Peeking Beneath the Hood of Uber

Le Chen
Northeastern University
Boston, MA
leonchen@ccs.neu.edu

Alan Mislove
Northeastern University
Boston, MA
amislove@ccs.neu.edu

Christo Wilson
Northeastern University
Boston, MA
cbw@ccs.neu.edu

ABSTRACT

Recently, Uber has emerged as a leader in the "sharing economy". Uber is a "ride sharing" service that matches willing drivers with customers looking for rides. However, unlike other open marketplaces (e.g., AirBnB), Uber is a black-box: they do not provide data about supply or demand, and prices are set dynamically by an opaque "surge pricing" algorithm. The lack of transparency has led to concerns about whether Uber artificially manipulate prices, and whether dynamic prices are fair to customers and drivers.

In order to understand the impact of surge pricing on passengers and drivers, we present the first in-depth investigation of Uber. We gathered four weeks of data from Uber by emulating 43 copies of the Uber smartphone app and distributing them throughout downtown San Francisco (SF) and midtown Manhattan. Using our dataset, we are able to characterize the dynamics of Uber in SF and Manhattan, as well as identify key implementation details of Uber's surge price algorithm. Our observations about Uber's surge price algorithm raise important questions about the fairness and transparency of this system.

Categories and Subject Descriptors

H.3.5 [**Online Information Services**]: Commercial services, Web-based services

Keywords

Uber; Surge Pricing; Sharing Economy; Algorithm Auditing

1. INTRODUCTION

Recently, there has been an explosion of services that facilitate the so-called "sharing economy". Websites like AirBnB, Care.com, TaskRabbit, and Fiverr allow individuals to advertise their services and set their own schedules without the necessity of working for a company. Typically, these websites function as marketplaces where participants set their own prices and choose who to accept jobs from. By acting as open platforms, these websites are poised to unlock the productive potential of millions of people.

Uber has emerged as a leader in the sharing economy. Uber is a "ride sharing" service that matches willing drivers (i.e., anyone with a car) with customers looking for rides. Uber charges passengers based on time and distance of travel; however, during times of high demand, Uber uses a "surge multiplier" to increase prices. Uber provides two justifications for surge pricing [11]: first, it reduces demand by pricing some customers out of the market, thus reducing the wait times for the remaining customers. Second, surge pricing increases profits for drivers, thus incentivizing more people to drive during times of high demand. Uber's business model has become so widely emulated that the media now refers to the "Uberification" of services [4].

While Uber has become extremely popular, there are also concerns about the fairness, efficacy, and disparate impact of surge pricing. The key difference between Uber and other sharing economy marketplaces is that Uber is a black-box: they do not provide data about supply and demand, and surge multipliers are set by an opaque algorithm. This lack of transparency has led to concerns that Uber may artificially manipulate surge prices to increase profits [25], as well as apprehension about the fairness of surge pricing [10]. These concerns were exacerbated when Uber was forced to publicly apologize and refund rides after prices surged during Hurricane Sandy [16] and the Sydney hostage crisis [19].

In order to understand the impact of surge pricing on passengers and drivers, we present the first in-depth investigation of Uber. We gather four weeks of data from Uber by emulating 43 copies of the Uber smartphone app and distributing them in a grid throughout downtown San Francisco (SF) and midtown Manhattan. By carefully calibrating the GPS coordinates reported by each emulated app, we are able to collect high-fidelity data about surge multipliers, estimated wait times (EWTs), car supply, and passenger demand for all types of Ubers (e.g., UberX, UberBLACK, *etc.*). We validate our methodology using ground-truth data on New York City (NYC) taxicabs.

Using our dataset, we are able to characterize the dynamics of Uber in SF and Manhattan. As expected, supply and demand show daily patterns that peak around rush hours. However, we also observe differences between these cities: although SF has many more Ubers than Manhattan, it also surges much more often. Surge prices are extremely noisy: the majority of surges last less than five minutes. Finally, by analyzing the movements and actions of Uber drivers, we

IMC'15, October 28–30, 2015, Tokyo, Japan.
Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM 978-1-4503-3848-6/15/10 ...$15.00.
DOI: http://dx.doi.org/10.1145/2815675.2815681 .

show that surge prices have a small, positive effect on vehicle supply, and a large, negative impact on passenger demand.

Additionally, our measured data enables us to identify key implementation details of Uber's surge price algorithm. Based on our understanding of Uber's algorithm, we make the following observations:

- Uber has manually divided cities into "surge areas" with independent surge prices. Prices update every 5 minutes and show high correlation with supply, demand, and estimated wait time over the previous 5-minute interval. This suggests that surge prices are set algorithmically, not manually, and that the algorithm is quite responsive.
- However, as April 2015, Uber began serving surge prices to users that do not always match the prices returned by the Uber API. Furthermore, surge prices were no longer uniform for users, even if they were in the same surge area at the same time. We reported this finding to Uber, and their engineers confirmed that this behavior was caused by a bug in their system.
- Although we show that surge prices cannot be forecast in advance, given knowledge of the surge pricing algorithm, we demonstrate how passengers can significantly reduce surge prices 10-20% of the time by exploiting differences between surge areas.

Our observations about Uber's surge price algorithm raise important questions about fairness and transparency. For example, users may receive dramatically different prices due to small changes in geolocation. Furthermore, the vague, changing aspects of the algorithm impacts drivers' ability to predict fares. Finally, the black-box nature of Uber's system makes it vulnerable to exploitation by passengers (as we show), or possibly by colluding groups of drivers [2].

Outline. We begin in §2 by introducing Uber. In §3, we present and validate our data collection methodology. In §4, we characterize supply and demand on Uber, and analyze Uber's surge pricing algorithm in §5. Based on these findings, we present a strategy for avoiding surge prices in §6, followed by related work and discussion in §7 and §8.

2. BACKGROUND

In this section, we briefly introduce Uber, surge pricing, and technical details about the Uber service.

Uber. Founded in 2009, Uber is a "ride sharing" service that connects people who need transportation with a crowd-sourced pool of drivers. Unlike typical transportation providers, Uber does not own any vehicles or directly hire drivers. Instead, Uber drivers are independent contractors (known as "partners") who use their own vehicles and set their own schedules. As of May 2015, Uber is available in 200 cities in 57 countries[1], and claims to have 160K active drivers in the U.S. alone [15].

Uber provides a platform that connects passengers to drivers in real-time. Drivers use the Uber *Partner app* on their smartphone to indicate their willingness to accept fares. Passengers use the Uber *Client app* to determine the availability of rides and get estimated prices. Uber's system routes passenger requests to the nearest driver, and automatically charges passengers' credit cards at the conclusion

[1]https://www.uber.com/cities

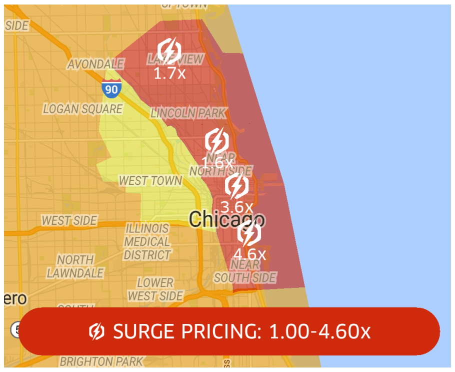


Figure 1: Screenshot of the Uber Partner app.

of each trip. Uber retains 20% of each fare and pays the rest to drivers.

Depending on location, Uber offers a variety of different services. UberX and UberXL are basic sedans and SUVs that compete with traditional taxis, while UberBLACK and UberSUV are luxury vehicles that compete with limousines. UberFAMILY is a subset of UberX and UberXL cars equipped with car seats, while UberWAV refers to wheelchair accessible vehicles. UberT allows users to request traditional taxis from within the Uber app. UberPOOL allows passengers to save money via carpooling, i.e., Uber will assign multiple passengers to each vehicle. UberRUSH is a delivery service where Uber drivers agree to courier packages on behalf of customers.

Surge Pricing. Uber's fare calculation changes depending on local transportation laws. It typically incorporates a minimum base fare, cost per mile, cost per minute, and fees, tolls, and taxes. The base fare and distance/time charges vary depending on the type of vehicle (e.g., UberX vs. UberBLACK).

In 2012, Uber introduced a dynamic component to pricing known as the *surge multiplier*. As the name suggests, fare prices are multiplied by the surge multiplier; typically the multiplier is 1, but during times of high demand it increases. Uber has stated that there are two goals of this system: *first*, higher profits may increase supply by incentivizing drivers to come online. *Second*, higher prices may reduce demand by discouraging price-elastic customers.

Little is known about Uber's surge price algorithm, or whether the system as a whole is successful at addressing supply/demand imbalances. As shown in Figure 1, surge multipliers vary based on location, and recent measurements suggest that Uber updates the multipliers every 3-5 minutes [6]. Uber has a patent pending on the surge price algorithm, which states that features such as car supply, passenger demand, weather, and road traffic may be used in the calculation [23].

Passenger's Perspective. Uber's apps provide different information to passengers and drivers. The Client app displays a map with the eight closest cars to the user (based on the smartphone's geolocation), and the Estimated Wait Time (EWT) for a car. The app provides separate eight-car inventories and EWTs for each type of Uber. Users are not shown the surge multiplier until they attempt to request a car (and only if it is >1). Although the app initially assumes that the user's current location is their pickup point, the user may move the map to choose a different pickup point. This new location may have a different EWT and/or surge multiplier than the original location.

