# SMARTER PARKING - USING AI TO IDENTIFY PARKING INEFFICIENCIES IN VANCOUVER


**Devon Graham**
Department of Computer Science
University of British Columbia
Vancouver, BC
drgraham@cs.ubc.ca

**Satish Kumar Sarraf**
Department of Electrical and Computer Engineering
University of British Columbia
Vancouver, BC
ssatish@ece.ubc.ca

**Taylor Lundy**
Department of Computer Science
University of British Columbia
Vancouver, BC
tlundy@cs.ubc.ca

**Ali MohammadMehr**
Department of Computer Science
University of British Columbia
Vancouver, BC
alimm@cs.ubc.ca

**Sara Uppal**
Department of Electrical and Computer Engineering
University of British Columbia
Vancouver, BC
sarau@ece.ubc.ca

**Tae Yoon Lee**
Department of Statistics
University of British Columbia
Vancouver, BC
taeyoon.harry.lee@gmail.com

**Hedayat Zarkoob**
Department of Computer Science
University of British Columbia
Vancouver, BC
hzarkoob@cs.ubc.ca



April 30, 2019



## ABSTRACT

Drivers searching for on-street parking are distracted, reducing the safety of all road users, and cause unnecessary congestion of city streets. Taking such drivers off the road is a natural goal in any modern municipality. To that end, we estimate the search time for a vehicle looking for an on-street spot and the time for a driver who simply parks in the nearest lot and walks back. We compare these times for each block in Vancouver's downtown core and identify areas where parking in nearby lots can actually save drivers time over searching the streets for a spot. To identify these areas, we use deep learning to estimate the probability of being able to park in each block and build a simulation of drivers searching for parking both on- and off-street. Since lots often have excess capacity, these areas provide an opportunity to repurpose valuable curbside space for community-friendly uses more in line with the City's transportation goals.



***K*eywords** Parking · Street Parking · Deep Neural Networks · Occupancy Prediction · Parking Simulation


## 1 Introduction

In many urban areas, trying to find curbside parking can be an unpleasant experience. In peak hours, available spots are rare and drivers can cruise from block to block without finding one. Even having found a spot, drivers may then need to walk a significant distance back to their destination. This takes time and clogs up city streets with unnecessary congestion. In fact, in a well-known study, Shoup found that on average 30% of traffic on city streets can be the result of such cruising (Shoup, 2006). This excessive congestion has many disadvantages. Drivers searching for parking are distracted drivers, making the streets unsafe for pedestrians and cyclists (Edquist *et al.*, 2011). With more cars on the road, travel times are increased for everyone, including public transit vehicles. More slow-moving cars on roads means increased emissions (Rouphail *et al.*, 2001). Thus, it is important for modern cities to manage their parking resources efficiently, and Vancouver is no exception.

As an alternative to curbside parking, drivers have the option to park in off-street lots. Such lots can be found throughout the city and usually have many available spots. However, parking off-street has its own drawbacks. Drivers need to first find the lot, find a free spot, pay, and then walk back to their destination. It is therefore not immediately clear which of on- or off-street parking is the better choice for drivers in terms of saving time.

While previous studies have examined the trade-off between searching for curbside parking and parking in the nearest lot, most consider different pricing schemes to make their arguments (Ottosson *et al.*, 2013; Shoup, 2006). In particular, Shoup looked at how drivers might think about on- and off-street by assuming that curbside parking is free but rare, while off-street spots are always available, but are expensive (Shoup, 2006). He argues that if the time it takes for drivers to cruise for free curbside parking exceeds a certain threshold, people will prefer paying for off-street parking to save time. However, price is not the only factor affecting drivers' decisions (Yan *et al.*, 2019), and people trade time for money differently depending on their situations: if you have an important meeting with a client, you will gladly pay more for parking than be late and risk loosing the client. If you've bought an expensive concert ticket, paying a high price for parking is not as bad as missing part of the show.

Our work approaches the problem of curbside vs. lot parking from another angle: we consider only the difference in time between the two methods, and think of price as a tool that can be used to change driver behaviour. Because all the lots we consider are owned by the City of Vancouver, adjusting prices in these lots is within the power of the City's decision makers. Thus, price is a mechanism by which the city can encourage drivers to park in off-street lots more often. We use cutting edge AI tools along with real-world data to build realistic models of on- and off-street parking procedures. We show that despite the costs of off-street parking mentioned above, drivers can sometimes save time by parking in off-street lots rather than cruising in search of on-street parking.

In order to simulate cruising for parking, we predict curbside parking occupancy from observed payment data using a deep neural network. Such predictions can be problematic because aggregated payment data aggregation can be a poor representative of actual occupancy rares (Ottosson *et al.*, 2013). For example, drivers can leave before or after the time for which they paid for parking. The reason for using deep neural networks to predict occupancy from payment data, instead of already existing methods in the literature, is that our data was not as complete as what the existing methods require (Rajabioun and Ioannou, 2015; Tamrazian *et al.*, 2015; Yang and Qian, 2017; Yang *et al.*, 2019). While curbside payments are performed via on-street meters which accept credit/master card, cash, and app payments, we only had access to app payments. Also, we were only provided with a limited real occupancy checks which were performed by in person observations. Deep neural network is a strong AI tool to predict desired quantities from such in-complete datasets.

