CX 4242 Team 26 Final Report

Grace Busby, Nicholas Darby, Emma Hudock, Daniel Kim, Harrison Smith, Christie Suszko

1 Introduction

What did we do? We created an interactive map of metro Atlanta that predicts current and semi-accurate estimates of the travel time of an Uber trip between two locations.
How is it done today; what are the limits of current practice? Uber movement is an existing interactive map tool that attempts to inform of the travel time of an Uber ride between general destinations. It estimates these times solely based on previously aggregated data of Uber rides and is limited by using chunks of locations rather than specific addresses. It is not a live feed that accounts for wrecks, weather, or other anomalies that might impact traffic or provide wait times or an estimated price for an Uber ride.
What's new in your approach? Why will it be successful? In our approach, we intend on providing live estimations travel times between two exact locations. We accomplished this by using a combination of previously aggregated data and real time data from sources the likes of Uber and Google Maps in order to craft accurate price and time estimates that take into account current traffic happenings. We have decided to use a web app platform, but we did initial research into the way ride share currently works on mobile devices. An article showed the system design of the Uber app, their models, and how they handle their failures [14]. This will be helpful in deciding on how to define the map regions within our platform, but it lacks any explanation on how Uber handles traffic in their models.
Who cares? The ride sharing industry is a rapidly growing market that competes with traditional taxis due to modern internet-based mobile technology with more drivers from ride shares spending a significantly higher amount of time with riders [6]. From initial research we see that people prefer ride shares to traditional taxis, but we conducted more research on ride share time estimation. An interactive map that allows for viewing trip times between multiple places on one map while taking into account current traffic conditions would be a valuable tool for planning trips while minimizing costs and eliminating wait time. It will be a valuable tool in densely populated areas where the time between destinations play a major role in determining transportation. The users, in this case Uber customers, will find this application extremely useful for getting realistic travel times so they can accurately plan their travels while using the service. They will have better expectations than what are already provided by the company in the app and tools like Uber Movement. Furthermore, if ride share providers choose to, they could use this algorithm and visualization within their own services to more effectively support their customers.
If you're successful, what difference and impact will it make, and how do you measure them? Our project could impact the way people interface with ride share apps, booking in advance due to more knowledge of travel times. It will also impact city planners, which is explained in an article that used Uber Movement to help develop city planning [11]. Since Uber Movement is using past data, our platform will have a larger impact. This is measured by the number of people using our map cross referenced with the increase in planned ride share trips.
What are the risks and payoffs? Our tool could be not desired and it does not improve user experience with ride share services. However, the payoff is that it could revolutionize the way people interface with, plan with, and use ride share services. The introduction of ride sharing has led to a significant decrease in alcohol related crashes in Houston and an increase in the planning of ride shares with our platform could have the same affect on Atlanta [4].
How much does it cost? Free, as we are anticipating using entirely public data.
How long did it take? We predict it was about thirty hours per team member, around 180 hours total.
What are the midterm and final "exams" to check for success? How will progress be measured? We measured progress first by checking our outputs compared to readings on ride share apps and our final check is comparing to Google Maps.

2 Problem Definition

In today's current availability, ride share services are able to provide estimated travel times in app and a more general visual map on each providers own website. While users are able to have a general idea of how long it will take them to travel, they lack the ability to get exact travel times as these estimates lack the ability to fluctuate with traffic variations. Ride share customers have high expectations for these companies, so with many unexpected delays it leads to stress and frustration for the users, drivers and the service providers. With current data collection and an algorithm we develop, we took a step to improve current predictive models by creating a visualization that can provide up to date and semi-accurate travel times between two exact travel destinations. Our interactive map is able to take into account traffic variations based on the time of day and the day of the week along with the impact of weather using current data, rather than solely using calculations of past data and general locations. This tool will allow Uber users to have more realistic travel times, saving future frustrations and ease of planning for their next trip. It will also be advantageous for the service providers, as they can provide customers with more reasonable travel estimates if implemented into their programs.

