A Methodology for Taxi Demand Prediction Using Stream Learning

David Faial[1], Flavia Bernardini[2], Edwin Mitacc Meza[1], Leandro Miranda[2] and José Viterbo[2]
[1]Institute of Science and Technology, Fluminense Federal University, Rio das Ostras, RJ, Brazil
[2]Institute of Computing, Fluminense Federal University, Niterói, RJ, Brazil
dlfaial@gmail.com, fcbernardini@ic.uff.br, emitacc@id.uff.br, {lmiranda,viterbo}@ic.uff.br

Abstract—**Intelligent transport support systems have had a major impact on people's urban mobility. In large urban centers, transportation services still need ways to optimize vehicle supply in certain areas, according to the demand in each of them. Optimized distribution of on-demand taxi services can be part of an intelligent urban mobility plan, causing direct impacts on urban traffic, improving transport accessibility, improving safety at taxi standpoints by reduced waiting times, reduce transportation fare etc. Many vehicle-mounted sensors currently generate real-time information that is not used for processing and generating information with value. This paper proposes a Taxi Demand Forecasting methodology using stream machine learning algorithms that tackle concept drift detection on taxi data stream. A real data source made available on the New York open platform feeds a stream learning model, constructed using the Massive Online Analysis (MOA) tool — a framework for data stream mining. The stream model shows promising results in forecasting taxi demand, reaching 78% accuracy. Despite using data from a specific city, the methodology and results of this work can contribute to a more proactive demand management in other cities.**

Keywords—**urban mobility, stream data mining, big data, predictive algorithm.**

I. Introduction

The urban transport applications such as Lyft, Cabify, Uber, 99Taxi, among other applications, started to transform the transport scenario, drastically changing the population urban mobility [1], [2]. So, this service is growing fast in the last years and it's argued that the main reasons for it are the reliability and the affordability when compared to traditional taxi services. In 2011, with failures in the red subway line in Boston, "leaving 35,000 people waiting in the cold", Uber took the opportunity to offer free races and low prices in the areas affected [3].

In this scenario, for example, the service of New York's traditional yellow taxis needs to adopt a more imperative and proactive strategy in order to remain competitive with transport applications, maximizing driver earnings, offering a better and more attractive service to passengers, and potentially reducing traffic jam. A specific type of problem still calls attention to the demand for taxis in certain regions: regions with high demand and low number of available cars, as well as regions with many idle drivers and low passenger demand. Huang and Powell (2012) [4] presented a framework for detecting imbalance in taxi service regions, a possible solution for passengers who wait too long at taxi stands and drivers who make less money, even with demands in other regions. Liu et al (2017) [5] used a Parallel Genetic Algorithm (PGA) with a strategy to divide regions to tackle the problem between demand and supply of passengers and taxis. The recent paper that inspired our work is from Zhao, Khryashchev and Vo (2019) [6] that tackle the problem of imbalance between demand and supply of taxis, which problem modeling was also used in this work, and explored batch learning algorithms for predicting taxi demand. They used the dataset available by TLC from New York, also used by us. Currently, the TLC (Taxi and Limousine Commission) is the agency responsible for maintaining direct communication with taxi drivers in New York City. The agency alerts drivers where there is a shortage of taxis, but only after this shortage has been identified and potentially affected taxi availability.

This work presents a methodology for using stream learning algorithms in forecasting taxi demand, using an approach with continuous data flow under concept drift. The demand forecasting model of this work could help in a proactive communication strategy, instead of a reactive strategy. The knowledge of where taxis will be available or which areas will have the highest passenger demand can help resolve major imbalances between demand and service offering, as well as idle drivers, long waiting time, fuel costs, traffic jams, etc. The constant change in the data distribution of taxi demands observed in Zhao et al. (2017) [6] can be better represented by a model that perceives changes in the context of the environment from which the data comes. The changes can originate from different characteristics of the districts (neighborhoods) or from events that happen in each one of them.