In the remainder of this paper, we first give a brief literature review in Section 2. Then, we describe the different methods that were used in this work to simulate both on- and off-street parking in Section 3. Sections 4 and 5 describe the data used in and the results of our simulations. Finally, we conclude the paper in Section 6, where some future works are also discussed.

## 2 Literature Review

Others have previously considered methods for simulating the parking search procedure, as well as comparing on- and off-street parking, and predicting occupancy from payment data. We review some relevant works below.

### 2.1 On-Street Parking Simulators

Mannini *et al.* (2017) assert that there has been a lack of attention to on-street parking time estimation in the literature. Mannini *et al.* (2017) compute on-street parking search time by using Floating Car Data obtained from probe vehicles. Belloche (2015) use survey-based data to compute the parking search time based on an off-street parking search time model proposed by Axhausen *et al.* (1994). We contribute a simulation based approach to this body of literature. Our approach is inspired by Dowling *et al.* (2019). Dowling *et al.* (2019) view streets as a network, where streets and intersections are represented as edges and nodes, respectively. Then, using the transaction data to estimate true occupancy, Dowling *et al.* (2019) calculate various network-level measures such as congestion. Our work uses transaction data to estimate true occupancy using a deep learning model and estimates the parking search time with a simple search algorithm using Monte-Carlo simulation.

### 2.2 Off-Street Parking Simulators

While there exist agent-based choice models for parking in general (Vuurstaek *et al.*, 2018; Benenson *et al.*, 2008; Waraich and Axhausen, 2012; Bischoff and Nagel, 2017) and survey-based studies for off-street parking search time (Axhausen *et al.*, 1994; Teng *et al.*, 2002), literature on explicitly modeling parking in an off-street parking lot is quite limited. We propose a simple simulation-based approach to compute the time it takes for a car to park in an off-street parking lot.

### 2.3 On-Street vs. Off-Street Parking

There has been active research in parking policy and regulation for both on-street parking (Marshall *et al.*, 2008; Biswas *et al.*, 2017) and off-street parking (Barter, 2010). On-street parking is evaluated on a combination of three areas: safety for drivers, cyclists, and pedestrians; road congestion; and economic activity (Marshall *et al.*, 2008). Off-street parking lots are created as extra supply that accommodate excess on-street parking demand. Studies have suggested using pricing to manage the markets for on-street and off-street parking individually (Ottosson *et al.*, 2013). However, these studies do not offer a collective assessment of on-street and off-street parking. Our own work uses time as a common metric to compare on- and off-street parking and identify key areas where it takes less time to park off-street than on-street.

### 2.4 Parking Occupancy Prediction from Transaction Data

A naive method of estimating parking occupancy from transaction data is to divide the active transactions by the total number of occupancy. In (Fiez and Ratliff, 2017; Dowling *et al.*, 2019), parking occupancy is estimated this way at a block-face level over the course of an hour. They acknowledge the limitations of this method, particularly because drivers tend to leave before or after their paid parking time. Often it is the former case, leading to over-estimation of occupancy. In recent years, statistical and machine learning models have been developed to deal with such issues and predict parking occupancy more accurately using additional data. These include multivariate spatial-temporal models (Rajabioun and Ioannou, 2015), unsupervised clustering (k-means and k-nearest neighbours) (Tamrazian *et al.*, 2015), probabilistic model (Yang and Qian, 2017), and most notably a series of deep learning approaches. (Yang *et al.*, 2019; Zheng *et al.*, 2015; Kepaptsoglou *et al.*, 2014)

These deep learning approaches rely heavily on the increasing amount of actual occupancy data from sensors and surveys. Both (Yang *et al.*, 2019) and (Kepaptsoglou *et al.*, 2014) explore more complex network structures (recurrent neural nets) to predict occupancy rates from complete transaction data. Another approach by (Zheng *et al.*, 2015) uses the data from SFPark (approximately 2 million block-level samples) to construct a neural network whose performance is compared to a regression tree and support vector machines. In contrast to these works our training data is sparse in two ways. First, our transaction data is not complete as we have access only to a single mode of payment which comprises about 60% of total meter transactions. Second, the observations used for our ground truth occupancy levels are based on human surveys rather than sensors, giving us a selection of samples which is significantly smaller than other approaches and not evenly dispersed over the course of a day. In Section 3, we propose another deep learning model to estimate and predict parking occupancy based on two data sources: incomplete transaction data from parking meters and in-person counts.

## 3 Methods

### 3.1 On-Street

We construct a simulation of the time it takes to park on-street by modeling the behaviour of a driver searching for a spot. To generate accurate search times, we first need predictions of the availability of parking at each individual block of the city, for every hour of the day. To generate these predictions we use a deep neural network trained to predict occupancy from payment data. In this section, we first give some brief background on feed-forward neural networks. We then described the specific network we construct to solve this problem, and finally, we describe how we use these predictions to simulate search times for on-street parking.