Driver's Perspective. In contrast to the Client app, the Partner app displays very different information to drivers. As shown in Figure 1, the centerpiece of the Partner app is a map with colored polygons indicating areas of surge. Unlike the Client app, the locations of other cars are not shown. In theory, this map allows drivers to locate areas of high demand where they can earn more money. In practice, drivers often use the Partner and Client apps concurrently, to see the exact locations of competing drivers [3]. The Partner app's map also suggests that Uber calculates discreet surge multipliers for different geographic areas. We empirically derive these *surge areas* in § 5.

3. METHODOLOGY

There are three high-level goals of this study. *First*, we aim to understand the overall dynamics of the Uber service. *Second,* we want to determine how Uber calculates surge multipliers. *Third*, we want to understand whether surge pricing is effective at mitigating supply/demand imbalances. To answer these questions, we need detailed data about Uber (e.g., supply of cars, surge multipliers, *etc.*) across time and geography.

In this section, we discuss our approach for collecting data from Uber. We begin by motivating San Francisco (SF) and Manhattan as the regions for our study. Next, we discuss our methodology for collecting data from the Uber Client app, and how we calibrated our measurement apparatus. Finally, we validate our methodology using simulations driven by ground-truth data on all taxi rides in NYC in 2013.

3.1 Selecting Locations

In this study, we focus on the dynamics of Uber in SF and Manhattan. As we discuss in §3.2 and §3.3, there are practical issues that force us to constrain our data collection to relatively small geographic regions. Furthermore, not all regions are viable or interesting: Uber does not offer service everywhere, and many places will have few cars and passengers (i.e., rural areas).

We chose to focus on SF and Manhattan for four reasons. *First*, San Francisco and New York City have the 2nd and 3rd largest populations of Uber drivers in the U.S. (Los Angeles has the largest population) [15]. *Second*, SF was Uber's launch city, and recent measurements suggest that it accounted for 71% of all "taxi" rides in the city in 2014 (the highest percentage of any U.S. city; Uber accounted for 29% of all rides in NYC during 2014) [27]. *Third*, SF and NYC are very different cities in terms of culture and access to public transportation, which may lead to interesting differences in the dynamics of Uber.

Fourth and finally, Manhattan is a useful location to measure Uber because there also exists a publicly available dataset of all taxi rides in NYC for 2013 [22]. We leverage this data in §3.5 to validate the accuracy of our Uber measurement methodology.

3.2 The Uber API

Now that we have chosen locations, the next step is to collect data from Uber in these two areas. Like many modern web services, Uber provides an HTTP-based API for third-party developers to retrieve information about the state of the service. In our case, the `estimates/price` and `estimates/time` endpoints are most useful. The former takes longitude and latitude as input, and returns a JSON-encoded list of price estimates (including surge multipliers) for all car types available at the given location. The latter is similar, except it returns EWTs. Uber imposes a rate limit of 1,000 API requests per hour per user account.

While data from the Uber API is useful, it is not sufficient for this study, since it does not include information about car supply or passenger demand. Thus, we only rely on API data for specific experiments in §5.

3.3 Collecting Data from the Uber App

To overcome the shortcomings of the Uber API, we leverage the Uber Client app. After a user opens the app and authenticates with Uber, the app sends `pingClient` messages to Uber's server every 5 seconds. Each ping includes the user's geolocation, and the server responds with a JSON-encoded list of information about all available car types at the user's location. For each car type, the nearest eight cars, EWT, and surge multiplier are given. Each car is represented by a unique ID, its current geolocation, and a path vector that traces the recent movements of the car.

To gather this data, we wrote a script that emulates the exact behavior of the Client app. Our script logs-in to Uber, sends `pingClient` messages every 5 seconds, and records the responses. By controlling the latitude and longitude sent by the script, we can collect data from arbitrary locations. We created 43 Uber accounts (each account requires a credit card to create), giving us the ability to "blanket" a small geographic area with measurement points. To simplify our discussion, we refer to these 43 measurement points as "clients".

While we were collecting data we never encountered rate limits or had our accounts banned. This indicates that we very likely were not detected by Uber. Although it is possible that Uber detected our clients and fed them false data, it is much more plausible that Uber would have simply banned our clients if they were concerned about our measurements.

Measuring Demand and Supply. Using the data returned by `pingClient`, we can approximate the aggregate supply and demand within our measurement region. To measure *supply*, we can simply count the total number of unique cars observed across all measurement points; each of these cars represents a driver who is looking to provide a ride. To measure *demand*, we can measure the aggregate number of cars that go *offline* (disappear) between responses; one of the reasons a car may go offline is because it picked up a rider (we discuss other potential reasons, and how we handle them, below).

Limitations. Although `pingClient` returns more information than the Uber API, there are still four limitations that we must address. *First*, clients only receive information about the eight closest cars. Thus, to measure the overall supply of vehicles in a geographic area, we must position the 43 clients such that they completely cover the area. This situation is further complicated by the fact that each client's visibility changes as the density of Uber cars fluctuates (e.g., cars are dense during rush hour but sparse at 4am). In §3.4, we perform calibration experiments to determine the appropriate distance to space our clients.

Second, the demand we are able to estimate from our data is *fulfilled demand*, i.e., the number of cars that pick up passengers. Uber does not provide public data about *quantity demanded*, i.e., the number of passengers that request rides. The difference between fulfilled and quantity demand is that some passengers may request a ride but not receive one due to supply shortages. Thus, in this study, when we refer to "demand", we are talking about fulfilled demand.

Third, our measurement of demand may overestimate the true demand because there are three reasons why a car might disappear between one response and the next: 1) the car drives outside our measurement area, 2) the driver accepts a ride request, or 3) the driver goes offline. We can disambiguate case 1 since the Client data includes the path vector for each car. Although we cannot disambiguate cases 2 and 3, we can still use car disappearances as an upper-bound on the fulfilled demand within the measurement area.

Fourth, data from the Client app does not allow us to track individual Uber drivers over time. Although each car is assigned a unique ID, these IDs are randomized each time a car comes online. Unfortunately, there is no way to overcome this limitation, and thus none of our experiments rely on tracking individual drivers.

Phantom Cars. Several press articles claim that Uber's Client app does not display data about actual Uber cars; instead, they claim that the cars are "phantoms" designed to give customers the illusion of supply [26]. Uber has publicly disputed these claims [5,32], explaining that the data shown in the Client app is as accurate as possible, given practical constraints like the accuracy of smartphone GPS measurements. Furthermore, Uber stated that car locations may be slightly perturbed to protect drivers' safety. We have not observed any evidence in our data to suggest that the cars are phantoms; on the contrary, the cars in our data exhibit all the hallmarks of human activity, such as diurnal activity patterns (see §4). If Uber does present phantom cars, it is likely that they only do so in rural areas with low supply, rather than in major cities like Manhattan and SF.

Uber Driver App. As shown in Figure 1, the Driver app also includes useful information (i.e., the surge map). However, only registered Uber drivers may log in to the Driver app. We attempted to sign-up as an Uber driver, but unfortunately Uber requires that drivers sign a document prohibiting data-collection from the Driver app. We opted not to sign this agreement. Instead, in §5, we reconstruct the surge map based on data from the Uber API.

Ethics. While conducting this study, we were careful to collect data in an ethical manner. *First*, we do not collect any personal information about any Uber users or drivers. We discussed our study with the Chair of our University's Institutional Review Board (IRB); she evaluated it as not being subject to IRB review because we did not collect personal information or impact any user's environment.

Second, we minimize our impact on Uber users and drivers. Before we began our data collection, we conducted an experiment to see if our measurements would impact Uber users by artificially raising the surge price. Fully discussed below, our results strongly suggests that our measurements have no impact on the surge multiplier. Moreover, at no point in this study did we actually request rides from any Uber driver, and drivers are not able to observe our measurement clients in the Driver app.

Third, we minimized our impact on Uber itself by collecting just the data we need to perform the study. The overall effect of our measurements was the same as 43 extra users running the Uber Client app. Given that Uber claims millions of users worldwide, we believe this is a worthwhile tradeoff in order to conduct this research.

Other Ride-Sharing Services. Although we attempted to collect data from other ride sharing services, these efforts were not successful. Lyft implements "prime time" pricing, but this data is only available *after* a user requests a ride. Thus, there was no ethical way for us to collect this data. Sidecar does not implement surge pricing; instead, drivers set their own rates based on time and distance. These additional variables make it difficult to systematically collect price information.

3.4 Calibration

The next step in our methodology is determining the locations for our 43 clients in SF and Manhattan. This step is crucial; on one hand, if we distribute the clients sparsely, we may only observe a subset of cars and thus underestimate supply and demand (recall that `pingClient` responses from Uber only contain the closest eight cars to each client). On the other hand, if the clients are too close together, the cars they observe will overlap, and we will fail to observe supply and demand over a sufficiently large geographic area.

To determine the appropriate placement of our 43 clients, we conducted a series of experiments between December 2013 and February 2014. In our first experiment, we chose a random location in Manhattan and placed all 43 clients there for one hour. We then repeated this test over several days with different random locations around Manhattan and SF. The results of these experiments reveal two important details about Uber: *first*, during each test, all 43 clients observed exactly the same vehicles, surge multipliers, and EWTs. This strongly suggests that the data received from `pingClient` is deterministic.

Second, when the clients were placed in areas where we would not expect to see surge (e.g., residential neighborhoods at 4 a.m.), all 43 clients recorded surge multipliers of 1 for the entire hour. This strongly suggests that our measurement methodology does not *induce* surges. As we show in Figure 8, fulfilled demand in midtown Manhattan peaks around 100 rides per hour, so 43 clients is a significant enough number that we would expect surge to increase if the algorithm took "views" into account.