3 Literature Reviews

The significant goal of our project is predicting travel times. We reviewed a case study that was conducted on data collection and machine learning methods in the context of predicting travel times with Uber movement data [13]. This gave us insights on how to select which features to use in our model when determining travel time. Another literature review focused on predicting travel time patterns on the Traffic Analysis Zones (TAZ) to propose a model based on an origin to destination travel time matrix using graph analytics and optimization methods [12]. The forward model and non-negative least squares optimization that this research uses can be used as inspiration in our platform, but needed to research into how to dive further into TAZs with supplemental data. Another article compared and expected travel times of trips from Uber and Google Maps, determining the validity of the crowd-sourced data sets [16]. Taking into account Uber driver motivations for completing a trip in a short time explains why Uber has shorter travel times than Google, but when adjusted, the difference in travel times are statistically insignificant. We looked into using Google Maps as travel time data as a comparison tool to cross verify our ride share travel time results. At Uber probabilistic time series forecasting is used to predict the number of trips, but one article suggested using end-to-end Bayesion deep model for time series prediction with uncertainty estimation, which we looked into for our platform [17]. The final article for predicting travel time focused on factors that impact the supply and demand limitations of on-demand ride-share services [8]. It is useful to us as it provides analysis on the less thought of characteristics that affect the availability and efficiencies of ride share services in an area. This insight coupled with drive cycle producing data would help us implement an accurate gauge on timing of a ride share trip by knowledge of regional demographics. This article suffers from a lack of research due to the infancy of the market.

Another aspect that we researched was predicting pricing. One article explained statistical pricing against dynamic pricing for ride share platforms in the context of a Markov Chain and queuing processes for the riders and drivers perspectives [2]. This is useful to understand the different variables that influence pricing when the rider opens the app to receive service. This article did not incorporate pick up time or how far the driver is when the rider looks for the ride. Another article explained pricing based on location using weather and time series analysis in NYC, which would be useful for predicting prices in Atlanta [4]. However, due to lack of research and lack of innovation we decided no longer to include predictive pricing.

We researched a new approach on how to make predictions for the total number of rides in a day. Ideally, it would account for uncertainty, which will be helpful in our development [18]. After looking into these ideas, we decide to exclude this process from our project as there is not enough current data to make these predictions. The model presented in this article took into account model uncertainty, model misspecification, and inherent noise. The explanation of how networks that are used can be utilized will be a good start on deciding what networks will work best for our project. This article does not address external factors that may impact predictions, which we researched. Another factor to consider is the number of drivers, which will alter travel availability [3]. We used our research to consider this in our algorithm. This article gives insight on what variables and constraints to develop.

Drive cycle patterns were developed in our algorithm. Research summarized the process of formulating accurate drive cycle patterns and behaviors for the city of Chengdu, China based on data that provides GPS location trajectories from ride share passenger cars [9]. We created an algorithm for Uber rides in Atlanta and will reference this research when developing characteristics and constraints, but we did not rely on solely GPS data. Another article explained how to construct a weighted time travel index that acknowledges congestion for each traffic zone, which will help when dividing Atlanta into traffic zones for our algorithm, but we had to decide how to split up traffic zones because this research did not explain how they did it [15].

Our platform handles big data. In our research, we found an article that explained how Uber handles these in real time with recovery plans for outages. This is helpful, but we had to apply it for all rideshare platforms

and not just Uber, which is what the article focuses on [7].
We found an article which focused on matching drivers and riders with the goal of minimizing system wide miles driven in the city of Atlanta. Since our platform is based in Atlanta, their simulation was valuable in building our model, specifically how they split up regions in the city [1].
Our platform also models our algorithm so we researched visualization techniques for graphical representations which will be useful although the article did not go into visualizations for maps, which we used [10].