### 3.1.1 Neural Networks

Neural networks are a broad class of functions that map inputs to outputs in a complex fashion. These networks are built from a sequence of "layers", where the output of each layer is fed as the input to the next. Each layer consists of a non-linear function $\sigma$ applied to a linear transformation of its input. The linear transformation in each layer is parameterized by a weight matrix, $W \in \mathbb{R}^{n,m}$, and a bias vector, $b \in \mathbb{R}^m$. Thus, a neural network is just a function of the form

$$f(x) = W_k\ \sigma\Bigg(W_{k-1}\ \sigma\Bigg(\ \dots\ \sigma(W_2\ \sigma(W_1 x + b_1) + b_2) \dots \Bigg) + b_{k-1}\Bigg) + b_k, \tag{1}$$

where $x$ is the input. Neural networks can approximate any function to arbitrary precision, if the matrices $W$ are chosen to be large enough. Given inputs $x$ and target values $y$, training the network means finding values of the parameters $W$ and $b$ for each layer such that the network's output, $f(x)$ is close to the true, known output $y$.

### 3.1.2 Occupancy Prediction Network

The data we are working with is a partial recording of payments made at meters in each block face in the city. However, inferring occupancy from payment information is not as straightforward as it may seem. Drivers may leave before their time is up, in which case payment data overestimates occupancy. Alternatively, another driver may subsequently park in and pay for the same spot, in which case payment data will underestimate occupancy. In order to try and learn a relationship between payments and occupancy, we train a neural network to predict the true occupancy of a block given payment data for that block.

For each block face and time, our model is given two input values which are derived from the payment data. We augment this with two input values that do not directly depend on the payment data, but relate to the block face in question. The input values are represented as a vector of length 4 consisting of the following values:

1. The total number of people who have paid for parking in the block face for that time,
2. The total number of people who have paid for parking on that block face in the 3 hour period before that time (i.e., how "popular" is the block in terms of parking),
3. The predicted time to drive the length of the block divided by the length of the block (i.e., how congested is traffic in the block), and
4. The number of meters on the block.

For each block and time, the neural network outputs a vector of length two, whose elements are forced to sum to 1 and are thus interpretable as probabilities. To train the network, we minimize the cross-entropy loss between these probabilities and the true, observed occupancy state of the block. In this way, the network learns to approximate the true probability of being able to park in a block at a given time, based on the data that we had available. Using the notation of Section 3.1.1, our network uses three weight matrices of sizes $3 \times 30$, $30 \times 30$, and $30 \times 2$ respectively (with corresponding bias vectors). As a non-linear function we use ReLU: $\sigma(x) = \max(0, x)$. We train the network on ten different random training and validation splits, with the validation set always being 20% of the total data, and iterating over the training data 150 times for each split.

Once the network is trained, we predict the existence of an empty meter at a given time (and thus whether a driver can park) in any block by simply feeding the model the payment and block-level features. Since the model has been trained to reproduce the probability that there will be an available spot, we can use this output directly in our simulation. To evaluate our network's performance, we compare its results to a logistic regression model that is given the same input variables our network is given. We evaluate both the network and the regression in terms of accuracy (with threshold $0.5$) and cross-entropy (see Section 5.1 for results).

### 3.1.3 On-street Search Simulator

We construct a representation of downtown Vancouver using OSMnx (Boeing, 2017), a Python package for modeling road networks. Intersections are represented by "nodes" and streets are represented by "edges" connecting pairs of nodes. Using the probabilities predicted by the neural network, we model the on-street parking search process for each block face, as shown in the flow diagram in Figure 1. We assume that there is a minimum amount of time, $t_{min}$, that is required to park at any on-street spot. This includes the time it takes to parallel park and pay at the meter. We set $t_{min} = 210$ seconds, following Shoup (2006). The total simulated search time is the sum of three factors: the minimum search time, the time spent driving on nearby streets while searching, and the time to walk back to the destination. During the search process, for any edge the vehicle traverses, we say it can park on that block with the probability predicted by the neural network. If it parks, we end the search process and add the time it takes to walk back to reach a