II. Background

A. Smart Cities

In smart cities, applications produce high-speed data in environments that change over time, producing what we call a continuous data stream. Because of this new form of data availability, new necessities have been identified for machine learning. Some of them are: (i) new data sampling techniques, (ii) new learning algorithms, (iii) techniques for processing data at the speed at which it is made available and the ability to forget old data [7].

The distribution of sensors in vehicles, water and GPS networks produces a massive amount of data about the environment in which they are installed, with enormous potential to be

978-1-7281-7539-3/20/$31.00 ©2020 IEEE

made available in the form of services. These services can be classified as data producers, data consumers or a combination of the two. For example, a set of sensors that mark the number of free parking spots on the streets is a data-producing service, in this case the the service is number of available spots and maybe the coordinates for each one. A car consuming this service can be classified as a data consumer. The third example may come from a taxi equipped with a service that warns other vehicles of traffic conditions, warning of possible dangers and routes to be avoided.

Among the main challenges of smart cities is the construction of an efficient transport system, which can significantly influence daily life of citizens. Thus, the distribution of vehicles through the urban transport network is one of the main factors to be considered to guarantee quality in people's mobility and reduce the occurrence of some problems such as: crowded transport, chaotic traffic, crowded spots with people waiting, few transportation in areas of high demand, accidents and high emission of pollutants.

B. Machine Learning

Nowadays, mathematical models are used for breast cancer detection, credit card fraud, sales projection, consumption and detection of suspicious activity. All of these models can be derived from one of the AI subfields used in this work, called Machine Learning (ML).

Haykin (2009) [8] defines machine learning as "the field of study that gives computers the ability to learn without being explicitly programmed", with its roots in mathematical and statistical optimization, covering forecasting, prescription, diagnosis and planning techniques. ML problems are typically classified into three broad categories, depending on the nature of the*input* ou *feedback* available for the system. Machine learning has a categorization regarding the type of problem to be addressed. If the objective of the problem is to relate each input instance to an element of a discrete set, then this is a classification problem. If the output consists of mapping an instance to an element of a continuous set, so this is a regression model. In this work, we tackle taxi prediction as a regression problem.

Batch and Stream Learning: Batch learning algorithms take the entire data set in a data structure for the learning to be carried out, and the first prediction can only be made when all the learning examples have been analyzed. In contrast, stream learning updates the model every n visited training examples, where the set of these n examples is called a window of size n. According to Domingos and Hulten (2003) [9] and Bifet et al. (2018) [10], there are desirable properties in this type of learning:

- Short access time to each training instance, due to the large number of instances that arrive at all times. A long access time can cause the algorithm to fail;
- Amount of memory used is limited, regardless of the number of examples;
- Single reading of the training example for building the model:
  - There may be no time to revisit old examples;
  - Examples may not be available on secondary storage for future reference;
- Availability of a data model usable at any time of a request, since, as opposed to batch learning, the model is updated with each learning example visited;
- In an ideal scenario, it should be able to produce a model equivalent to the model that would be produced by the corresponding algorithm in batch learning;
- The model must be able to be updated even when there is a change in concept, that is, even when there are changes in the relationships between input and output variables.

Perceptron: Perceptron was firstly proposed by Rosenblatt [11], inspired by the biological, human neural model. It receives inputs x_1, x_2,... x_n and produces a binary output: 0 or 1, inspired by whether or not to trigger an electrical impulse from a biological neuron. A network of perceptrons (neurons) is built with connections between neurons on multiple layers, wchich is called Multilayer Perceptron [12]. These artificial connections are then passed through an activation function. Activation functions in a network are used to compute the weighted sum of the inputs and the bias, deciding whether a neuron will be triggered or not. In addition, they are used to introduce nonlinear properties in the networks. Its main function is to convert the input signal of a neuron in a neural network into an output signal, which is used in the next layer of the network. Among the main types of activation functions we may cite sigmoid, step, ReLu and Softmax. In MOA, Perceptron Online is implemented using stochastic gradient descent for incremental optimization of a loss function on a stream of instances, or examples. This algorithm is called SGD (Stochastic Gradient Descent) in MOA.