4 Proposed Method

Data Collection/Processing:
In order to create our interactive map of Atlanta with travel time for Uber rides, we started with data that incorporates the necessary factors to which we can build our model on. First, we downloaded a dataser from Uber Movement of about 13 million rows, which consisted of travel data from 2020. It has many useful features, but we eventually decided to take a subset including date, which was used to extract the day of the week, hour bucket, which groups together hours of the day with similar traffic patterns, starting and ending census tract ids from which we can calculate distance, and the mean travel time, which was used in training our model. We then combined this data with weather data for each documented trip we had data for from the National Centers of Environmental Information API, which included temperature, precipitation, and wind speed. This data processing was done within SQL and Open Refine. Finally, we normalized the dataset to allow the model to be trained more precisely. To normalize the data, the mean of each column was calculated, then the mean was subtracted from each row, and finally that was divided by (maximum value - minimum value). This combined dataset is what was used to train and test the model that is described next. For our interactive map we used the GeoJSON file of Atlanta census tracts provided by Uber Movement to map the census coordinates for metro Atlanta. For the supplemental data that accompanies user input, weather data was collected from Visual Crossing's weather API for the specific date and time prompted by the user. To process the user input start and destination points, we needed to find the coordinates of these locations. In the case the user input an address, we used the Python geopy library to convert the address to coordinates, and the United States Census Bureau API to convert the address to a census id, and in the case the user input a census tract id, we calculated the centroid coordinate from the GeoJSON file. Then, the distance between the start and destination could be calculated with the Haversine Formula. This computes the great circle distances between two points on a sphere by approximating the earth distance.
Algorithm:
With our data, we determined that the most influential way of filtering and discerning connections in this data is through a regression neural network method. Our main goal with the neural network was to obtain predictions for travel times through running a repeated series of cross training and testing on mini-batches of data that endeavors to recognize underlying relationships between all the factors attributed to census id paths in order to produce an accurate, anticipated mean travel time. The neural network was built using pytorch and is made up of 7 linear layers that first take in the 6 starting features (distance, day of the week, hour bucket, temperature, precipitation, and wind speed) and then map them to 64, 32, 16, 8, 4, and 2 features, resulting in the final output of expected travel time. The network makes use of the Leaky ReLU nonlinearity function, mean squared error (MSE) loss function, and Adam optimizer while training. Originally, the model did not result in the most accurate estimates as the error averaged around +/- 11 minutes. However, after fine-tuning hyperparameters such as the optimizer and nonlinearity parameters, normalizing the data used to train, adding a linear layer, and decreasing the number of features we were considering, we were able to boost the accuracy so that the estimates average an error of +/- 5 minutes. Since tuning the neural network to its highest accuracy, we have a model that will output an anticipated travel time when given input start and end locations along with date and time based off our data processing methods.
Visualization:
After completing our algorithm, we created an interactive map of metro Atlanta that responds and updates when two designated locations on the map are clicked to produce the predicted travel time between those two locations. When clicking certain locations, our tool finds the center coordinate of the census ID based on averages of the GeoJSON, and that is used to compute the distance and therefore travel time. Our locations on the map are demarcated by the Census ids we have obtained from our Atlanta Census Tract data. Alternatively to clicking, our interactive map also has the ability to type in specific locations for start and end destinations and have them populate on the map after they are translated into Census id areas. Once the user has entered their desired inputs, they can click a button that then displays to them the estimated travel time based the the conditions they gave.

List of Innovations:
1) We are able to handle finding travel times between two exact locations that are mapped to census tract ids rather than only having the option to select census tracts.
2) Through the use of a localized interactive map of Atlanta, we are able to predict the estimated travel time for Uber rides between two points based on the current conditions for the time and day entered by the user rather than just relying on aggregated past data.

5 Experiments/Evaluations

We have successfully conducted multiple experiments on our neural network algorithm for predicting travel times. It should be noted that we have deemed this successful not on the grounds of completion and accuracy, but on the grounds of consistent results, ability for improvement and progress, and eventual provision of trustworthy and reliable results. To ensure that our model and visualization worked as intended, we performed the following experiments and evaluated how they worked, and if they did not pass our expectations, we had to alter our approach or fix mistakes.