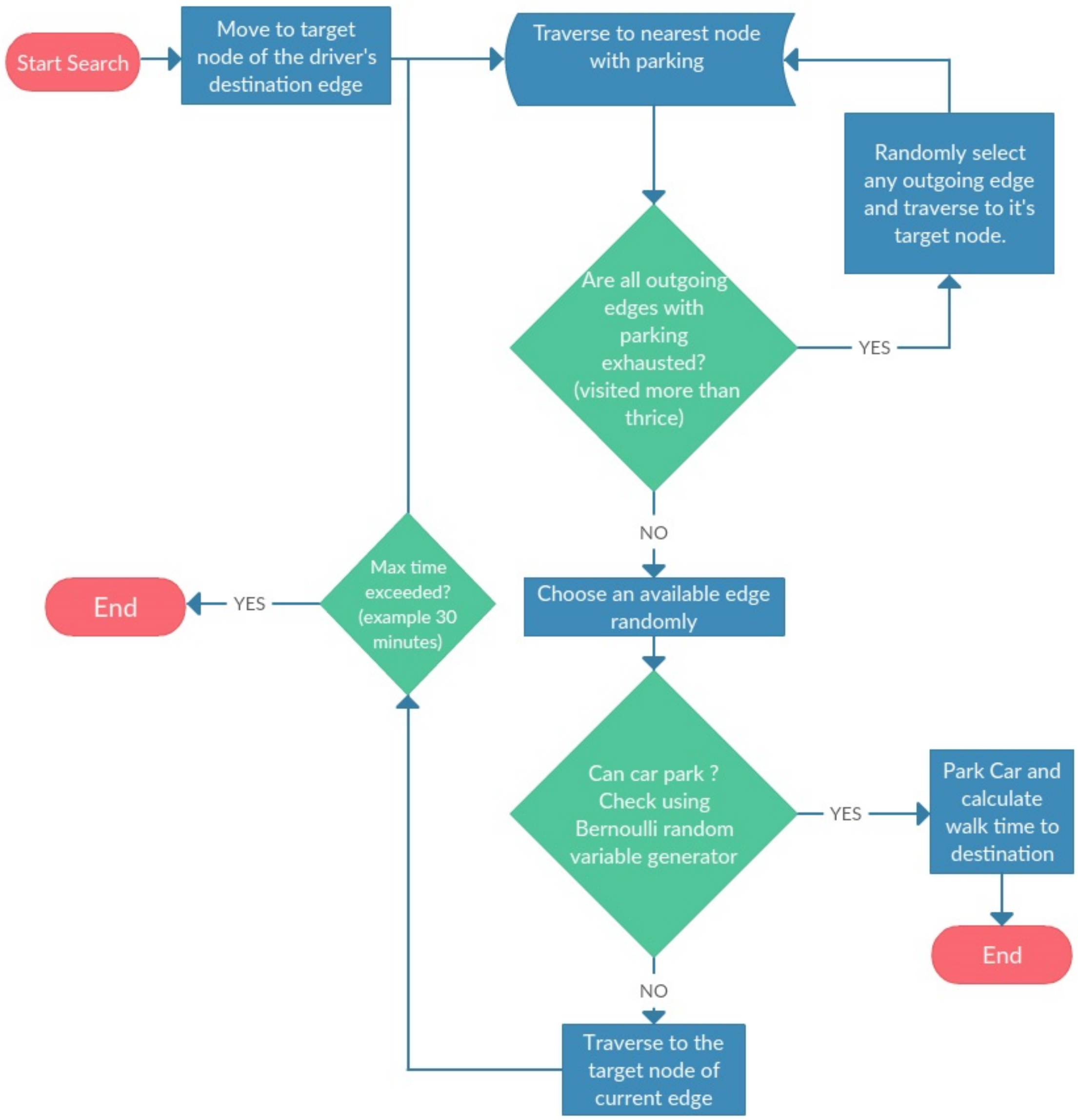


Figure 1: *Flow diagram showing the simulation process used to measure total search time when parking on-street. In the diagram, "edges" refers to block faces, and "nodes" refers to intersections. The search process starts in the top left and either ends in the bottom right if the vehicle finds a spot, or at the left if the maximum search time is reached.*

final search time. If the vehicle fails to park, it moves to the next edge and tries again (see Fig. 1 for details). Driving and walking times on roads are retrieved from the Google Maps Directions API. For a single vehicle, let $d_1, d_2, ..., d_n$ be the drive times for the blocks the vehicle drives, and $w_1, w_2, ..., w_m$ be the walking times for blocks the driver walks on[1]. Then the total on-street search time, $T$, for this vehicle, is modeled as

$$T = t_{min} + \left(\frac{d_1}{2} + \sum_{i=2}^{n-1} d_i + \frac{d_n}{2}\right) + \left(\frac{w_1}{2} + \sum_{j=2}^{m-1} w_j + \frac{w_m}{2}\right). \quad (2)$$

With each block face and for each time of the day, we use the simulator to produce a number of samples of the value $T$, and then average to produce the estimated search time. We divide the driving and walking times for the first and last blocks by two since we don't know where exactly in the block the driver starts or ends up.

[1]Note that walking time does not depend on time of the day.

### 3.2 Off-Street

We estimate the time it takes to park in the closest off-street lot for each block face in the city. To do so, we find the expected time it will take to drive to that lot, approximate the time spent in the lot searching for a spot, and find the time to walk back to the block face. To get the expected time for driving to the closest lot and then walking back at each particular time of day we simply make a request to the Google Maps Directions API. So these times are based on historical real-world data. To estimate the time spent searching in the lot, we use a Monte Carlo simulation (described below) to model the behaviour of people parking in each off-street lot that we considered.