The goal of our next experiment is to measure the *visibility radius* of clients. Intuitively, this is the distance from a client to the furthest of the eight cars returned by `pingClient`.

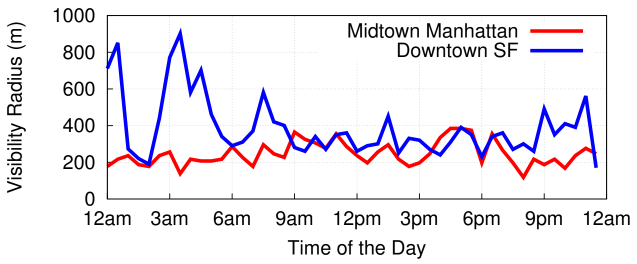
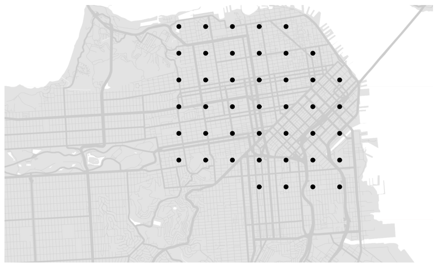
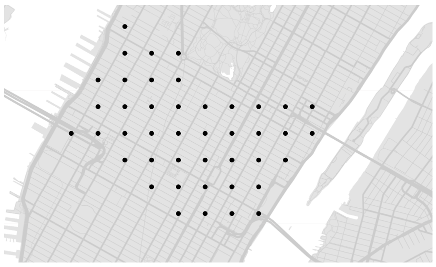


Figure 2: Visibility radius of clients in two cities.


(a) Uber in downtown SF.


(b) Uber in midtown Manhattan.

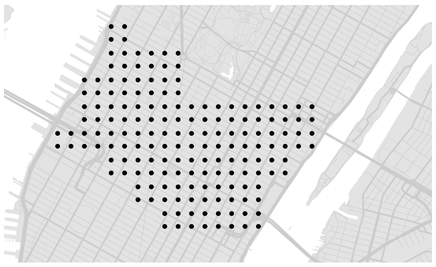

(c) Taxis in midtown Manhattan.

Figure 3: Locations of our Uber and taxi measurement points in SF and Manhattan.

Once we know the visibility radius in SF and Manhattan, we can determine the placement of our 43 clients.

To calculate the visibility radius, we conduct the following experiment. We 1) place 4 clients, denoted as $C = \{c_1, c_2, c_3, c_4\}$, at the same geolocation; 2) each of the clients "walks" 20 meters Northeast, Northwest, Southeast and Southwest (respectively) every 5 seconds; 3) the experiment halts when $|\bigcap_c V_c| = 0$, where V_{c_i} is the set of cars observed by client c_i; 4) record the distance D_c from each $c \in C$ to the starting point. Given this information, we calculate the visibility radius r (consider a 45°-45°-90° triangle where r is the leg and D_c is the hypotenuse) as:

$$r = \frac{1}{4}\sum_c \frac{D_c}{\sqrt{2}} \approx 0.1768 \times \sum_c D_c$$

Figure 2 shows the measured radii in meters when the clients were placed in downtown SF and midtown Manhattan. We chose these specific locations because they are the "hearts" of these cities, and thus they are likely to have the highest densities of Uber cars. As expected, the visibility radius changes throughout the day, with the most obvious difference being night/day in SF. There are also differences between the cities: the average radius is 247 ± 2.6 meters in Manhattan, versus 387 ± 6.8 meters in SF.[2]

In the end, we chose 200 meters as the radius for our data collection in midtown Manhattan, and 350 meters in downtown SF. These values represent a conscientious tradeoff between obtaining complete coverage of supply/demand and covering a large overall geographic area. Figures 3a and 3b depict the exact positions where we placed our clients in SF and Manhattan.[3]

3.5 Validation

Our final step is to validate our measurement methodology. The fundamental challenge is that we do not have ground-truth information about supply and demand on Uber; we attempt to mitigate this through careful placement of our clients, but the key challenge is having confidence that we will observe the vast majority of cars.

To address this issue, we constructed an Uber simulator powered by ground-truth data on NYC taxis [22]. The NYC taxi data includes timestamped, geolocated pickup and dropoff points for all taxi rides in NYC in 2013. Each taxi is assigned a unique ID, so its location can be tracked over time. Our simulator takes the taxi data as input, and plays the rides back in real-time. Since the taxi data only includes pickup and dropoff points, the simulator "drives" each taxi in a straight-line from point-to-point. We assume that a taxi has gone "offline" if it is idle for more than 3 hours (this filter only removes 5% of taxi sessions in the data).

We built an API in our simulator that offers the same functionality as Uber's `pingClient`: it returns the eight closest taxis to a given geolocation. Just as with Uber, the ID for each taxi is randomized each time it becomes available. Given this API, we used our methodology from §3.3 and §3.4 to measure the supply and demand of taxis over time. If the measured values from the simulator's API are similar to the ground-truth values, we can confidently say that our methodology will also collect accurate data from Uber.

Calibration. To make our simulation fair, we calibrated it by using four taxi clients to determine the visibility radius for taxis in midtown NYC (see §3.4). Taxis are much denser than Ubers in this area, so $r = 100$ meters is commensurately smaller. Figure 3c shows the locations of our 172 taxi clients; compared to Uber clients, it takes 300% more taxi clients to cover midtown.

Results. Using our taxi clients, we measured the supply and demand of taxis in the simulator between April 4–11, 2013. We chose these dates because they correspond to the same month and week of our Uber measurements (except in 2013 versus 2015, see §4).

As we discuss in §3.3, neither Uber nor our simulator return direct information about demand. Instead, we assume that cars that 1) disappear from the measured data, and 2) were not driving near the outer edge of the measurement polygon were booked by a passenger.[4] We refer to these events as *deaths*.

Figure 4 plots the measured and ground-truth taxi supply and demand per 5-minute interval. The two lines are almost indistinguishable since our taxi clients capture 97% of cars and 95% of deaths. The results provide strong evidence that our measurement methodology captures most of Uber's supply and demand.

4. ANALYSIS

In this section, we analyze supply, demand, and wait times on Uber. We begin by briefly introducing our datasets and how we cleaned them. Next, we examine how the dynamics of Uber change over time and spatially across cities. We

[2]Throughout this study, we present the 95% confidence interval (CI) of the mean value.

[3]All map images used in this paper are ©2015 Google.

[4]Restriction (2) is conservative: cars near the edge of the measurement area may disappear because they were booked, or because they drove outside the measurement polygon.

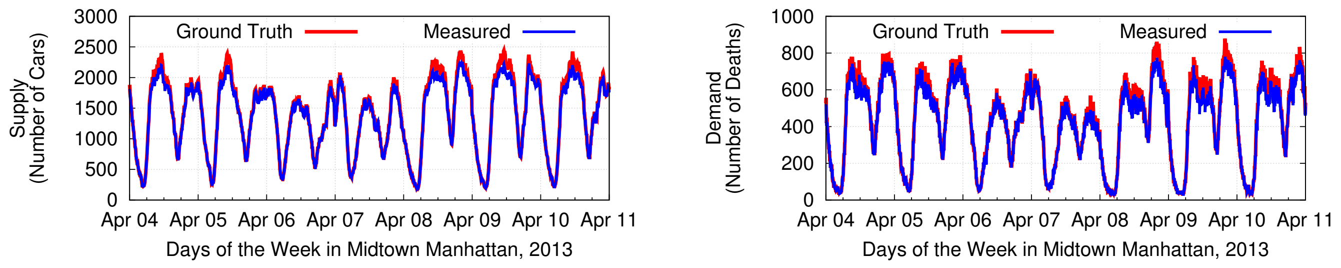


Figure 4: Measured and ground-truth supply (left) and demand (right) of taxis in midtown Manhattan.

focus the majority of our analysis on UberX, since (as we will show) they are the most common type of Uber by a large margin. We defer detailed analysis of surge multipliers and the surge pricing algorithm to §5.

4.1 Data Collection and Cleaning

For this study, we collected data from 43 clients placed in midtown Manhattan and downtown SF between April 3rd–17th (391 GB of data containing 9.3M samples) and April 18th–May 2nd (605 GB of data, 9.4M samples), respectively. Data was collected using the Uber client methodology described in §3.3. The locations of our clients are shown in Figures 3a and 3b. We also collected more limited datasets from the Uber API, as well as client data between December 27th, 2014–March 1st, 2015; we use this data in §5.

Before proceeding with analysis, our data must be cleaned to remove outliers. Figure 5 shows the *lifespan* of UberX cars in our dataset, measured as the time delta during which we observe a car's ID. Problematically, ∼50% of UberXs have a lifespan of zero, i.e., we only observe them in a single `pingClient` response. We refer to these cars as being *short-lived*, and observe similar trends for other types of Ubers.

The obvious question is: what is the cause of short-lived cars? Fortunately, we discovered that short-lived cars are an easily understood artifact of our measurement methodology. We examined the GPS coordinates of all short-lived cars, and found that 80% of them are *outsiders*: they are $> r$ meters away from the nearest client in our measurement polygon. Figure 6 plots the distances of short-lived *insiders* (those within the measurement polygon), and shows that 100% of them are $< r$ meters from the polygon boundary.