5.1 Map manipulation

In order for the user to be able to see the map clearly, we decided to give them the ability to zoom in and out and move around the map using the buttons above the map. Testing this simply included ensuring that clicking each button performed the correct task on the map and that the map is easy to navigate.

5.2 Map interactions fill input data

Because our product gives the user the option to either type in their own start and end locations or select the census tracts from the map, we had to make sure that double clicking a specific area of the map set the start location properly and single clicking a specific area of the map set the destination location properly. This can be seen to be functioning as when you click on a census tract in the map, the respective field on the left is populated with the matching id number.

5.3 Connection between user input data and back end algorithm

A crucial aspect of our product is the user's ability to pass in inputs that they wish to be used when calculating the estimated travel time. In our product, the user interface is a separate entity than the algorithm that runs in the back end, so we had to be sure that when a user enters data, the back end is able to extract this data to be used in the algorithm. This was done by building an API that communicates between the front and back end and passes the necessary fields between the two. Upon completion of implementing this, we were able to print out the inputted data from the back end Python script, verifying that this connection was complete and functional.

5.4 Getting day of week and hour bucket from date and time

Within the back end algorithm, we must parse the given date and convert it to the day of the week. Originally, the function we used to do this did not give matching results as the function used when processing the training data because we were using Python and SQL versions of the function respectively, but with some minor adjustments to the output of this function, we were able to achieve proper day of the week conversions. The hour buckets were predetermined in the Uber Movement data, so we just had to ensure that we were able to map input times to the same hour buckets.

5.5 Differentiate and process start and destination inputs

This task was a bit more involved since we needed to test the functionality based on whether an address or a census tract id was given. If census tract ids were given, then we retrieved a center coordinate for that tract based on a mapping creating from the Altanta Census Tract GeoJSON file to be used in the distance calculation. Otherwise, if an address was given, the address had to be converted to coordinates via the Python library geopy. Many tests were necessary here to ensure that the coordinates from the address were correct and the address was interpreted as the correct address, and that the center coordinates for a census tract was relatively close to the center. In order to differentiate between a census tract id and an address, we performed a test on the input data to see if it could be cast to a number. In the case that it is an address, the string with words would throw an error, which we could decipher as meaning it is an address. We also had to account for invalid addresses and census tract ids. In this case, if the back end algorithm is unable to recognize an address or census id and can't get the coordinates, an alert message appears to the user prompting them to input valid locations. All of these tests were able to be passed, where if given an address, we are able to calculate the proper coordinates, and if given a census id, we can find the center coordinate.

5.6 Computing distance between coordinates

Using the coordinates calculated by the means described above, we had to make sure we were able to get the correct distance between the two points. After some research, we found the Haversine formula, which yields the great-circle distance between 2 points on a sphere. Although the calculations were idealized by the assumption that the earth is a perfect sphere, upon comparing the results to actual distances computed by Google Maps, the calculated distances were quite accurate.

5.7 API connection to retrieve current weather for input date and and time

In order to obtain all the necessary features to pass into the neural net, we needed to establish a connection to an API that would provide us with the necessary weather data. Once we obtained an API key for Virtual Crossing's weather API, we were able to test the connection by retrieving the current days weather, which passed our expectations.