#### 3.2.1 Time Spent Searching in Lots

Our simulation of the time spent in a lot is based on entry and exit data collected from several city-owned lots. For simplification, we assume that all lots are 1 dimensional structures: they have a single entrance and exit, and no branching paths. We assume there is some minimum amount of time that people will always take to park in a lot ($t_{min}$). In practice we set $t_{min} = 60$ seconds. We think this is an over-estimation, which, for fairness' sake, is least helpful for our argument. We assume that parkers will take the first available spot that they find, and that they will wait 30 seconds for a departing car and then take that spot if they can. We model the time a driver spends in the lot as being influenced not only by the number of vehicles currently in the lot, but also by the number of other vehicles entering and exiting the lot at the same time.

We consider a small unit of time (in practice, 60 seconds) and model the number of cars to arrive and to depart in that unit of time as Poisson distributed random variables with means $\lambda_a$ and $\lambda_d$, respectively. The parameters $\lambda_a$ and $\lambda_d$ are determined from the observed entry and exit data for each lot and each hour of the day and for each day of the week, averaged over 12 weeks. We keep track of the state of the lot by updating each spot in each unit of time. When vehicles depart, we assume they do so uniformly at random from the vehicles currently in the lot.

We assume that vehicles park in order of their arrival. We use a constant time $t_1$ as the time it takes to drive past a single parking stall. The first vehicle to arrive simply drives to the first vacant stall and parks there, waiting for the first departing vehicle if there is any. Subsequent vehicles to arrive may be slowed down by the vehicles in front of them. The $k^{\text{th}}$ vehicle to arrive may wait for up to $k$ vehicles to vacate stalls, or may wait for none, so we assume that on average it waits for half of them.

Because the minimum time $t_{min}$ includes paying for parking, and unlike pulling into spots, not all vehicles can always pay at the same time, we assume that the $k^{\text{th}}$ vehicle must also wait 1/2 of the minimum time for the vehicle in front of it, 1/4 of the minimum time of the second vehicle in front of it, 1/8 of the minimum time of the third vehicle, etc.

Then, if $N_a \sim_d Poisson(\lambda_a)$ vehicles arrive and $N_d \sim_d Poisson(\lambda_d)$ vehicles depart in this small period of time, the $k^{\text{th}}$ vehicle to arrive waits for $T_k$ seconds, where

$$T_k = t_{min} + S_k t_1 + \frac{\min(k, N_d)}{2} t_{wait} + \sum_{i=1}^{k-1} \frac{1}{2^i} t_{min} \tag{3}$$

seconds, where $S_k$ is the spot that the $k^{\text{th}}$ driver parks in according to the simulation, updated based on $N_a$. For each $k = 1, ..., N_a$, we consider $T_k$ as a single Monte Carlo sample, repeat the process 20 times, and average over all samples to determine the total average search time for each lot and each time of day. We use a value of $t_1 = 0.54$ seconds, which is based on observations made in a large local lot, averaging the time it taken to drive through the whole lot (while following the posted speed limit) over the number of stalls.

## 4 Data

### 4.1 Geographic Data

The city of Vancouver's open data catalogue consists of many publicly available datasets. We utilize information from two of these: *Parking Meter Data* contains the location, rates and time limits for approximately 10,000 parking meters in the city, and *City Streets* associates each meter in the Parking Meter dataset with the block face it is located on.

### 4.2 Occupancy Surveys

To train the occupancy prediction neural network, we use data from occupancy surveys done by city of Vancouver. We only use surveys of the blocks in Vancouver's downtown area. The survey data consists of timestamped, meter-level surveys. Since the times of the surveys of the meters in each block face are close to each other, we consider all surveys of the meters in a block face which fall in a period of 30 minutes as a single snapshot of the block face. The neural network features for each sample are derived from merging data from occupancy checks and online payment data. The occupancy check data is for 372 block faces in Vancouver's downtown core. It is important to note that we train our model using very sparse occupancy checks provided by the City for only a few days out of the year. The occupancy checks include about 8 checks for every block face consisting of 4 checks during the morning and 4 checks during the afternoon. In total, we have 3084 block-level occupancy check samples. What's more, about 17% of the meter-level occupancy checks are not timestamped, making those checks unusable, which means that on average every meter is missing in at least one of the occupancy checks.

### 4.3 Google Maps Data

The Google Maps Directions API can be used to find travel time between any two points in the city that are connected by roads. Ideally we would simply tell Google Maps the route to drive and report the time to drive it. However, the number of such calls that can be made to the API is limited. Instead, we use the API to find driving and walking times for each block face in the city by getting the travel time between the centers of the intersections at either end of the block. We can then reconstruct travel time between any two points in the city by using these values and executing a shortest path algorithm (e.g., Dijkstra's algorithm).

### 4.4 City-Owned Lot Data

We use data from 21 city-owned lots, containing between 9 and 643 stalls. For each lot we use hourly data from a 3 month period. In each hour the data records the number of entries and exits from the lot.

Unfortunately, this data is based on the amount of time each vehicle paid for, rather than the time actually spent in the lot. As a result, the exit data often exhibits huge artificial spikes at times that represent rate boundaries (e.g., a flat rate to park until 6:00pm results in a spike at 6:00pm). To deal with this, we first apply a Gaussian smoothing procedure to the departure data for each of these peaks. We redistribute the departures in the peaks back 12 hours. The quantities distributed to each hour are proportional to the probabilities of a left-half-Gaussian distribution with mean at the peak and standard deviation 3.5.