C. Concept Drift

One problem with modeling real data streams is that the relationship between variables can change over time, making models used for analysis quickly obsolete. In machine learning, the phenomenon that comprises the change in the probability distribution of the data is known as Concept Drift [10], [13]. In this work, the days of the week, period of the month or weather conditions can cause the rules used in the forecasting model to change, making machine learning more challenging. The change in concept can happen in some ways, making it easier to track when it occurs over time, causing data collected at specific time intervals to reveal a pattern of change in the data. Some ways of changing the data include [14]:

- **Abrupt Change:** occurs when at one point in time t a data source A is suddenly replaced by a data source B;
- **Smooth, temporal change:** occurs in two ways, the first when there are initially two sources of data present in the reading of the instances. Over time the probability of the sample coming from A decreases, while the probability of reading instances of B increases. The second form of gradual data change occurs when the difference between

the two sources A and B is very small, so much so that it is only perceived as the predictor learns over time;

- **Cyclical change:** occurs when contexts that were previously active reappear. It differs from seasonality, as the period of a supposed cycle is not known;
- **Gradual or Incremental change:** occurs when, for a long time, the distribution experiences at each time step a tiny, barely noticeable change, but these accumulated changes become significant over time.

III. Literature Review

In this work, we present some papers found in literature that inspired us to conduct our work. As we were interested in using a huge amount of taxi data available by New York City, we analyzed what reflections and inspirations each work brought to us.

Anwar, Volkov and Rus (2013) [15] presented a tool called ChangiNOW, used to optimize taxi queues at Singapore airport. The statistical model developed is based on two lines of passengers and vehicles per airport terminal. The work models the distribution of passengers by a Poisson distribution, using historical data on the numbers of passengers arriving at each terminal, with 15-minute intervals. The work also estimates the arrival time based on the GPS positions of the taxis running nearby, in real time to the system. This shows that the TLC taxi agency has a lot of potential with the entire data collection in hand, still lacking a strategy on how to use that data. Kamga, Yazici and Singhal (2013) [16] analyze the temporal and climate-related changes in balancing the demand and supply of taxis. The study comes to the conclusion that under any rainy conditions, regardless of the intensity, taxi drivers get more trips. It is interesting to conclude from the same study that snow conditions can hardly affect or affect little the demand for pickups.

More recently, Tong et al (2017) [17] present a spatio-temporal model for forecasting demand per unit of time, in regions of high demand. A n-dimensional linear regression model called LinUOTD is proposed. Coviensky (2017) [18] used tree-based predictive models to predict hourly taxi demand at Laguardia airport. The work also presents a neural network with an accuracy of approximately 50 percent in forecasting demand. In the end, a web application was produced that allowed users to use the trained model to make small demand predictions for taxis at the airport. Despite the results, the environment created at the airport is designed in a way that many passengers need to stay in line waiting for transport taxis. The problem with this situation is that the dataset made available by the New York data platform considers only passengers who actually entered a taxi as a pickup situation, and not those who also demand the vehicle waiting in the queues at the airport.

Our work is more similar to the one presented by Zhao et al. (2017) [6]. They used four predictive algorithms to measure the predictability of the demand for Uber and Taxis. The predictors are Markov predictor (a probabilistic method), Lempel-Ziv-Welch (sequence based method), ARIMA (a time series method) and Neural Network (method based on machine learning). The four algorithms were used in batch mode, where the predictor is generated from learning the entire data set. In this work, we used the same process they used for processing data. However, we used SGD as stream learning to analyze its results. We chose SGD due to efficiency for the mathematical calculations. SGD obtained 78% accuracy with prediction for all the districts, and 89% accuracy for a specific district.

IV. Methodology

This section describes the process of the scientific method of our work. Firstly, we describe a set of features extracted from our investigated dataset. Afterward, Subsection IV-B explains all the geographic data pre-processing and the process of collecting the data. Subsection IV-C describes an analysis of the relationship between the data collected under distribution over time.