5.8 Obtaining a result from the neural network

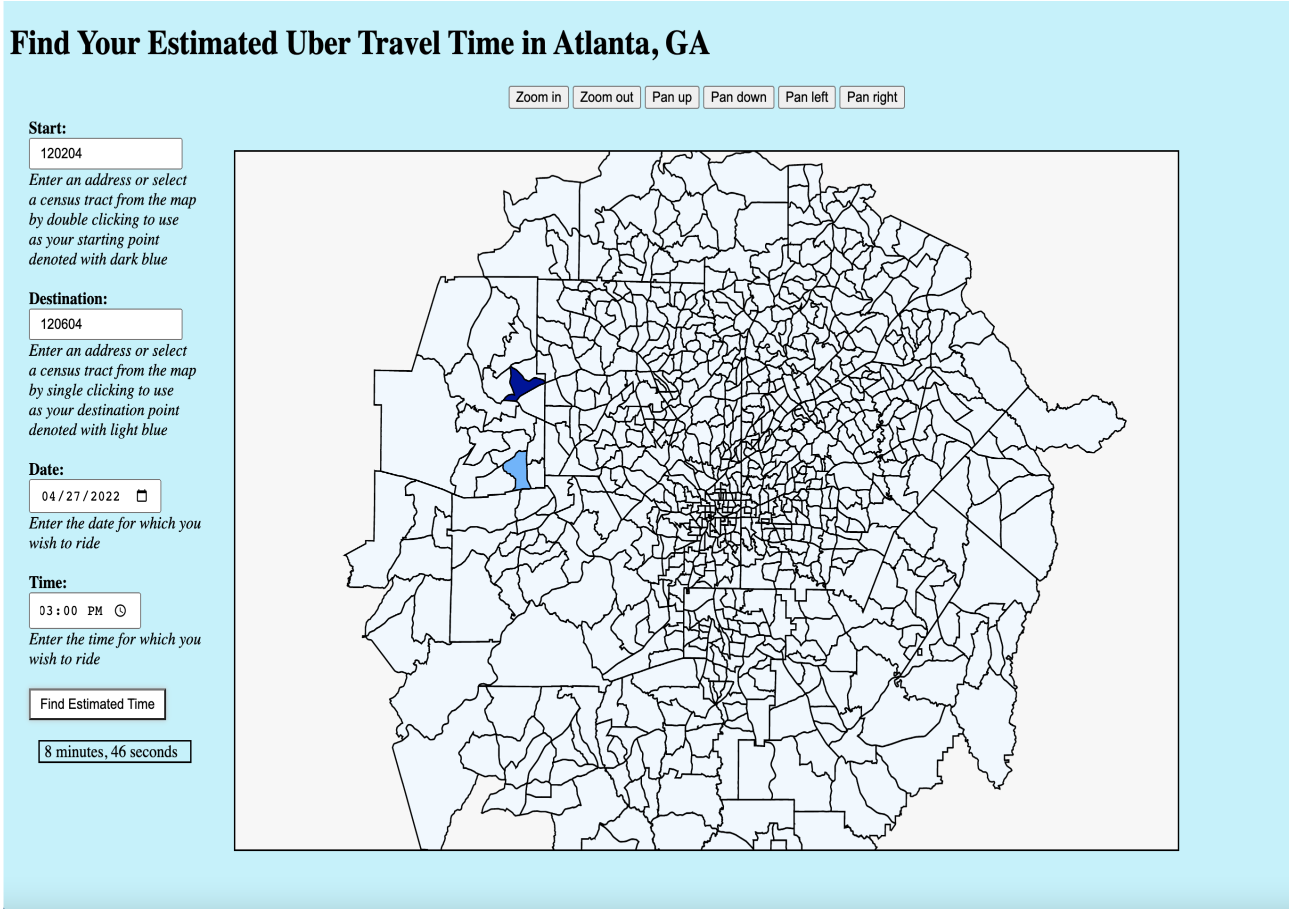
The main part of this product is encompassed by the neural network outputting a semi-accurate estimation for the travel time. As previously mentioned, the initial results of the net were not very accurate, and there was room for improvement. Upon altering the neural nets layer structure, fine-tuning several hyperparameters, and normalizing the training data, we were able to improve the accuracy of the algorithm. The current average error of an estimated travel time is about +/- 5 minutes as opposed to the initial +/- 11 minutes. Although this is still not incredibly accurate, we considered this test to pass because of the amount of improvement we were able to achieve.

5.9 Passing estimation from back end to user interface

Finally, in order for the product to be effective, we needed to be able to display to the user the result from the algorithm. As described before, the back end algorithm is separate from the user interface, so using the same API for connecting the user interface to the back end, we were able to return the result from the back end to the user interface. This test passed since we are able to display the result properly on the web page.