These results reveal that short-lived cars are due to the fact that `pingClient` only returns information about the eight nearest cars. Cars that are outside the measurement polygon, or $< r$ meters from its edge, may only be observed by a single client. These cars may get pushed out of `pingClient` responses by closer vehicles, or the car may only briefly be near our measurement area. In contrast, cars well-within the boundaries can potentially be observed by many clients, even as they drive around. Thus, we can safely filter short-lived cars from our dataset, and focus in the remainder of the paper only on cars that are driving within the bounds of our measurement area.

Car Lifespan. Figure 7 shows the lifespans of Ubers after removing short-lived cars. We observe similar trends in Manhattan and SF: ∼90% of low-priced Ubers (X, XL, FAMILY, and POOL) live for <10 minutes, while ∼65% of high-priced Ubers (BLACK and SUV) live for <10 minutes. There are two possible reasons for this observation: first, demand for low-priced Ubers may be higher than for expensive Ubers. Second, as we show in §4.2, low-priced Ubers greatly outnumber high-priced Ubers, so there may simply be more churn amongst the former due to their greater prevalence.

4.2 Dynamics Over Time

Next, we examine how supply, demand, surge multiplier, and EWT vary over time. We calculate supply by counting all the unique car IDs observed by our 43 clients during each 5-minute interval. Demand is the number of cars that disappear during each 5-minute interval (excluding cars that drive outside the measurement area). Surge multiplier and EWT are averaged across each 5-minute interval and all 43 clients. We explain why we chose 5-minute intervals in §5.

Note that our measurements for supply and demand are approximations. As shown in §3.5, our methodology may miss some cars, leading us to slightly underestimate supply. Similarly, we cannot disambiguate cars that pickup a passenger from cars that simply go offline. Thus, our demand numbers are an upper-bound. Finally, a driver that repeatedly goes online and offline during a 5-minute interval will appear as multiple unique cars to us, since cars are assigned a fresh ID each time they come online.

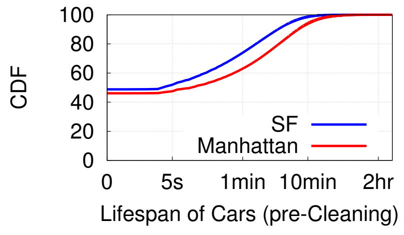


Figure 5: Lifespan of UberX cars in Manhattan and SF before data cleaning.

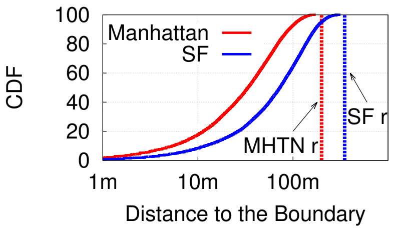


Figure 6: Distance of short-lived insiders to the boundary.

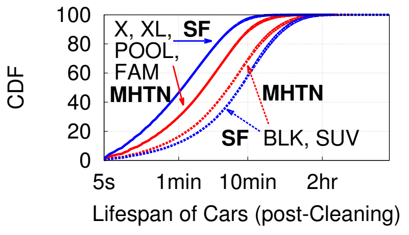


Figure 7: Lifespan of different types of Uber cars after data cleaning.

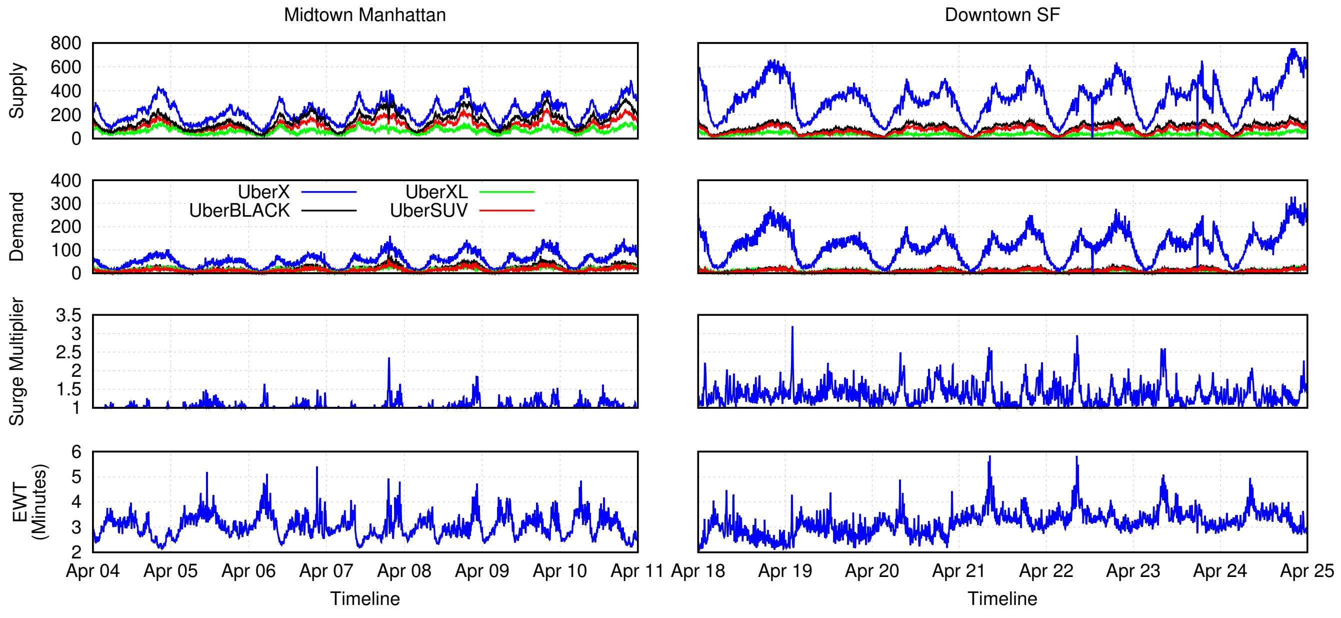
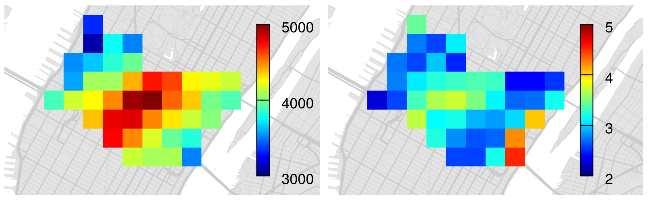
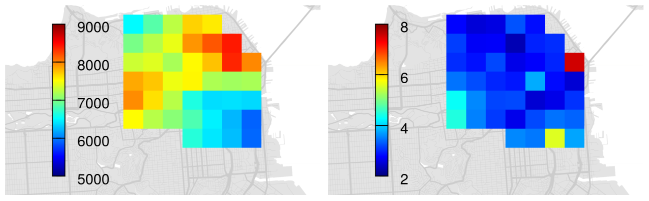


Figure 8: Supply, demand, surge multiplier, and EWT over time for midtown Manhattan and downtown SF. Surge multiplier and estimated wait time (EWT) are only shown for UberX. Diurnal patterns are observed in supply and demand, but the characteristics of the surge multiplier show less predictability.



(a) Avg. Number of Cars (b) Avg. EWT (Minutes)

Figure 9: Heatmaps for UberX cars in Manhattan.



(a) Avg. Number of Cars (b) Avg. EWT (Minutes)

Figure 10: Heatmaps for UberX cars in SF.

Unsurprisingly, Figure 8 shows that Uber exhibits regular daily trends: all four quantities peak during the day and decline at night. We also observe that supply, demand, and EWT have local peaks during morning and afternoon rush hour.[5] Rush hour peaks are less prominent for surge pricing, and we show in §5 that surge pricing is extremely noisy.

Both cities exhibit the same rank ordering of Uber types. UberXs are most prevalent, followed by UberBLACK, UberSUV, and UberXL. Both cities also have other types of Ubers (e.g., UberRUSH, UberFAMILY, *etc.*), however there are only 4 cars of these types on the road on average. Manhattan does have a significant number of UberT's, but these are not interesting in our context since they are not subject to surge pricing (recall that UberT corresponds to an ordinary taxi). Comparing the NYC taxi data in Figure 4 to Figure 8 we see that there are an order of magnitude more taxis in midtown Manhattan than all Ubers combined.

Despite their similarities, Figure 8 also reveals differences between Manhattan and SF. In our measurement region, SF has 58% more Ubers overall than Manhattan, mostly due to the large amount of UberXs in SF. The relative dearth of Ubers in Manhattan may be due to greater availability of taxis and better public transport. However, Manhattan has more UberXLs, UberBLACKs, and UberSUVs than SF.

Manhattan and SF also exhibit different surge pricing characteristics. As shown in Figure 8, downtown SF *surges* (i.e., surge >1) more frequently than midtown Manhattan. Surge multipliers are also tend to be higher in SF: $1.36 \pm 1 \times 10^{-4}$ on average versus $1.07 \pm 7 \times 10^{-5}$ in Manhattan. In midtown Manhattan, surge tends to increase starting at 3pm through evening rush hour Monday–Thursday. On the weekends, surge tends to peak between noon and 3pm, likely due to the influx of tourists. In SF, surge peaks at around 2.0 during morning rush hour (6am–9am) Monday–Friday. Surge also has a localized peak at 2am every night (but especially on weekends, reaching up to 3.0), which is "last call" throughout California.

Although Figure 8 focuses on surge multipliers of UberX, other types of Ubers exhibit similar trends. A recent report estimated that Uber accounts for 71% of "taxi" rides in SF versus 29% in NYC [27], so it is possible that this difference in demand explains the differences in surge characteristics. We defer in-depth discussion of surge pricing to §5.

