Survival of Taxi Companies Using Machine Learning Algorithms: A Case Study in New York City

Cristhian Lizarazo clizaraz@purdue.edu

Tomas Hrdlovics thrdlovi@purdue.edu

Jin Young Son son74@purdue.edu

1 Introduction

1.1 Background

New York City (NYC) has recently tapped into an unprecedented shock where Transportation Network Companies (TNC) such as Uber or Lyft take away the majority of ridership [Jiang et al., 2019, Xu et al., 2017, Moreira-Matias et al., 2013, Zhang et al., 2016]. TNCs provide a similar taxi service via smartphone platform. User-friendly interface and remote request helped a rapid increase in the number of trips covered by these companies to approximately 3 billion in the last four years. Local governments have proposed multiple regulations to protect taxi companies and provide a more organized service from TNCs. As pointed out by [Jonas, 2015], the dangers of TNCs from operating without restrain and circumventing the existing law can be potentially harmful to unapprised users. As a response, local administration in NYC seeks to continue to cap the number of TNC vehicles aiming to fight congestion and provide higher wages. However, these regulations are commonly inefficient to protect taxi companies who struggle every day with unparalleled competition [Offenhuber and Ratti, 2014].

This issue is further emphasized with numerous attempts to build more efficient algorithms for TNC to thrive, but not so much for traditional taxi companies to counteract TNC. By having better strategies, taxi companies might have a chance at stopping monopolization of TNC. Models for solving problems such as these are often termed with time series forecasting. In this specific case, we are interested in short-term demand prediction. Forecasting for time series is known to be a very difficult task, but with blooming advance of machine learning techniques, many models have been proposed to come up with better solutions, allowing us to approach our problem from various directions.

Open data-sources of trips conducted by citizens such as the one provided by the New York City Open Data Project provides an invaluable source of information to estimate short-term forecast models and implement a myriad of machine learning models to improve coverage of taxi services. This allows keeping the valuable heritage of the Yellow taxis in New York. Through history, taxis have served in NYC not only as mode of transportation, but also they are a cultural icon of the city. As cited from [Deri and Moura, 2015] *"San Francisco has its trolley, London has its "tube", and New York City is well known for its yellow-checkered taxi cabs"*.

1.2 Problem Statement

Conventional taxi service in New York City is highly inefficient reflected on a considerable amount of time with empty trips conducted by drivers. [Offenhuber and Ratti, 2014] In addition, user-friendly interface and remote access from TNC companies such as Uber or Lyft burden taxi companies towards surviving in a market with more connected users. We would like to spin the problem towards a more applied issue in better preparing taxi companies in this evolving market. The objective of this project is to explore and understand various short-term forecast models, and implement them in our data, in an attempt to provide a better distribution of taxi vehicles, thus finding strategies for taxi companies to survive in the overflow of TNC. Users might be able to shift towards taxi companies if waiting times are lower and better service is provided [Dudley et al., 2017]. The assumption of this research proposal is that by modeling TNC's short-term demand distribution, we can achieve better distribution of taxi vehicles. If Taxi companies are able to dispatch more vehicles on those regions where short-term TNC demand is high, they will more likely achieve a modal shift towards their service.

The stated problem can be seen with the analogy of dispatching police officers to locations where short-term forecast of crime is high. Predictive models are trained based on crime reports and predicts locations with high chance of crime occurrence and the type of crime to happen (i.e. burglary, assault,...). These models improve crime detection up to 50% depending on the city [Smith, 2018]. In analogy to our project, crime scene is the place where taxi driver pick ups the customer and the type of crime is the type of taxi. This makes models predicting event locations based on data high in value.