6 Conclusions and Discussions

Conclusion:
Using data from Uber Movement, the National Centers of Environmental Information, and the Atlanta Census Tract GeoJSON file, we successfully created an algorithm that can predict semi-accurate travel times between two specific addresses with an average accuracy of +/- 5 minutes. We also developed a visualization tool that presents travel times with a localized, interactive map of metro Atlanta. The picture illustrated below shows an example of what our tool looks like when predicting a specific travel time. We overcame many difficulties and limitations with the data used to create the algorithm and the neural network, and our model sets a new standard for Uber rideshares.



Find Your Estimated Uber Travel Time in Atlanta, GA

Start: 120204

Enter an address or select a census tract from the map by double clicking to use as your starting point denoted with dark blue

Destination: 120604

Enter an address or select a census tract from the map by single clicking to use as your destination point denoted with light blue

Date: 04/27/2022

Enter the date for which you wish to ride

Time: 03:00 PM

Enter the time for which you wish to ride

8 minutes, 46 seconds

Future Possibilities:
Given more time and computing power, there are other changes we would make to this current product. First, it would have been ideal to collect more information on other features that could affect travel time to include in our regression neural network. With that, and more time, we could fine-tune our model and parameters to get an even more accurate estimate of a travel time between locations. Additionally, when building the neural network, we ran into the limitation of using only select ID's between locations, which make it difficult for predicting forward. This is an area we would have liked to explore more, as it would allow even more ease for customers looking a future traveling.

In our hopes to produce an interactive map that could produce a holistic overview of various ride share estimations, we intended on developing and conducting a similar methodology and approach to allocating and producing predicted prices and wait times for ride share services between two destinations. However, in our search of data, we found that nothing from the available data offers unique innovative potential when compared to the industry standards of current ride share services. Likewise, in our brief experimentation to predict prices and wait times for travels with data from ride share APIs, we found that we could not reconcile a connection point between these predicted estimates when overlayed with our machine learning predictions with travel times. If there was more current data to explore these options, these two features would be added to a future version of this model.

While this interactive map works well within the bounds of Atlanta, there is potential for it to be expanded and to a nationwide scale. This expansion would need a significant amount of data from cities across the country, nationwide census tracts, weather data, traffic patterns, geological information and backing from ride share companies to make it possible. Our innovative neural network and interactive map has the ability to impact the future of ride share programs and the accessibility for customers in major cities.

Statement

All team members have contributed a similar amount of effort.

7 References

[1] Agatz, N. A. H., Erera, A. L., Savelsbergh, M. W. P., & Wang, X. (2011, July 22). Dynamic ride-sharing: A simulation study in Metro Atlanta. Transportation Research Part B: Methodological. Retrieved March 1, 2022, from https://www.sciencedirect.com/science/article/pii/S0191261511000671

[2] Banerjee, Siddhartha, Ramesh Johari, and Carlos Riquelme. “Dynamic Pricing in Ridesharing Platforms.” ACM SIGecom Exchanges 15, no. 1 (2016): 65–70. https://doi.org/10.1145/2994501.2994505.

[3] Courtenay Brown. “Axios: Rideshare Companies Say Driver Shortage Is Pushing Prices Up.” Axios, Newstex, 2021.

[4] Chao, Junzhi. (2019). Modeling and Analysis of Uber’s Rider Pricing. 10.2991/aebmr.k.191217.127.

[5] Conner, Christopher R., et al. “Association of Rideshare Use With Alcohol-Associated Motor Vehicle Crash Trauma.” JAMA Surgery, vol. 156, no. 8, American Medical Association, 2021, pp. 731–38, https://doi.org/10.1001/jamasurg.2021.2227.

[6] Cramer, J., & Krueger, A. B. (2016, March 14). Disruptive change in the taxi business: The case of uber. NBER. Retrieved March 1, 2022, from https://www.nber.org/papers/w22083

[7] Fu, Yupeng, and Chinmay Soman. “Real-Time Data Infrastructure at Uber.” Proceedings of the 2021 International Conference on Management of Data, 2021. https://doi.org/10.1145/3448016.3457552.