## 5 Results

### 5.1 Occupancy Prediction Network

To evaluate the performance of the occupancy prediction neural network, we compare it with a logistic regression model using cross entropy loss on the validation set. Cross entropy is a measure of how different two probability distributions are. Having a low cross entropy loss means a model is doing a good job of approximating the true distribution of the data. Under this measure, our network on average outperforms the baseline by 9%: the network achieves average validation cross entropy loss of 0.51 compared to the baseline with 0.56. It is also worth mentioning that despite still making improvements, our network does not significantly outperform the baseline in validation accuracy, achieving validation accuracy of 79% compared to the baseline of 77% for the regression model. However, in our case, cross entropy loss is a more meaningful measure of performance than validation accuracy, because we are trying to see if our model is doing a good job of predicting the probability that vehicle can park in a given block face.

### 5.2 Parking Inefficiencies

Using the search times generated by our two simulators we visualize the estimated time it would take a driver to park both on- and off-street, with each block in the city as the driver's assumed destination. Figures 2 and 3 show the total estimated time for parking on-street and off-street, respectively. Not surprisingly, the on-street map (Figure 2) shows it will take a relatively long time to park on-street in the west end of the city and around the Granville St. area, where meters are less common. Similarly, the off-street map shows that parking off-street is quickest when a driver's destination is near an off-street lot.

The interesting result comes from comparing the difference between these two maps. Figure 4 shows how much time a driver could actually *save* by parking off-street, with each block as the assumed destination. Blocks coloured green indicate areas that are at least as accessible from lots as they are from on-street meters. Most of these areas can be reached around 5 minutes *faster* by parking off-street, with some being as much as 10 to 15 minutes faster. Some of these areas are simply caused by a lack of parking meters on the destination street, which means that drivers must walk

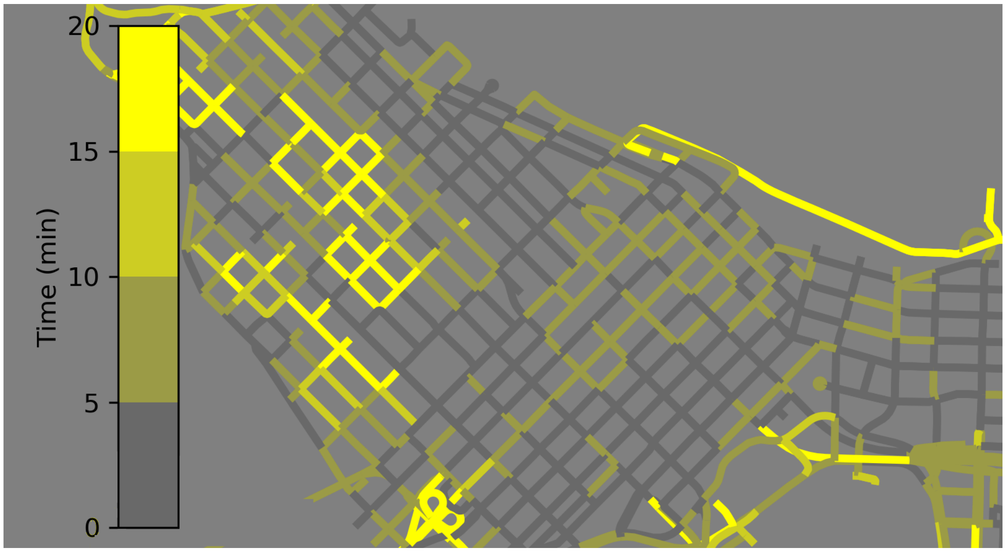


Figure 2: *On Street: Total estimated time to park on-street (including search, park, and walking times) with each block as the assumed destination of the driver. Yellow indicates it will take a relatively long time to park on-street if this block is the driver's final destination. Search times tend to be higher in areas with fewer meters. The map shown is for 12:00-1:00pm on a Friday.*

