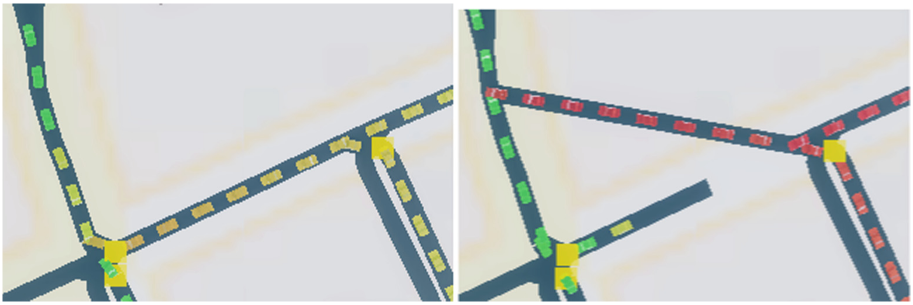

Fig. 4. A before and after example of rerouting by adding and removing a road.

and after the change. Figure 4 shows how traffic is rerouted with the addition and removal of roads. Changes to the road network can be adding or removing roads, altering the speed limit of a road, and automated adjusting of stop light timing. We can schedule changes to happen at a predetermined time during the simulation, allowing the study of shock events.

Just as we can perturb the environment, we can perturb the agents themselves. A simple example is an event where a subset of the agents converge on a location, which would approximate a sporting event, concert, political rally, etc. Another example would be to change the AIs of the cars. This could include a mixture of self-driving and human-driven cars and examining how their combined presence would impact traffic patterns. An advanced perturbation would be infrastructural in nature. An example would be a loss of the electrical power layer, which then causes failures in telecommunication and loss of navigation abilities for car agents. This causes the cars to either take less efficient paths or prefer commonly-known roads to reach their destinations.

Data are collected from the simulation following these perturbations, such as vehicle travel times, speeds, and distances. These data are gathered on a system-wide scale (such as an average speed) or on a local scale (such as the traffic volume of a single road). We can also gather data on an individual level (i.e. a single car). We can then read the database to produce real time level of service metrics. Examples of this include car density and road congestion heat maps, and power and telecommunication network loads.

5 Summary and Discussion

Because of its ability to provide analysis and insight both with, and in the absence of, previously existing data, we can use the simulation to answer questions about the future. These "what-if" questions are varied, from natural disaster planning ("What happens if we face a major storm and are forced to evacuate millions of people?") to military preparedness ("What are the mission impacts if we lose the ability to operate from certain roads or ports?") to business modeling

Table 2. Potential use cases

Sector/Industry	Use case
Defense	Evacuating an area in the event of a natural disaster or man-made disruption
Supply chain	Moving people, equipment, and supplies around the world in a dynamic supply, demand, and transportation network
Urban planning	Managing the impacts of massive area changes (ex. flooding, migration)
Urban planning	Managing the impacts of new construction projects (housing, commerce, industry)
Transportation	Improving mobility in a metropolitan area through changes to infrastructure, public transportation, and public-private partnerships
Transportation	Investigating consequences of a road closure during a large event (ex. the Olympics, a presidential visit, the building of a stadium or theme park)
Telecom	Optimizing the deployment of 5G infrastructure to serve cellular customers today, and self-driving vehicles and IoT devices tomorrow

("What are the implications if our company implements self-driven ride sharing or drone delivery?").

We build a digital twin of a city using information about building locations, cell towers, and the electrical grid. We drive entity behavior with user travel schedules generated from geolocation data. These data, captured from cellular phones, are initially analyzed to identify periods of movement and non-movement. We apply density-based clustering to group nearby points of non-movement into single locations if certain criteria are met. Next, we infer location labels (home or not-home) by temporal characteristics along with the inferred use type of the nearest building. Next, we link all stay locations to Census blocks to abstract away from exact coordinate locations, and calculate transition time and geographic distributions across the dataset.

With the help of cloud computing, we can run the simulation and model many infrastructural layers of a city, including traffic, telecom, and the electrical grid. Lastly, we can perturb the system by introducing some new future scenario to see how the system responds. The data and insights gleaned from this process can allow policymakers and corporations to take proactive measures to exploit opportunities and minimize vulnerabilities in the future. See Table 2 for potential use cases.