A. Dataset Features

We collected two datasets to conduct our experiments, as used by Zhao et al. (2017) [6]. The first data set comes from the ***NYC Trip Record Data***. This public dataset contains: date, time, pickup location and dropoff of passengers, distance of travel, fares, types of payment and number of passengers. These data were provided by the New York Taxi and Limousine Commission by technology providers authorized by the Taxicab and Livery Passenger Enhancement program [19]. Each dataset record was generated by GPS devices installed in each taxi, a total of 13,237 yellow taxis in New York City made 13,813,031 passengers pickups in July 2014. Among the 18 attributes present in this data set, the main ones are; the driver id code, date and time of pickup, date and time of dropoff, travel distance in kilometers, latitude of pickup and dropoff. The second dataset (**NYC PLUTO — *Primary Land Use Tax Lot Output***) is the set of geographic data, showing information about each lot, measurements of each building within a given lot and information about address, owners, use and geographic location expressed in the New York-Long Island coordinate system. This dataset is made available online by the New York City Department of Planning (*NYC Department of City Planning*).

B. Geographic Data pre-processing

The geographic data of the New York districts, provided by the New York City Planning Department, was treated with algorithms in Python, manipulations with Excel and with the QGIS software, used for manipulating files *.shp - shape* format and translating geographic coordinates. Fig. 1 shows a picture that is a vector image of the geographic maps of the Brooklin district, generated during the datapoint mapping.

In order to map each boarding point and the tuple (block, district) corresponding to that point, an algorithm in Python was developed and used. The set of boarding points and the geographic set were loaded into dataframes, data structures from the Pandas library. A p pickup point is associated with a b block when the distance between p and b is less than the distance between p and any other block $b_1, b_2, b_3, ..., b_n$. To

Fig. 1. Vector map of a brooklyn neighborhood

calculate the distance between two points on a sphere (Earth), the haversine formula, which is very important in navigation, was used [6]. The haversine equation is a case of spherical trigonometry, which the mathematical proof is outside the scope of this work. Let θ be an angle in radians. Eq. 1 defines how to calculate $haversine(\theta)$.

$$haversine(\theta) = \sin^2(\frac{\theta}{2}) \tag{1}$$

For each pickup point p, considering that (i) $pLat$ and $pLon$ are coordinates of the pickup point, in Latitude and Longitude; (ii) $bLat$ and $bLon$ are the coordinates of a block b in a district dx; (iii) Earth Radius $R = 6.672, 8Km$, $dLat = Radians(pLat - bLat)$, $pLat = Radians(pLat)$, $bLat = Radians(bLat)$ and $dLon = Radians(BLon - pLon)$, the distance d between the pickup point and the block is given by Eq. 2.

$$d(p,b)=2R\arcsin\sqrt{(\sin(\tfrac{dLat}{2})^2+\cos(pLat)\cos(bLat)\sin(\tfrac{dLon}{2})^2)} \tag{2}$$

or simply:

$$=2R\arcsin\sqrt{haversine(dLat)+\cos(pLat)\cos(bLat)haversine(dLon)} \tag{3}$$

For storing the mapping records, a csv file was used. Each line of the file contains the following fields: mapping id, pickup district name, pickup block id, pickup date, pickup time. It is important to observe that the mapping lasted 3 months, due to the high volume of data both in the pickup set and in the geographic set of the New York terrain. The final infrastructure used was a virtual machine on Google Cloud, with an 8-core processor and 16GB of RAM, running Ubuntu Linux.

C. Data Analysis

In order to learn more about the data set, some analyzes were generated with the instances. Figure 2 shows the relationship between the number of pickups per day of the month. It is possible to notice that despite the number of pickups falling over the month, a general periodic behavior occurs. The periodicity in the general behavior of the demands does not necessarily reflect the demand in smaller granularities (in districts or blocks, for example). Figures 3 and 4 show hourly demands (number of departures per hour) for two blocks, the first for a quiet residential block in Brooklyn, the second for the block next to Madison Square Garden (Manhattan). We can observe that the demands are completely different, in volume, hourly and in variation. The first, a residential block, has demands a thousand times lower than the block next to Madison Square Garden. The block in Brooklyn has a declining demand after 6pm, while the block in Manhattan has a peak in demand after 6pm, when most of the various events at Madison Square Garden begin.