We also observe that Uber offers expedient service in both cities. Average EWT for an UberX in midtown Manhattan

[5]Note that the y-axes of supply and demand have different scales.

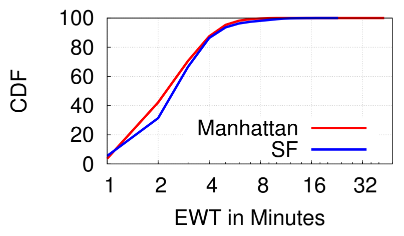


Figure 11: Distribution of EWTs for UberXs.

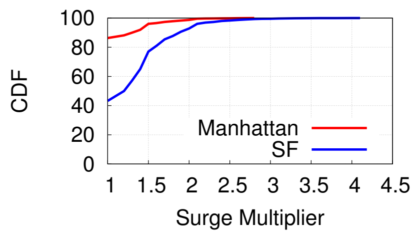


Figure 12: Distribution of surge multipliers for UberXs.

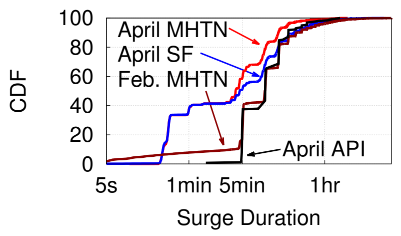


Figure 13: Duration of surges for UberXs.

is 3.0 $\pm$ 2 $\times$ 10^{-4} minutes, and 3.1 $\pm$ 2 $\times$ 10^{-4} minutes in downtown SF. In both cities, average EWT never exceeds six minutes. We observe similar trends for other types of Ubers, although rarer types (e.g., UberFAMILY) typically have slightly longer EWTs.

4.3 Spatial Dynamics and EWT

Next, we investigate the spatial dynamics of Uber. Figures 9(a) and 10(a) show the average number of unique UberX IDs seen per day by each of our 43 clients.[6] Since each square in these figures is an average, each one has a different confidence interval; the min and max CI for cars in Manhattan are $\pm$ 103 and $\pm$ 205 cars, respectively. The min and max CI for cars SF is are $\pm$ 170 and $\pm$ 250. As one might expect, the distribution of cars in both cities is skewed towards commercial and tourist locations. In Manhattan, UberXs congregate between Times Square and 5th Avenue. In SF, UberXs are densest in Russian Hill, Telegraph Hill, the Embarcadero, and the Financial District (upper-right corner of Figure 10(a)), as well as around the University of California, San Francisco (UCSF, lower-left corner).

In contrast, Figures 9(b) and 10(b) depict the distribution of EWTs across space. Each square is the average EWT for UberXs measured every five seconds over two weeks. The min and max CI for Manhattan are $\pm\, 6\times10^{-6}$ and $\pm\, 2\times10^{-4}$ minutes, respectively; the min and max CI for SF are $\pm$ 7×10^{-7} and $\pm$ 4×10^{-5}. We observe that there is a complex relationship between car density and EWT. In some cases, locations with low car density have commensurately higher EWTs (e.g., the lower-right corners in Figures 9(b) and 10(b)), suggesting that these areas are under-supplied. However, we also observe under-supply in several areas with high car density (e.g., Times Square and UCSF). This complex interplay between supply and demand supports Uber's case for implementing dynamic pricing.

Figure 11 presents the overall distribution of EWTs in Manhattan and SF for UberXs. 87% of the time, the wait for an UberX is $\leq$4 minutes. However, there are rare instances of severe supply/demand imbalance when EWT can go as high as 43 minutes. Note that our EWT data comes from the hearts of Manhattan and SF, and may not be representative for suburban areas.

5. SURGE PRICING

Now that we have an understanding of supply and demand on Uber, we turn our attention to surge pricing. Surge pricing is one of Uber's most controversial features [19, 24], and thus it warrants special attention. We begin by analyzing the basic characteristics of surge pricing. Next, we use our measured data to extrapolate how Uber updates surge prices over time and geography, and explore the features that Uber uses when calculating surge. Finally, we examine the impact of surge pricing on car supply and passenger demand.

[6]Note that these numbers are strict upper-bounds on the true number of UberX cars, since IDs are randomized each time a car comes online.

5.1 The Cost of Surges

We begin by answering the questions: *how often and how much does it surge on Uber?* Figure 12 presents the distribution of surge multipliers for UberXs over two weeks. We observe that Manhattan and SF have drastically different characteristics: 86% of the time there is no surge in Manhattan, versus 43% of the time in SF. Furthermore, the maximum surge multiplier we observe in Manhattan is 2.8, versus 4.1 in SF. In both cities, the surge multiplier is $\leq$1.5 during the majority of surges.

The results in Figure 12 reveal that prices surge on Uber a large fraction of the time (in SF, it's surging the *majority* of the time). Although most of the time the multiplier makes UberXs 25-50% more expensive, there are times (especially in SF) when the multiplier can double, triple, or even quadruple prices.

5.2 Surge Duration and Updates

Next, we answer the question: *how long do surges last?* Figure 13 plots the duration of surges for UberXs over two weeks. We define the duration of a surge as the continuous length of time when the multiplier is >1.

When we first began collecting data from Uber in February 2015, we observed that 90% of surges had durations that were a multiple of 5 minutes. This is shown by the stair-step pattern in the "Feb. Manhattan" line in Figure 13. We also measured surge durations using the Uber API in April 2015,

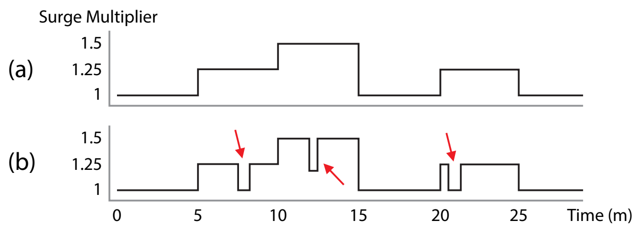


Figure 14: Examples of surge over time as seen from (a) the API and (b) the Client app. Although surge is recalculated every 5 minutes in both cases, clients also observe surge jitters for 20-30 seconds (red arrows).

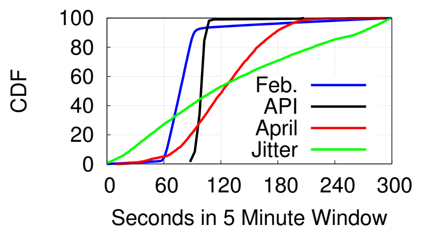


Figure 15: Moment when surge changes during each interval.

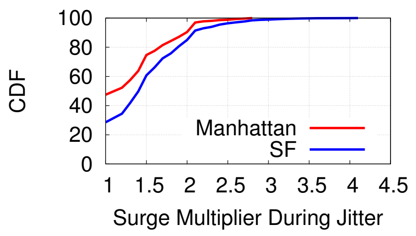


Figure 16: Distribution of surge multipliers seen during times of jitter.

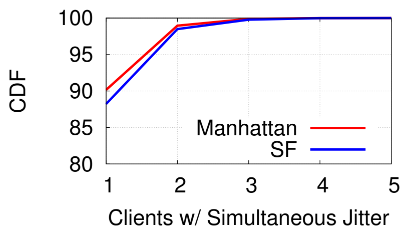


Figure 17: Fraction of clients that observe jitter at the same time.

and observed the same 5-minute pattern. In both of these cases, ∼40% of surges last 5 minutes, and ∼20% last 10 minutes. Less than 10% of surges last more than 20 minutes.

Strangely, the behavior of the surge price algorithm changed in April 2015. As shown by the two "April" lines in Figure 13, 40% of surges are now less than 1 minute long. The remaining 60% of surges still loosely follow the 5-minute stair-step pattern.

There are two takeaways from Figure 13. First, under normal circumstances, Uber appears to update surge prices on a 5-minute clock (we discuss jitter in greater detail below). Second, the vast majority of surges are short-lived, which suggests that savvy Uber passengers should "wait-out" surges rather than pay higher prices.

Figure 14 illustrates how Uber updates surge multipliers. Figure 14(a) shows the datastream from the API (and what clients observed, prior to April 2015): surge changes at regular 5-minute intervals. Figure 14(b) shows the behavior of `pingClient` responses as of April 2015: surge multipliers are still updated on a 5-minute clock, but within each interval there may be brief periods of jitter. Jitter breaks up a 5-minute surge interval into three separate components, which explains why we observe many surge durations less than 1 minute long in Figure 13. These two processes are the same in Manhattan and SF, and for all types of Ubers.

Timing. Figure 15 examines the fine-grained timing of surge updates. We chose one day of data in February and April, divided time into 5-minute intervals starting at 12:00am, and calculated the exact moment in each interval when surge changed due to the clock and jitter. We observe that the old client-update and the current API-update mechanisms are extremely regular: updates occur during a 35-second range in each interval. The new client-update mechanism is less precise: updates occur during a 2-minute range in each interval. Jitters are distributed almost uniformly throughout the interval, which suggests that jitter is driven by a stochastic or non-deterministic process.

Jitter. Next, we examine the behavior of jitter. In our data, 90% of jitter events last 20-30 seconds, and 100% are less than 1 minute. Furthermore, during jitter, we observe that the surge multiplier is equal to the multiplier from the *previous* 5-minute interval. Because most surges only last 5 minutes, this means that jitter causes the surge multiplier to drop 74% of the time in Manhattan, and 64% of the time in SF. As shown in Figure 16, in 30-50% of jitter events the surge multiplier drops to 1, depending on location. Thus, jitter almost always reduces prices for passengers, if they are lucky enough to request a car during the brief window when the low multiplier is available.