In the past, reactive and suboptimal decisions were often made because a problem was not anticipated, or because it was too difficult to prepare for. Even when ample time is available to make a careful decision, the interconnectedness of our world makes it difficult to anticipate the full impact of each option, or to identify the best one. It would not be safe or feasible to shut off the electrical grid over a large portion of Manhattan to observe how the telecom layer reacts, or to build new roads in Detroit for a traffic simulation and then remove them a few minutes later. However, the simulation platform allows these types of scenarios to be reproduced and tested safely and quickly. With the help of the capabilities described in this paper, these types of simulations are no longer limited by scale, complexity, or ability to consider collateral and cascading effects. The simulation platform allows decision makers to take action in a manner that is proactive rather than reactive. By combining widely available data, cloud computing, agent-based simulation methods, and data science, we demonstrate a new approach for proactive planning and decision making where future scenarios can be evaluated in a high-fidelity virtual environment at low risk and cost.

References

1. Adiga, A., Marathe, M., Mortveit, H., Wu, S., Swarup, S.: Modeling urban transportation in the aftermath of a nuclear disaster: the role of human behavioral responses. In: The Conference on Agent-Based Modeling in Transportation Planning and Operations. Citeseer (2013)
2. Anderson, P.W., et al.: More is different. Science **177**(4047), 393–396 (1972)
3. Attiya, H., Welch, J.: Distributed Computing: Fundamentals, Simulations, and Advanced Topics, vol. 19. Wiley, Hoboken (2004)
4. Barrett, C., et al.: Cascading failures in multiple infrastructures: from transportation to communication network. In: 2010 5th International Conference on Critical Infrastructure (CRIS), pp. 1–8. IEEE (2010)
5. Barrett, C.L., Eubank, S., Marathe, A., Marathe, M.V., Pan, Z., Swarup, S.: Information integration to support model-based policy informatics. Innov. J.: Public Sector Innov. J. **16**(1) (2011). https://www.innovation.cc/volumes-issues/vol16-no1.htm
6. Goldberg, A.V., Kaplan, H., Werneck, R.F.: Reach for a*: efficient point-to-point shortest path algorithms. In: 2006 Proceedings of the Eighth Workshop on Algorithm Engineering and Experiments (ALENEX), pp. 129–143. SIAM (2006)
7. Kesting, A., Treiber, M., Helbing, D.: Enhanced intelligent driver model to access the impact of driving strategies on traffic capacity. Philos. Trans. R. Soc. Lond. A: Math. Phys. Eng. Sci. **368**(1928), 4585–4605 (2010)
8. Lin, Z., Yin, M., Feygin, S., Sheehan, M., Paiement, J.F., Pozdnoukhov, A.: Deep generative models of urban mobility. IEEE Trans. Intell. Transp. Syst. (2017)
9. OpenCell ID. Data. http://opencellid.org/downloads. http://opencellid.org
10. OpenStreetMap contributors: Planet dump (2017). https://planet.osm.org. https://www.openstreetmap.org
11. Ren, Y., Ercsey-Ravasz, M., Wang, P., González, M.C., Toroczkai, Z.: Predicting commuter flows in spatial networks using a radiation model based on temporal ranges. Nat. Commun. **5**, 5347 (2014)
12. Simoudis, E., Han, J., Fayyad, U.M. (eds.): A density-based algorithm for discovering clusters in large spatial databases with noise. Proceedings of the Second International Conference on Knowledge Discovery and Data Mining. AAAI Press (1996)
13. Southeast Michigan Council of Governments: Data portal. http://maps-semcog.opendata.arcgis.com/datasets
14. US Census Bureau: Tiger/line shapefiles. https://www.census.gov/cgi-bin/geo/shapefiles/index.php
15. Yin, M., Sheehan, M., Feygin, S., Paiement, J.F., Pozdnoukhov, A.: A generative model of urban activities from cellular data. IEEE Trans. Intell. Transp. Syst. **19**(6), 1682–1696 (2017). https://doi.org/10.1109/TITS.2017.2695438