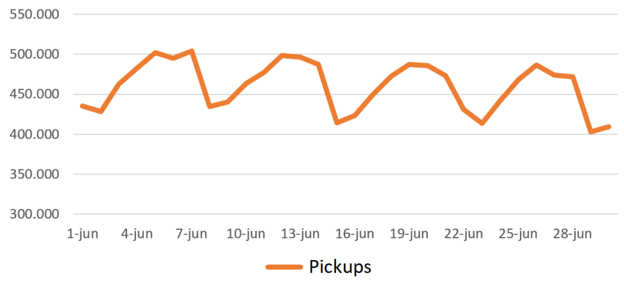

Fig. 2. Daily number of pickups (June of 2014).

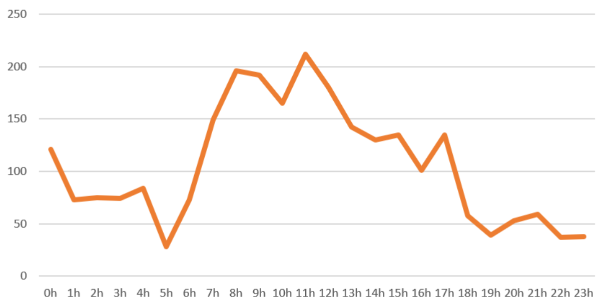

Fig. 3. Pickups on a Brooklyn residential neighborhood

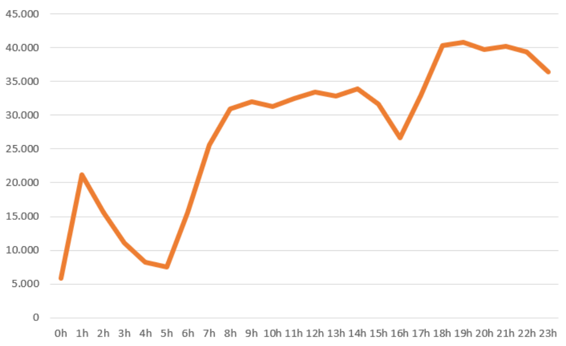

Fig. 4. Pickups on a Madison Square residential neighborhood

Three types of groupings were made in the initial data set, grouping by hour, day of the month and day of the week. It is important to remember that the initial set had 13.8 million instances (shipments). New York has approximately 41,000 blocks, and each grouping resulted in the following number of instances (examples):

- Hourly: 986.808 resulting rows;
- By day of the month: 1.233.340 resulting rows;
- By day of the week: 287.700 resulting rows.

Tables I and II show, respectively, samples of mapping the taxi pickups by hour and day of the week. The frequency column means the number of pickups.

TABLE I
MAPPING, HOURLY EXAMPLE.

District	Block ID	Hour	Frequency
MN	1	0h	68
MN	623	18h	13
BK	34	3h	2
SI	99	17h	20

TABLE II
DAY OF THE WEEK MAPPING EXAMPLE.

District	Block ID	Weekday	Frequency
MN	623	Saturday	508
MN	623	Saturday	127

V. EXPERIMENTS AND RESULTS

An approach using concept drift was implemented for the training of a neural network, using the Softmax activation function. The reason for using the neural network is due to the good (better) performance measured in Zhao et al. (2017) [6], presenting the best performance among the three other predictors also analyzed by the same article. The frequency of the demand (number of demands) is the attribute class. The MOA (Massive Online Analysis) software was used to train the two models. The following parameters were varied in the stream model: the learning rate (we adopted the value 0.01 due to is a traditionally common default value) and size of the evaluation window. The error metric used was MAPE — Mean Absolute Percentage Error.

We firstly trained the stream learning model using all the data from all districts. Table III shows the results of the stream neural network model. The purpose of this table is to show the error variation according to the size of the evaluation window used in the stream learning algorithm. Because they are the most adopted in the literature [10], [14], windows with 200 and 1000 instances were used. In addition to varying the window size, the table also shows the results grouped by hour, day of the month and day of the week (column Grouping), in order to find the best grouping to be worked on.