2 Literature Review

Interest for short-term demand forecast models has emerged in recent years duo to a exponential growth on the number of methodologies to apply and data from taxis and TNC companies. Forecasting has been a widely discussed topic across multiple generations better represented by the M4 competition. This competition has been around for more than 45 years with the primary objective to learn how to improve forecasting accuracy Indeed, primary findings of the 2018 M4 competition suggests confirmation of the superiority of ML algorithms in forecasting. [Makridakis et al., 2018]

Short-term forecast demand for taxis can be implemented using parametric and non-parametric methods. Among the most widely applied statistical approaches reported in the literature include Moving Average (MA) [Xu et al., 2017], Exponential Smoothing (ES) [Davis et al., 2018] and Autoregressive Integrated Moving Average (ARIMA) [Li et al., 2012]. These parametric models have yielded good results for simpler, and more linear problems. In our case, we will first approach our problem with ES, as the algorithm utilizes exponentially decaying weights to fit the time series data.

There have been a numerous of non-parametric machine learning algorithms that were implemented to solve short-term demand prediction tasks as well. The primary advantage of these non-parametric models is the ability to provide a better characterization of non-linearity and complexity of traffic events. The objective of non-parametric methods include identifying historical data similar to the instant to be forecasted. Identification of epochs with similar characteristics can be implemented using typical machine learning methods including neuronal networks, decision tree models among others [Mukai and Yoden, 2012]. More advanced deep learning models include LS-SVM approaches [Jiang et al., 2019], LSTM Layers [Zhao et al., 2016], convolutional operators [Ke et al., 2017] and Deep Multi-View Spatial-Temporal Network [Yao et al., 2018].

Of them all, the research team decided to apply LS-SVM based on the new concepts learned in class and Long Short-Term Memory (LSTM) network and its variants. LS-SVM provides a good approaximation for They have been gaining popularity for their use in time series analysis, due to their modules' ability to "remember" and "forget" past observations. Indeed, LSTM-based architectures seem to outperform most of the algorithms proposed in the M3 competition. [Laptev et al., 2017]

3 Data

3.1 Data Acquisition

Data is obtained from the New York City Taxi and Limousine Commission (NYCTLC) [Donovan and Work, 2014]. The data includes information of individual trips done by New Yorkers in the month of June 2018. There are three different datasets: Green Taxi (GTX), Yellow Taxi (YTX), and Transportation Network Companies (TNC). The three datasets offer a similar set of features including:

1. Pickup location and time
2. Drop-off location and time
3. Duration of the trip

GTX and YTX datasets provide three additional features including distance covered during the trip, fare rates, and number of passengers.

There are 265 regions that are used for reporting the pickup and drop-off locations as a substitute for latitude and longitude coordinates. This level of aggregation is applied by the New York City Open Data Project to protect privacy of users utilizing these transportation modes.

Additional features that can be used in these prediction models include land use of these regions and historical weather information. Land use information can be retrieved from the New York City Department of City Planning (NYCDCP) department. This data set provides information of land use at the block level categorizing it within residential, commercial, industrial areas among others. In regards to weather data, the National Oceanic and Atmospheric Administration database can be used for this purpose [Rutledge et al., 2006].

Some possible limitations of the data include the lack of sociodemographic information of passengers. Also, the discrepancy between the number of features of GTX, YTX, and TNC also could limit the dimensions for analysis.

3.2 Data Preprocessing

Due to the large number of observations available in the NYCTLC dataset, pre-processing was required to remove possible outliers and low reliable information. A total of 26,786,664 trips were obtained for further analysis. Two data preprocessing steps were performed:

1. First, using QGIS Software, the land use data and weather was matched into the 265 taxi zones in the NYCTLC data. Land use data, which is available by lot, was aggregated to the area and weighted based on its surface to obtain the predominant land use for each taxi zone.

2. Second, the identification of possible outliers was obtained by looking at the distributions of individual variables. We could identify some evident outliers from the trip distance and trip fare. Nonetheless, the behavior of trip duration seems to be consistent with an almost gamma-like shape. In the case of trip duration, we concluded that a trip longer than 3 hours (10,800 seconds) will be excluded from the analysis. These trips accounted for 0.1% of the data.

The spatial distribution of TNC and yellow trips are shown in Figure 1 and Figure 2. Based on these results it is possible to visualize similar patterns across these to transportation modes. In general, taxi services tend to have a more concentrated number of trips across Manhattan as compared to TNC services. The following project provides forecasting of the total number of trips implemented by these two transportation modes.