Jitter also deviates from the 5-minute surge intervals in another key way. As we demonstrate in the next section, the 5-minute surge intervals are uniform over specific geographic regions. However, jitter occurs on a per-client basis. Figure 17 plots the number of our clients that observed jitter at the same moment in time (recall, we have 43 total clients). We see that ∼90% of jitter events are only observed by a single client, and none are observed by more than 5 clients simultaneously.

In August 2015, we contacted Uber to make them aware of our findings. They were very concerned about the presence of jitter in the datastream, and the implication that customers were receiving inconsistent surge multipliers. We provided log data to Uber's engineers, and they quickly determined that jitter was being caused by a consistency bug in their system. The manifestation of this bug was that random customers could receive stale surge multipliers. This precisely coincides with our observations that jitter appeared at random times for random clients, and that the surge multiplier during jitter was equal to multiplier from the previous 5-minute window. As of this writing, Uber is fixing the bug and restoring consistency for all customers.

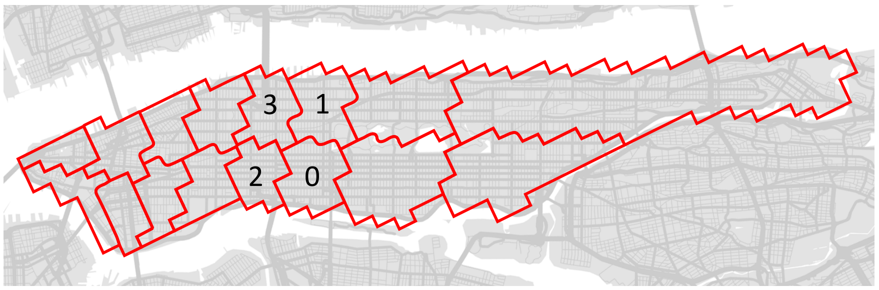


Figure 18: Surge areas in in Manhattan.

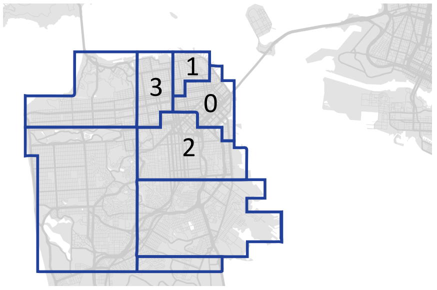


Figure 19: Surge areas in SF.

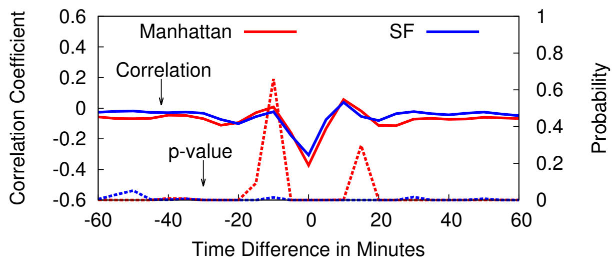


Figure 20: (Supply - Demand) vs. Surge for UberX.

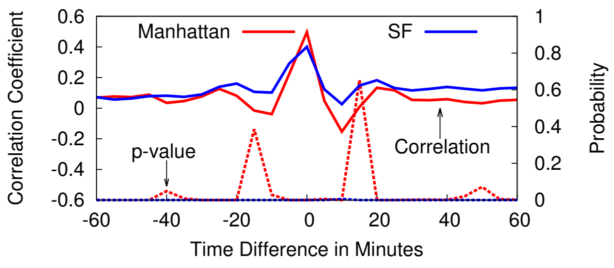


Figure 21: EWT vs. Surge for UberX.

5.3 Surge Areas

Next, we answer the question: *how do surge prices vary by location?* Intuitively, it is clear that surge must vary geographically: a rural location is very different than Times Square.

To answer this question, we used the Uber API to query surge prices throughout Manhattan and SF over the course of eight days. During these tests, we queried data from adjacent locations that obey our visibility radius constraints (see §3.4). Fortunately, since we know that surge prices only change every 5 minutes (and the API does not contain jitter), this enabled us to scale up our measurements to large geographic areas.

Using this data, we looked for clusters of adjacent locations that *always* had equal surge multipliers. Figures 18 and 19 show the regions of Manhattan and SF where the surge multipliers are always in lock-step. These figures reveal the granularity of Uber's surge price algorithm: Uber partitions cities into *surge areas* and computes multipliers independently for each area. The numbered areas correspond to the locations where we collected client data (see Figures 3a and 3b). Given the odd shape of some regions, it is likely that Uber defines surge areas manually. Note that there are some areas where we did not observe sufficient surging samples during our measurements (e.g., Upper Manhattan and Washington Heights); in these cases we cannot isolate individual surge areas, so it is possible these large areas may be composed of several smaller areas.

5.4 Algorithm Features and Forecasting

Next, we investigate the question: *what features does Uber use to calculate surge prices?* Uber has a pending patent that lists potential features [23], but it is unclear what features are truly used in the calculation.

To answer this question, we examine the cross correlation between observed supply, demand, and EWT versus surge price. In these tests, we treat each feature as a continuous time series. For surge, the time series is simply the observed surge multipliers during each 5-minute interval (we discard jitters since they are unpredictable). For supply, demand, and EWT we construct corresponding time series by averaging each quantity over the 5-minute window. We construct independent time series for the four areas in Manhattan and SF where we collect client data, since each area has its own surge characteristics. As before, we focus on UberXs.

Although we examined many possible correlations between these features (such as supply and surge, demand and surge, supply-demand ratio and surge, *etc.*), we obtained the strongest results when comparing *(average supply - average demand)* and *average EWT* to surge. Figure 20 shows the cross correlation (and p-value) between supply/demand difference and surge price. The correlation coefficient at time shift Δt is computed using surge at time t and feature values in the interval $[t + \Delta t - 5, t + \Delta t)$. We observe a relatively strong negative correlation when $-10 \leq \Delta t \leq 10$, which indicates that the surge multiplier rises when supply/demand difference shrinks. It appears that Uber's goal is to maintain a certain amount of slack in their car supply: when demand approaches available supply, surge pricing is instituted to increase supply and reduce demand. Figure 20 also reveals that Uber's surge pricing algorithm is quite responsive, since the correlation is strongest when $\Delta t = 0$.

Figure 21 shows the cross correlation between average EWT and surge price. In this case, we see a relatively strong positive correlation at $\Delta t = 0$, i.e., EWT and surge price increase at the same time. This result also makes sense: if surge increases during times of strained supply, then the wait times for cars should also increase.

Forecasting. Given that we observe correlations between supply, demand, EWT, and surge prices, this raises a new question: *can future surge prices be forecast?* To answer this question, we fitted three linear regression models that take the current supply/demand difference, EWT, and surge multiplier for UberXs as input, and predict the surge multiplier in the next 5-minute interval. As before, we filter jitter out of the time series. We also evaluated models that accept historical data (e.g., samples from previous 5-minute intervals), but this resulted in worse predictive performance.

Table 1 presents the overall R^2 scores for three linear regression models evaluated on two weeks of UberX data. We fitted separate models for all four surge areas in Manhattan and SF (see Figures 18–19), but in the interest of space we present the average R^2 scores in Table 1. In all three models,

City	Raw				Threshold				Rush			
	θ_{sd_diff}	θ_{ewt}	θ_{prev_surge}	R^2	θ_{sd_diff}	θ_{ewt}	θ_{prev_surge}	R^2	θ_{sd_diff}	θ_{ewt}	θ_{prev_surge}	R^2
NY	0.02	0.03	0.13	0.37	0.01	0.12	0.25	0.43	-0.01	0.26	0.33	0.43
SF	0.01	0.34	0.45	0.40	0.01	0.23	0.43	0.43	0.01	0.20	0.43	0.57

Table 1: Estimated Parameters and R^2 scores of linear regression in Manhattan and SF

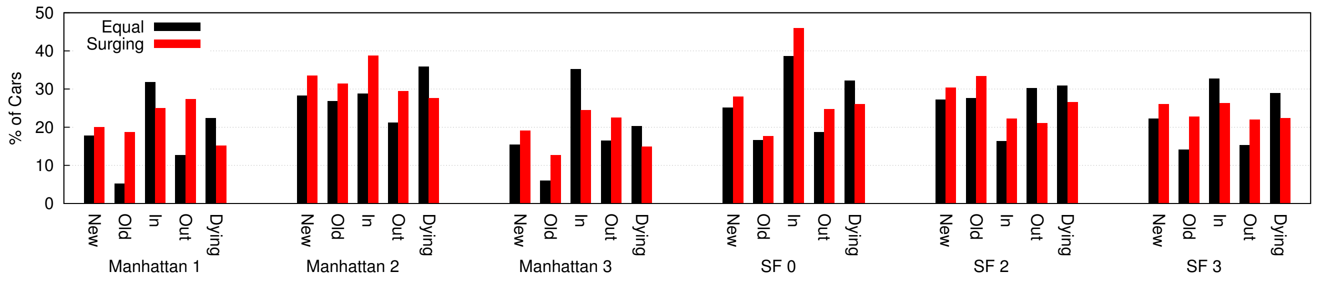


Figure 22: Transition probabilities of UberXs when all areas have equal surge, and when one area is surging.

we remove time intervals when the surge multiplier equals 1 from the data[7] before fitting, since this would make the prediction task too easy (e.g., you could achieve 86% accuracy in Manhattan by always predicting that the surge multiplier will be 1, see §5.1).