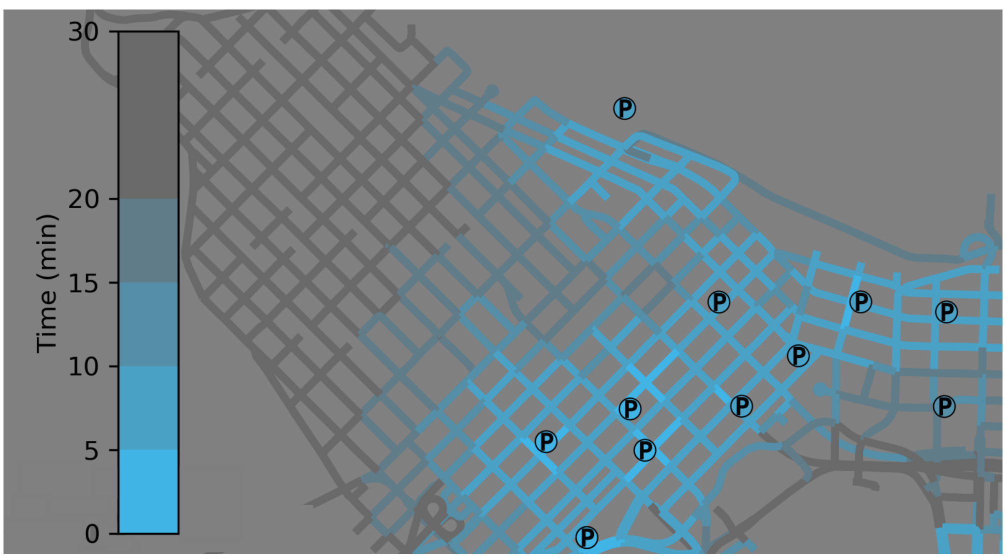


Figure 3: *Off Street: Total estimated time to park in an off-street lot (including driving and walking time, and searching in the lot) with each block as the assumed destination of the driver. Off-street lots are indicated with $P$ icons. Blue indicates a relatively short time to park in a nearby lot if this block is the driver's final destination. Search times tend to be lower in areas nearer to lots. The map shown is for 12:00-1:00pm on a Friday.*

some distance regardless of their choice of parking. However, the more interesting areas are those around Granville St., Yaletown, and Chinatown (seen in the bottom right corner of Figure 4). These are caused by the overlap of congested roads with areas that are easily accessible from off-street lots. In these areas searching for on-street parking can take a long time due to the congestion, but nearby lots mean off-street parking is relatively quick. These areas represent an opportunity for the city to effect change in the way people use parking resources.

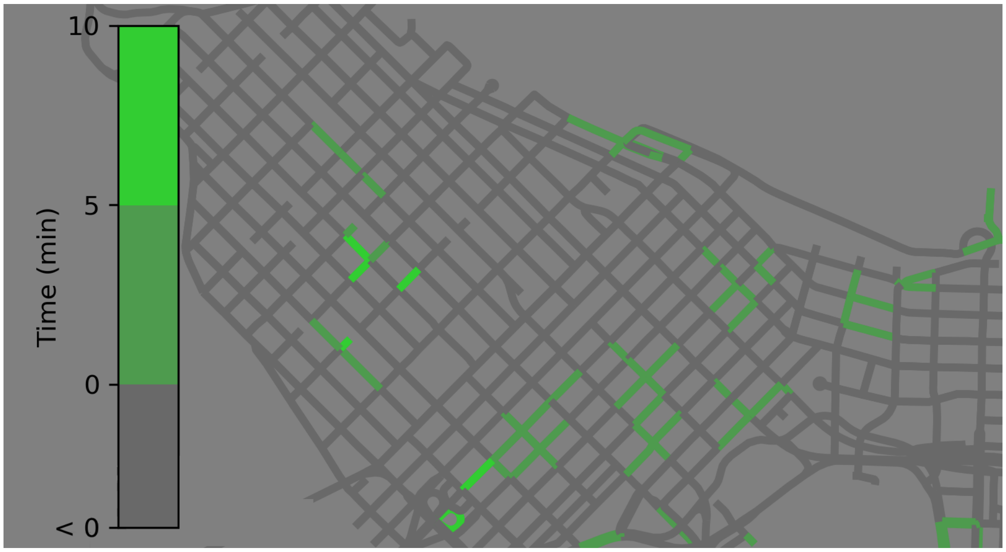


Figure 4: *Areas of Opportunity: Difference in estimated time between parking off-street and parking on-street. Blocks coloured green indicate drivers can actually save time by parking off-street rather than searching for curbside spots if these blocks are their destinations. Curbside parking in these areas can be eliminated, with nearby lots absorbing the excess capacity. This space can then serve more community-friendly purposes. The map shown is for 12:00-1:00pm on a Friday.*

## 6 Conclusion

While on-street parking of course has its purposes, it also comes with negative side effects that impact everyone in the city. Drivers who are looking for parking occupy valuable road space unnecessarily and also tend to be distracted. Parked cars can obscure pedestrians and cyclists from drivers' sight. At the same time, many off-street parking lots have large amounts of excess capacity at key times of day.

We have examined parking usage on metered blocks and in city-owned, off-street lots in downtown Vancouver. We used deep learning to predict true on-street occupancy rates from partial payment and other, block-level data. We used Monte Carlo simulation in conjunction with Google Maps driving and walking times to model the amount of time taken to park both on- and off-street, with each block face in the city as the assumed destination. We identified areas where it would be at least as fast for drivers to park in nearby lots as to search the area and then park on-street.

Since the particular lots we considered are city-owned, and so the City has the power to change the way they are operated, these areas represent a clear opportunity to improve the city as a whole. Curbside space is a valuable resource that modern municipalities can utilize to improve the lives of all their residents. In particular, on-street parking reduces safety for drivers, pedestrians and cyclists alike by obscuring vision and encouraging distracted drivers who are searching for spots. What's more, curbside lanes can be converted to dedicated bike or transit lanes, helping the City of Vancouver maintain its vehicle mode-share targets. Alternatively, these lanes can be converted to parklets or green spaces, making these areas more community-friendly.