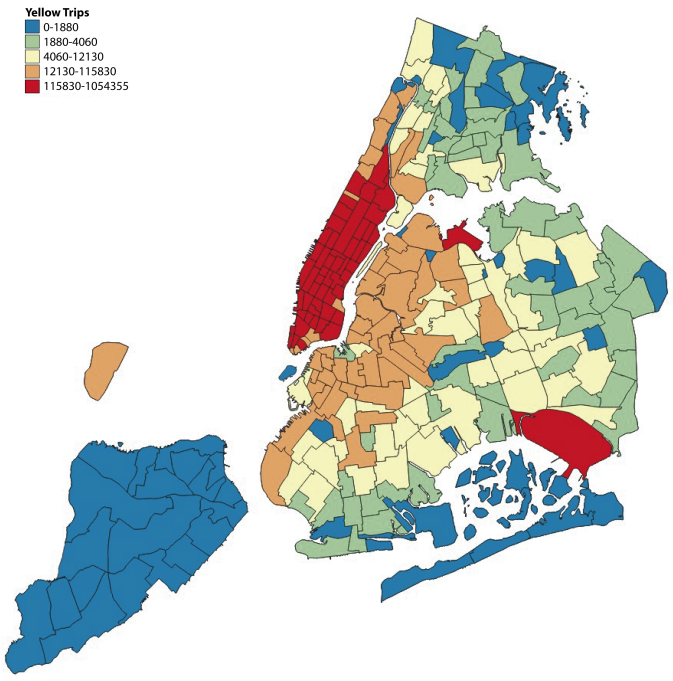

Figure 1: Spatial distribution of Yellow Taxi Trips in NYCα, γ and δ

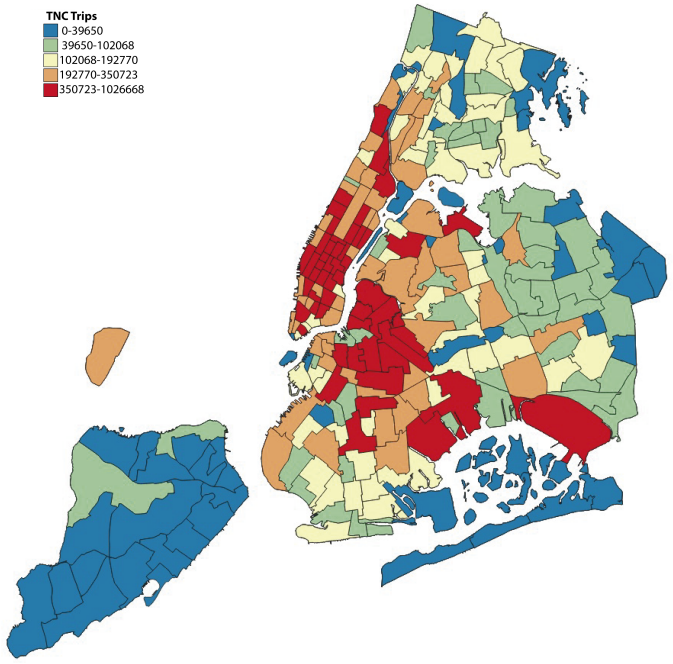

Figure 2: Spatial distribution of TNC Trips in NYC α, γ and δ

4 METHODOLOGY

4.1 MODEL FORMULATION

To familiarize the reader with transportation terminology, demand forecast is defined as estimation of the number of vehicles needed on a given future time frame for a given region. Short term forecast usually refers to prediction within a time frame of one hour. We also define:

1. Region i: Zones with similar characteristics in terms of travel dynamics
2. Timeframe t: The time window of a constant length
3. Demand $d(t, i)$: The sum of starting trips in a specific time interval t in region i