We also provide the parameters learned in the linear regression models in Table 1. Note that the learned parameters do not represent the relative importances of the variables; they are simply the slopes of the fitting surfaces.

Unfortunately, even with our large corpus of data, forecasting surge multipliers is an extremely difficult task. The *Raw* model in Table 1 is the most permissive of our three models: it was fitted and evaluated on the entire time series. We see that it has the worst performance, which suggests that either we are missing some key data that is used in Uber's surge calculation, or surge pricing is simply very noisy. The *Threshold* model improves performance by being more restrictive: it only attempts to predict the surge at time t if surge was >1 at $t-1$. We instituted this filter because surge cannot go below 1, i.e., we know less about the state of the system when surge is 1 than when surge is >1.

Finally, the *Rush* model is the most restrictive: it was fitted and evaluated on the subset of data corresponding to rush hours (6am–10am and 4pm–8pm). Although the performance of the Rush model clearly benefits from the predictable characteristics of rush hour traffic (see §4.2), it does not perform uniformly better than the more general Threshold model.

In summary, our forecasting results demonstrate that it is very difficult to predict surge multipliers, even with large amounts of supply and demand data. None of our models exhibits strong predictive performance in absolute terms (i.e., $R^2 \geq 0.9$). This suggests that Uber relies on non-public data to calculate surge prices, and motivates us to examine alternative strategies for obtaining lower prices in §6.

5.5 Impact of Surge on Supply and Demand

The final question that we address in this section is: *what is the impact of surge prices on supply and demand?* Uber has stated that the goals of surge pricing are to increase supply and intentionally reduce demand, and they claim that the system increased the number of drivers by 70-80% after it was introduced [11]. However, it is unclear if and how surge pricing continues to impact supply and demand, now that drivers and passengers have acclimated to the system. Furthermore, recent measurements of Uber suggest that surge pricing redistributes existing supply, rather than encouraging new drivers to come online [6], but these observations have not been verified at-scale.

[7]The two exceptions to this data cleaning rule are intervals where surge=1 directly preceding or proceeding an interval where surge>1.

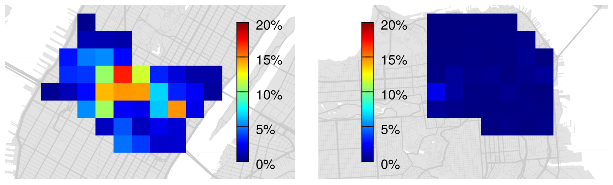
To answer these questions, we treat the cars in our data as state-machines, and examine how they transition between states when there is and is not surge. At a high-level, we divide time into 5-minute intervals, and compare the states of cars at the beginning and end of each interval. Cars that appear for the first time in interval t are placed in the initial, **new** state, while cars that disappear go into the terminal **dying** state. Cars that start and end in surge area a are **old**. Finally, cars may transition into states that are relative to surge areas, e.g., a car that moves from area a_i to area a_j during t is placed in the move-**in** state relative to a_j, and the move-**out** state relative to a_i.

Based on this model, we examine the behavior of cars during times when all four surge areas have the *same* surge multipliers (in Manhattan and SF, respectively), and times when a single area has a surge multiplier that is at least 0.2 *higher* than its neighboring areas (again, excluding jitter) in the immediately proceeding interval. Intuitively, the former case captures times when there is no monetary incentive for drivers to choose one area over another, while in the latter case there is a monetary incentive to relocate to the surging area.

Figure 22 shows the probability of cars in specific surge areas being in each of our five states. The black bars show the probabilities when all areas have equal surge multipliers, while red corresponds to times when the given area has a multiplier that is at least 0.2 higher than its neighbors. For example, the Manhattan Area 1 "New" bars show that 18% of new cars appear in the area when surge is equal in all four Manhattan areas, whereas 20% of cars appear in the area when it has higher surge than its neighbors. We omit results for two areas because they rarely had higher surge prices than their neighbors.

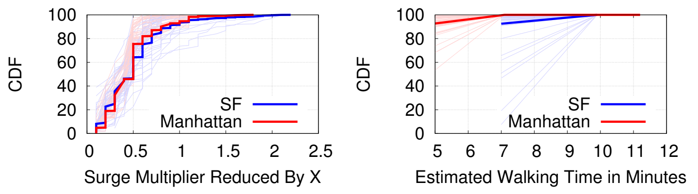
Results. First, we examine the impact of surge pricing on the behavior of **new** cars. In all six areas, we observe that the fraction of **new** cars increases when that area has higher surge than its neighbors. Although this effect is not large (3.7% on average), it is consistent. This suggests that surge pricing is effective at drawing drivers onto the roads.

Second, we examine the impact of surge on the distribution of existing supply. In five areas, we observe that the fraction of cars that move **out** of an area increases when it is surging. This is the opposite result of what we expected. Furthermore, in three areas (Manhattan 2, SF 0, and SF 2) we observe more cars moving **in** during times of surge, while



(a) Manhattan (b) SF

Figure 23: Fraction of the time when passengers would receive lower surge prices on UberXs by walking to an adjacent area.



(a) Surge reduced. (b) Walking time.

Figure 24: Amount that surge is reduced and walking time needed when passengers walk to an adjacent area.

in three other areas (Manhattan 1, Manhattan 3, and SF 3) we observe less cars moving **in** during surges. This result is inconclusive, and also unexpected: we assumed that cars would flock to the surging area.

Third and finally, we examine the impact of surge on passenger demand. In all six areas, the fraction of **old** cars increases while **dying** cars decrease within the surging area. Both of these observations point to reduced demand within the surging area.

Discussion. The results in Figure 22 paint a complex picture of surge pricing's impact on supply and demand. On one hand, surge does seem to have a small effect on attracting new cars. On the other hand, it also appears to have a larger, negative effect on demand, which causes cars to either become idle or leave the surge area. Although we cannot say with certainty why surge has such a large, negative effect on demand, one possibility is that customers have learned that surges tend to have short duration (see Figure 13), and thus they choose to wait for 5 minutes before requesting a ride. Another possibility is that surging areas may be impacted by adverse traffic conditions, which prevents drivers from flocking to them.

To make the surge pricing algorithm more effective, we propose that Uber alter the algorithm to update surge prices more smoothly. For example, rather than oscillating between periods of no and high-surge, Uber could use a weighted moving average to smooth the price changes over time. This would make surge price changes more predictable and less dramatic, which may encourage driver flocking, as well as discourage sudden, temporary drops in customer demand. Another alternative would be for Uber to adopt Sidecar's pricing approach, in which drivers set their own prices independently. This free-market approach obviates the need for a complex, opaque algorithm and empowers customers to accept or decline fares at will.

6. AVOIDING SURGE PRICING

In the previous section, we show that short-term surge prices cannot be forecast, even with large amounts of data directly from Uber. This is disappointing, since forecasting short-term changes in surge prices would be a useful capability for drivers and passengers.

In this section, we propose an alternative method that passengers can use to obtain lower prices from Uber. Since we cannot forecast surges, this means that only the price information during the current 5-minute surge interval is reliable. Thus, our goal is to locate the lowest price car for the passenger given their current location and the instantaneous surge prices.

Our Approach. Our key insight is to leverage our knowledge of surge areas. Suppose a user observes that the surge multiplier at their current location is m', and there are a set of adjacent surge areas A. We can use the Uber API to query the surge multiplier m_a and EWT e_a for each $a \in A$, as well as the walking time w_a to each area. If $m_a < m'$ and $w_a \leq e_a$ for some a, then this means the user could reserve an Uber immediately at a lower price, and walk to the pickup point in the adjacent area before the car arrives. Startups have proposed similar approaches to avoiding surge prices [30], however their techniques do not leverage precise knowledge of surge areas, or take EWTs into account.

Results. To demonstrate the feasibility of our approach, we plot Figure 23, which shows the percentage of time that each of our 43 measurement clients could have obtained a cheaper UberX using our approach. We assume that people can walk 83 meters per minute (i.e., 5km/hour). We also assume that, for our technique to be implemented in practice, it would need to rely on data from the Uber API. Thus, surge prices change every 5 minutes and there is no jitter, but EWTs may change moment to moment.

In Manhattan, we see that users around Times Square would have been able to request cheaper cars 10-20% of the time, depending on the user's precise location. In contrast, users in SF would not generally benefit from our approach; only users at UCSF would be able to save money 2% of the time. Our approach works better in Manhattan for two reaons: first, the surge areas in Manhattan are smaller, so it is more feasible for users to walk from one area to another in only a few minutes (50% of EWTs on Uber are less than 3 minutes, see Figure 11). Second, the surge areas in SF tend to be more correlated than those in Manhattan, i.e., it's rare for one area in downtown SF to have significantly higher surge than all the others.

Figure 24(a) shows how much surge multipliers are reduced when users reserve Ubers using our approach. The faded lines correspond to our 43 clients in Manhattan and SF, while the solid lines are the combined results. We see that our approach brings substantial savings: in more than 50% of cases, surge multipliers are reduced by at least 0.5, i.e., a 50% savings or more.

Figure 24(b) shows how many minutes users would need to walk while using our approach. In all cases, users walk for less than 7 and 9 minutes to meet the car in the adjacent surge area in Manhattan and SF, respectively. Note that the surge areas in SF are larger than those in Manhattan, so the shortest walks in SF are 7 minutes long, versus 5 minutes in Manhattan. Overall, these walking times are quite reasonable, given the dramatic savings that this strategy enables.