Defining the best grouping (hourly), Table IV shows the results for each disctrict varying both window size evaluation and learning rate. The tested rates were: 0.01, 0.05 and 0.1, the most commonly found in the researched literature [12]. It is interesting to observe that some districts are less " predictable" than others. The results presented in this table are also subject to variation in the size of the evaluation window.

TABLE III
RESULTS USING STREAM NEURAL NETWORK WITH CONCEPT DRIFT DETECTION, VARIATING THE WIZE OF THE EVALUATION WINDOW. MODEL TRAIDEND WITH ALL DISTRICTS.

Learning Rate	CPU Training Time	MAPE	Window Size	Grouping
0.01	4:30h	0.52	1000	daily
0.01	27min	0.48	1000	weekday
0.01	2:09h	0.23	1000	hourly
0.01	6:02h	0.51	200	daily
0.01	41min	0.44	200	weekday
0.01	3:01h	0.22	200	hourly

TABLE IV
RESULTS USING STREAM NEURAL NETWORK WITH CONCEPT DRIFT DETECTION, VARYING THE SIZE OF THE EVALUATION WINDOW, PER EACH DISTRICT

District	Learning Rate	CPU Training Time	MAPE	Window	Grouping
MN	0.01	57min	0.69	1000	hourly
MN	0.01	1:02min	0.40	200	hourly
SI	0.01	21min	0.27	1000	hourly
SI	0.01	28min	0.11	200	hourly
BK	0.01	32min	0.31	1000	hourly
BK	0.01	38min	0.30	200	hourly
BX	0.01	20min	0.22	1000	hourly
BX	0.01	24min	0.25	200	hourly
QN	0.01	18min	0.53	1000	hourly
QN	0.01	22min	0.45	200	hourly

For purposes of comparison with the work of Zhao et al. (2017) [6], the grouping (segmentation) of demands per hour should be considered. Table III shows the results of the stream model, considering the type of grouping and size of the evaluation window. The last line of the referred table shows that the frequency grouping per hour and an evaluation window of 200 examples produce a 0.22 MAPE error in taxi demand prediction, almost twice lower than the noticed errors in the groupings by the day of the month and by the day of the week. Based this result and the recommendation of the work of Tong et al. (2017) [17], it is possible to affirm that the mapping segmented per hour leads the results to the minimum found in the experiments. The next analysis will also be segmented per hour, with a window size of 200 instances.

Table IV shows the performance of the stream model trained for each district separately. The performance analysis can reveal how predictable the demand can be in each one, according to their characteristics. Manhattan shows to be the district with the worst predictability and, State Island, with the highest. The evaluation window with 200 examples proved to be most effective in almost all evaluations, except the Bronx. The State Island district is a great candidate for implementing a pilot project to manage and recommend taxi demand. The performance in this district reaches 89% accuracy in demand prediction, a promising result compared to Zhao, Khryashchev and Vo (2019) [6].

VI. Conclusions and Future Work

This paper proposes a methodology for modeling taxi demand prediction using stream learning. In order to evaluate our methodology, we used data from New York City collected in July 2014, available on the city's opened data platform. Even without climate data, the final stream model showed considerable performance in demand prediction and is also promising when evaluated separately by district .

As noticed in Coviensky (2017) [18], the taxi demand at one point may not portray the reality very well in some cases, as the provided data set considers as a demand the fact that the passenger has already boarded the vehicle, disregarding other people who are possibly in line, waiting for a free taxi. If there were data about the passengers waiting in line, the waiting time of passengers at one point and the waiting time for taxis stopped at the point, the model would be able to measure demand more precisely. For future work, we think that should be interesting and improve the results (i) incorporating climatic data; (ii) incorporating traffic data; (iii) using other stream learning algorithms; (iv) adapting the methodology for predicting public transportation demand. The datasets and algorithms used in this work are available at Github, https://github.com/dlfaial.