[8] Gerte, R., Konduri, K. C., & Eluru, N. (2018). Is There a Limit to Adoption of Dynamic Ridesharing Systems? Evidence from Analysis of Uber Demand Data from New York City. Transportation Research Record, 2672(42), 127–136. https://doi.org/10.1177/0361198118788462

[9] Han, B., Wu, Z., Gu, C., Ji, K., & Xu, J. (2021). Developing a Regional Drive Cycle Using GPS-Based Trajectory Data from Rideshare Passenger Cars: A Case of Chengdu, China. Sustainability, 13(4), 2114. https://doi.org/10.3390/su13042114

[10] Nguyen, V.T., Jung, K. & Gupta, V. Examining data visualization pitfalls in scientific publications. Vis. Comput. Ind. Biomed. Art 4, 27 (2021). https://doi.org/10.1186/s42492-021-00092-y

[11] Roy, Shouraseni Sen, et al. “Analysis of Urban Mobility in South Florida Using Uber Movement.” Case Studies on Transport Policy, vol. 8, no. 4, Elsevier Ltd, 2020, pp. 1393–400, https://doi.org/10.1016/j.cstp.2020.10.003.

[12] Sathanur, A., Amatya, V., Khan, A., Rallo, R., & Maass, K. (2019). Graph analytics and optimization methods for insights from the uber movement data. In Proceedings of the 2nd ACM/EIGSCC Symposium on Smart Cities and Communities (pp. 1–7).

[13] Shokoohyar, S., Sobhani, A., Malhotra, R., & Liang, W. (2020). Travel Time Prediction in Ride-Sourcing Networks: A Case Study for Machine Learning Applications. Papers.ssrn.com. Retrieved 4 March 2022, from https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3534861.

[14] Anu Upadhyay. (2021, December 14). System design of Uber App - Uber System Architecture. GeeksforGeeks. Retrieved March 1, 2022, from https://www.geeksforgeeks.org/system-design-of-uber-app-uber-system-architecture/

[15] Vieira, Renato S., and Eduardo A. Haddad. “A Weighted Travel Time Index Based on Data from Uber Movement.” EPJ Data Science 9, no. 1 (2020). https://doi.org/10.1140/epjds/s13688-020-00241-y.

[16] Wu, H. (2019). Comparing Google Maps and Uber Movement Travel Time Data. Findings. doi:10.32866/5115

[17] Zhu, L., & Laptev, N. (2017). Deep and Confident Prediction for Time Series at Uber. In 2017 IEEE International Conference on Data Mining Workshops (ICDMW) (pp. 103-110).

[18] Zhu, L., & Laptev, N. (2017). Deep and Confident Prediction for Time Series at Uber. In 2017 IEEE International Conference on Data Mining Workshops (ICDMW) (pp. 103-110).

[19] United Sates Census Bureau. Census Geocoder. (n.d.). Retrieved from https://geocoding.geo.census.gov/geocoder/geographies/address

[20] Corporation, V. C. (n.d.). Weather Data &amp; APIGlobal Forecast &amp; History Data. Weather Data &amp; Weather API — Visual Crossing. Retrieved from https://www.visualcrossing.com/

[21] NOAA OneStop. National Centers for Environmental Information (NCEI). (2022, April 12). Retrieved from https://www.ncei.noaa.gov/access

[22] Uber movement: Let’s find smarter ways forward, together. Uber Movement. (n.d.). Retrieved from
https://movement.uber.com/explore/los_angeles/travel-times/query?lat.=34.058211334739156&amp;lng.=-118.24368490000
US&amp;si=1380&amp;ti=1687&amp;ag=censustracts&amp;dt[tpb]
=ALL_DAY&amp;dt[wd;]=1,2,3,4,5,6,7&amp;dt[dr][sd]=2018-01-01&amp;dt[dr][ed]=2018-01-31&amp;cd=&amp;sa;=&amp