7. RELATED WORK

Uber. Recently, researchers have begun to take an interest in Uber. Salnikov et al. compared 2013 NYC taxi and Uber API data to reveal when and where taxis are cheaper than Ubers [31]. In [6], Diakopoulos tracked surge prices via the Uber API in multiple locations around Washington D.C., and identified the same 5-minute surge update behavior that we observed, as well as a positive correlation between EWT and surge prices. Lee et al. surveyed the attitudes of Uber and Lyft drivers towards algorithmically directed work, surge pricing, and the driver-review system [18].

Auditing Algorithms. There is a growing body of literature that aims to understand the algorithms that impact people's daily lives. [7, 8] examined user perceptions of algorithmically curated content in the Facebook News Feed. [14] determined the features used by Google Search to personalize content, while [13, 20, 21] measure price discrimination and steering on e-commerce sites. [1, 12, 17] investigate online tracking techniques. Most disturbingly, [9, 29] revealed the existence of racial discrimination on AirBnB and Google Ads.

8. CONCLUDING DISCUSSION

In this paper, we present the first in-depth analysis of Uber. We leverage four weeks of data collected from Uber's smartphone app and official API that covers midtown Manhattan and downtown SF. In total, we collected 2.1 TB of data on supply, demand, EWTs, and surge prices for Uber vehicles. We validate the fidelity of our methodology using ground-truth information about all taxi rides in NYC in 2013.

Our results reveal high-level characteristics about Uber's service and the impact of surge pricing. We observe that SF has 3$\times$ more surges than Manhattan even though SF also has a much greater supply of cars. Furthermore, surge pricing is noisy: the majority of surges last less than 10 minutes. Finally, we observe that on a micro-scale, surge prices have a strong, negative impact on passenger demand, and a weak, positive impact on car supply. However, it is possible that different effects may occur at the macro-scale (i.e., a whole city).

Algorithmic Transparency. Using our measured data, we are able to infer some implementation details of Uber's surge pricing algorithm. Uber has divided cities up into discrete areas, and surge prices are updated in each area independently on a 5-minute clock. Our correlation analysis suggests that average estimated wait times, and the difference between supply and demand, are used to calculate surge multipliers.

Our investigation uncovered a bug in Uber's systems that was causing some customers to randomly receive out-of-data surge price information. To Uber's credit, after we informed them of this issue, they acted quickly to fix the bug. However, the existence of this "jitter" bug serves as a cautionary tale: as the complexity of algorithmic systems increases, they are more likely to exhibit unintended behaviors. Just as white-hat security research is critical for helping companies identify and patch software vulnerabilities, algorithmic audits [28] are a vital tool for identifying problematic behaviors exhibited by algorithmic systems.

We argue that Uber's reliance on discrete surge areas introduces unfairness into their system: two users standing a few meters apart may unknowingly receive dramatically different surge multipliers. For example, 20% of the time in Times Square, customers can save 50% or more by being in an adjacent surge area.

Lastly, we argue that Uber's reliance on black-box algorithms makes their system more vulnerable to manipulation than other online marketplaces. The forces at play on markets like eBay and AirBnB are well understood: the supply of goods is transparent, and prices are set by competing individuals. In contrast, Uber does not provide data about supply and demand, and the pricing algorithm is opaque. This leads to situations where, once the algorithm is known, it can be manipulated. For example, we show in §6 that users can obtain lower prices by exploiting differences between surge areas. Meanwhile, Uber drivers discuss strategies for inducing surges by colluding to artificially decrease supply [2]. Although we were unable to detect collusion in our dataset, our inability to track drivers over time limits our ability to detect this attack.

Acknowledgements

We thank the anonymous reviewers for their helpful comments. This research was supported in part by NSF grants CNS-1054233, CNS-1319019, and CHS-1408345. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF.

9. REFERENCES

[1] G. Acar, C. Eubank, S. Englehardt, M. Juarez, A. Narayanan, and C. Diaz. The Web Never Forgets: Persistent Tracking Mechanisms in the Wild. *CCS*, 2014.

[2] Anonymous. Uber Brotherhood, and a Few Sisters Too. Confessions of an Uber Driver Blog, 2014. `http://bit.ly/SVv4RE`.

[3] Anonymous. Empty Promises. Confessions of an Uber Driver Blog, 2015. `http://bit.ly/1c4JZmZ`.

[4] M. Boland. Apple Pay's Real Killer App: The Uber-ification of Local Services. Huffington Post, 2015. `http://huff.to/1IxEE3M`.

[5] L. Clark. Uber denies researchers' 'phantom cars' map claim. Wired, 2015. `http://www.wired.co.uk/news/archive/2015-07/28/uber-cars-always-in-real-time`.

[6] N. Diakopoulos. How Uber surge pricing really works. The Washington Post, 2015. `http://wapo.st/1yBf3EO`.

[7] M. Eslami, A. Aleyasen, K. Karahalios, K. Hamilton, and C. Sandvig. FeedVis: A Path for Exploring News Feed Curation Algorithms. *CSCW*, 2015.

[8] M. Eslami, A. Rickman, K. Vaccaro, A. Aleyasen, A. Vuong, K. Karahalios, K. Hamilton, and C. Sandvig. "I always assumed that I wasn't really that close to [her]": Reasoning about invisible algorithms in the news feed. *CHI*, 2015.

[9] B. G. Edelman and M. Luca. Digital Discrimination: The Case of Airbnb.com. *SSRN*, 2014.

[10] A. Griswold. Does Uber's Surge Pricing Take Unfair Advantage of Drunk People? Slate, 2014. `http://slate.me/10s0zXH`.

[11] B. Gurley. A Deeper Look at Uber's Dynamic Pricing Model. Above the Crowd, 2014. `http://bit.ly/1fmNgI1`.

[12] S. Guha, B. Cheng, and P. Francis. Challenges in Measuring Online Advertising Systems. *IMC*, 2010.

[13] A. Hannak, G. Soeller, D. Lazer, A. Mislove, and C. Wilson. Measuring Price Discrimination and Steering on E-commerce Web Sites. *IMC*, 2014.

[14] A. Hannak, P. Sapiezyński, A. M. Kakhki, B. Krishnamurthy, D. Lazer, A. Mislove, and C. Wilson. Measuring Personalization of Web Search. *WWW*, 2013.

[15] J. V. Hall and A. B. Krueger. An Analysis of the Labor Market for Uber's Driver-Partners in the United States. *Princeton Working Paper 587*, 2015.

[16] D. Kedmey. This is How Uber's 'Surge Pricing' Works. Time, 2014. `http://time.com/3633469/uber-surge-pricing/`.

[17] M. Lecuyer, G. Ducoffe, F. Lan, A. Papancea, T. Petsios, R. Spahn, A. Chaintreau, and R. Geambasu. XRay: Enhancing the Web's Transparency with Differential Correlation. *USENIX Security*, 2014.

[18] M. K. Lee, D. Kusbit, E. Metsky, and L. Dabbish. Working with Machines: The Impact of Algorithmic and Data-Driven Management on Human Workers. *CHI*, 2015.

[19] E. Mazza. Uber Raises Fares During Sydney Hostage Crisis, Then Offers Free Rides. The Huffington Post, 2014. `http://huff.to/18W6ybk`.

[20] J. Mikians, L. Gyarmati, V. Erramilli, and N. Laoutaris. Detecting Price and Search Discrimination on the Internet. *HotNets*, 2012.

[21] J. Mikians, L. Gyarmati, V. Erramilli, and N. Laoutaris. Crowd-assisted Search for Price Discrimination in E-Commerce: First results. *CoNEXT*, 2013.

[22] A. Monroy-Hernández. NYC Taxi Trips. GitHub, 2014. `http://www.andresmh.com/nyctaxitrips/`.

[23] K.M. Novak and T.C. Kalanick. System and method for dynamically adjusting prices for services. US Patent App. 13/828,481, 2013.

[24] I. Oh. Uber Will Stop Charging Ridiculous Prices During Emergencies. Huffington Post, 2014. `http://huff.to/1qJLs5c`.

[25] B. Popper. Uber kept new drivers off the road to encourage surge pricing and increase fares. The Verge, 2014. `http://bit.ly/1hoPgU8`.

[26] A. Rosenblat. Uber's Phantom Cabs. Motherboard, 2015. `http://motherboard.vice.com/read/ubers-phantom-cabs`.

[27] B. Stone. Uber is Winning Over Americans' Expense Accounts. Bloomberg, 2015. `http://bloom.bg/1IFZ0p7`.

[28] C. Sandvig, K. Hamilton, K. Karahalios, and C. Langbort. Auditing algorithms: Research methods for detecting discrimination on internet platforms. *Proceedings of "Data and Discrimination: Converting Critical Concerns into Productive Inquiry", a preconference at the 64th Annual Meeting of the International Communication Association*, 2014.

[29] L. Sweeney. Discrimination in Online Ad Delivery. *SSRN*, 2013.

[30] T. Schmidt. SurgeProtector. iTunes App Store, 2014. `http://apple.co/1GxhDMU`.

[31] V. Salnikov, R. Lambiotte, A. Noulas, and C. Mascolo. OpenStreetCab: Exploiting Taxi Mobility Patterns in New York City to Reduce Commuter Costs. *CoRR*, abs/1503.03021, 2015.

[32] M. Wilson. Uber Fudges the Position of Local Drivers, But They've Got a Pretty Good Reason Why. Fast Company, 2015. `http://www.fastcodesign.com/3049169/uber-fudges-the-position-of-local-drivers-but-theyve-got-a-pretty-good-reason-why?utm_content`.