Acknowledgements

We would like to thank to FAPERJ and CNPq for financial support that helped conducting this research.

References

[1] K. Hoffmann, P. G. Ipeirotis, and A. Sundararajan, "Ridesharing and the use of public transportation," *Digital Innovation At The Crossroads*, vol. 18, no. 6, pp. 11 – 14, 2016, special issue on Transportation Simulation. Advances in Air Transportation Research.

[2] L. Rayle, S. Shaheen, N. Chan, D. Dai, and R. Cervero, "App-based, on-demand ride services: Comparing taxi and ridesourcing trips and user characteristics in san francisco," *Transportation Sustainability Research*, vol. 18, no. 6, pp. 0–18, 2014.

[11] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain," *Psychological Review*, vol. 65, no. 6, pp. 386–408, 1958.

[3] Uber, "Red line fail? ride for free on boston's uber line," 2011, available at https://www.uber.com/newsroom/red-line-fail-introducing-the-u-line/. Accessed on March, 20th, 2020.

[4] Y. Huang and J. W. Powell, "Detecting regions of disequilibrium in taxi services under uncertainty," in *Proc. 20th Int. Conf. Advances in Geographic Information Systems — SIGSPATIAL'12*. ACM, 2012, pp. 139–148.

[5] Y. W. Liu, X. Y. Zhang, Y. J. Gong, W. N. Chen, and J. Zhang, "A parallel genetic algorithm with region division strategy to solve taxi-passenger matching problem," in *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, Nov 2017, pp. 1–7.

[6] K. Zhao, D. Khryashchev, and H. Vo, "Predicting taxi and uber demand in cities: Approaching the limit of predictability," *IEEE Transactions on Knowledge and Data Engineering*, 2019.

[7] E. Frank, M. A. Hall, and I. H. Witten, *Data Mining: Practical Machine Learning Tools and Techniques*, 4th ed. Morgan Kaufmann, 2016.

[8] P. Simon, *Too Big to Ignore: The Business Case for Big Data*, 1st ed. Wiley Publishing, 2013.

[9] P. Domingos and G. Hulten, "A general framework for mining massive data streams," *Journal of Computational and Graphical Statistics*, vol. 12, no. 4, pp. 945–949, 2003.

[10] A. Bifet, R. Gavalda, G. Holmes, and B. Pfahringer, *Machine Learning for Data Streams with Practical Examples in MOA*. MIT Press, 2018.

[12] S. S. Haykin, *Neural networks and learning machines*, 3rd ed. Pearson Education, 2009.

[13] G. Widmer and M. Kubat, "Learning in the presence of concept drift and hidden contexts," *Machine Learning*, vol. 23, no. 1, pp. 69–101, Apr 1996. [Online]. Available: https://doi.org/10.1007/BF00116900

[14] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A survey on concept drift adaptation," *ACM Comput. Surv.*, vol. 46, no. 4, pp. 1–44, 2014.

[15] A. Anwar, M. Volkov, and D. Rus, "Changinow: A mobile application for efficient taxi allocation at airports," in *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, Oct 2013, pp. 694–701.

[16] C. Kamga, M. A. Yazici, and A. Singhal, "Hailing in the rain: Temporal and weather-related variations in taxi ridership and taxi demand-supply equilibrium," in *Transportation Research Board 92nd Annual Meeting*, 01 2013.

[17] Y. Tong, Y. Chen, Z. Zhou, L. Chen, J. Wang, Q. Yang, J. Ye, and W. Lv, "The simpler the better: A unified approach to predicting original taxi demands based on large-scale online platforms," in *Proc. KDD 2017*, 08 2017, pp. 1653–1662.

[18] A. Coviensky, "Estimating demand for taxis at laguardia airport," *Taxi and Limousine Comission*, pp. 1–14, 2017.

[19] NYC TLC Comission, "Taxicab passenger enhancements program (t-pep)," 2014, available at https://rules.cityofnewyork.us/content/taxicab-technology-enhancement-providers. Accessed on March, 20th, 2020.