How then can drivers be incentivized to use lots instead of curbside space? One simple way is through better information, either in the form of signage or through a website or app showing availabilities. Setting aside a portion of certain lots to have special pricing and time restrictions more in line with on-street rates could encourage more drivers to move off-street. Additionally, a reservation system could also help by ensuring drivers have a spot and know where they are going before leaving their houses. Any of these measures would help the City make better use of these valuable spaces.

## References

Kay W Axhausen, JW Polak, M Boltze, and J Puzicha. Effectiveness of the parking guidance information system in frankfurt am main. *Traffic Engineering+ Control*, 35(5):304–9, 1994.

Paul A Barter. Off-street parking policy without parking requirements: a need for market fostering and regulation. *Transport Reviews*, 30(5):571–588, 2010.

Sylvain Belloche. On-street parking search time modelling and validation with survey-based data. *Transportation Research Procedia*, 6:313–324, 2015.

Itzhak Benenson, Karel Martens, and Slava Birfir. Parkagent: An agent-based model of parking in the city. *Computers, Environment and Urban Systems*, 32(6):431–439, 2008.

Joschka Bischoff and Kai Nagel. Integrating explicit parking search into a transport simulation. *Procedia Computer Science*, 109:881–886, 2017.

Subhadip Biswas, Satish Chandra, and Indrajit Ghosh. Effects of on-street parking in urban context: a critical review. *Transportation in developing economies*, 3(1):10, 2017.

Geoff Boeing. Osmnx: New methods for acquiring, constructing, analyzing, and visualizing complex street networks. *Computers Environment and Urban Systems*, 65:126–139, 07 2017.

Chase P Dowling, Lillian J Ratliff, and Baosen Zhang. Modeling curbside parking as a network of finite capacity queues. *IEEE Transactions on Intelligent Transportation Systems*, 2019.

Jessica Edquist, Christina M. Rudin-Brown, and Michale G Lenné. The effects of on-street parking and road environment visual complexity on travel speed and reaction time. *Accident Analysis and Prevention*, 2011.

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

Konstantinos Kepaptsoglou, Matthew G. Karlaftis, and Eleni I. Vlahogianni. Exploiting new sensor technologies for real-time parking prediction in urban areas. 2014.

L Mannini, E Cipriani, U Crisalli, A Gemma, and G Vaccaro. On-street parking search time estimation using fcd data. *Transportation Research Procedia*, 27:929–936, 2017.

Wesley E Marshall, Norman W Garrick, and Gilbert Hansen. Reassessing on-street parking. *Transportation Research Record*, 2046(1):45–52, 2008.

Dadi Baldur Ottosson, Cynthia Chen, Tingting Wang, and Haiyun Lin. The sensitivity of on-street parking demand in response to price changes: A case study in seattle, wa. *Transport Policy*, 25(C):222–232, 2013.

Tooraj Rajabioun and Petros A Ioannou. On-street and off-street parking availability prediction using multivariate spatiotemporal models. *IEEE Transactions on Intelligent Transportation Systems*, 16(5):2913–2924, 2015.

Nagui M. Rouphail, H. Christopher Frey, James D. Coylar, and Alper Unal. Vehicle emissions and traffic measures: exploratory analysis of field observations at signalized arterials. *Annual Meeting of the Transportation Research Board*, 2001.

Donald C Shoup. Cruising for parking. *Transport Policy*, 13(6):479–486, 2006.

Arbi Tamrazian, Zhen Qian, and Ram Rajagopal. Where is my parking spot? online and offline prediction of time-varying parking occupancy. *Transportation Research Record*, 2489(1):77–85, 2015.

Hualiang Teng, Yi Qi, and Ping Yi. Parking search time and information identification for off-street spaces in new york city. *Transportation research record*, 1800(1):44–52, 2002.

Jan Vuurstaek, Luk Knapen, Bruno Kochan, Tom Bellemans, and Davy Janssens. First steps towards a state-of-the-art parking simulator. Elsevier, 2018.

Rashid A Waraich and Kay W Axhausen. Agent-based parking choice model. *Transportation Research Record*, 2319(1):39–46, 2012.

Xiang Yan, Jonathan Levine, and Robert Marans. The effectiveness of parking policies to reduce parking demand pressure and car use. *Transport Policy*, 73:41–50, 2019.

Shuguan Yang and Zhen Sean Qian. Turning meter transactions data into occupancy and payment behavioral information for on-street parking. *Transportation Research Part C: Emerging Technologies*, 78:165–182, 2017.

Shuguan Yang, Wei Ma, Xidong Pi, and Sean Qian. A deep learning approach to real-time parking occupancy prediction in spatio-termporal networks incorporating multiple spatio-temporal data sources. *arXiv preprint arXiv:1901.06758*, 2019.

Yanxu Zheng, Sutharshan Rajasegarar, and Christopher Leckie. Parking availability prediction for sensor-enabled car parks in smart cities. *2015 IEEE Tenth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*, pages 1–6, 2015.