Using a similar notation proposed in [Jiang et al., 2019], the taxi demand of time t and region i can be recorded as:

$$d(t, i) \text{ where } t = 1, 2, ..T \text{ and } i = 1, 2, ..., I \tag{1}$$

Where t is the time index, i is the number of traffic regions, T is the total number of time intervals, and I is the total number of traffic regions. Traffic forecasting determines $d(t, i)$ using the historical data, $d_1, d_2, \ldots d_{t-1}$. Hence, the function to predict can be defined as follows:

$$d(t, i) = f(d(t-1, i), d(t-2, i), ..., d(t-K, i) \tag{2}$$

Where f is the model estimated for all regions through the training process. K is defined as the historical time step. It controls the length of past data to be used in prediction.
An alternative formulation can be proposed by including a single model f_i per region to better characterize intrinsic features for these zones. In this case, the expression (2) can be defined as:

$$d(t, i) = f_i(d(t-1, i), d(t-2, i), ..., d(t-K, i) \tag{3}$$

Where f_i is the model estimated for each one of the regions through the training process. To construct the training set for specific region i, the output vector of the training set is defined as:

$$Y = [d(1+K, i), d(2+K, i), , d(N, i)]^T \tag{4}$$

Being the vector Y a column vector of $(N-K, 1)$ dimensions. Hence, it is possible to specify the input vector for the training as:

$$D = \begin{bmatrix} d_{(1,i)} & ... & d_{(k,i)} \\ ... & ... & ... \\ d_{(N-K,i)} & ... & d_{(N-1,i)} \end{bmatrix} \tag{5}$$

Where $N-K$ is the total number of training set time intervals. We can use training sets D and Y to find the parameters of the model f_i.

4.2 Model Training Validation and Testing Criteria

In order to avoid temporal dependencies and simulate real world forecasting environment based on data in the past, the most suitable evaluation model seems to be Cross Validation using time forward chaining pattern [Cochrane, 2018]. Data in each round are split into training and validation set. The validation score is the average error over all rounds. After tuning the hyperparameters, the overall performance of the model will be evaluated based on the newest data, which were not used for tuning purposes. The general cross-validation process is depicted as follows:

Training-Validation

- Fold-1: Training [June 1st-June 7th] Validation [June 8th]
- Fold-2: Training [June 1st-June 8th] Validation [June 9th]
- Fold-3: Training [June 1st-June 9th] Validation [June 10th]
  ...
- Fold-k: Training [June 1st-June 20th] Validation [June 21st]

Testing

- Testing Data: Testing [June 22nd-June 30th]

Based on this process time forward chaining pattern is applied adding a new day for training on each round. Training and validation is implemented the first 21 days of June and testin is appied in the remaining days.

5 Exponential Smoothing

Exponential smoothing describes a class of time series model. The method was first conceived by Robert G. Brown in 1944 in the US Navy Operations. The idea in this concept was finding a mechanical computing device for tracking the velocity and angles applied for submarines [Gardner, 2006]. Based on [Hyndman et al., 2008], time series problems can be decomposed into trend (T), seasonal (S) and irregular or error (E) components. Definition of these components are provided as follows:

1. Trend T: Long-term direction of the prediction
2. Seasonal S: Patterns that repeats with certain periodicity.
3. Error or noise E: Components that cannot be predicted using the time series model.

A general classification of exponential smoothing methods is provided based on estimation of these components: (1) Simple exponential smoothing (SES) characterizes error and noise and (2) Holt-Winters seasonal method (HWSM) characterizes trends and seasons.

5.1 SIMPLE EXPONENTIAL SMOOTHING (SES)

Let us consider the demand for Taxi and TNC services for region i in time stamp t denoted as $d_{t,i}$. Aggregation is applied in 30 minutes intervals. The method of simple exponential smoothing takes the forecast of the previous period and adjusts it using the forecast error characterized on the expression :

$$\hat{d}_{t+1,i} = \hat{d}_{t,i} + \alpha[(d_{t,i} - \hat{d}_{t,i}] \tag{6}$$

Being α a parameter between 0 and 1. In SES the new forecast is the addition of the old forecast plus an adjustment for the error that occurred in the last forecast. This equation can be also expressed as:

$$\hat{d}_{t+1,i} = \alpha d_{t,i} + (1-\alpha)\hat{d}_{t,i} \tag{7}$$

When considering the additional historic time steps for prediction K, the formulation can be expanded as:

$$\begin{aligned}\hat{d}_{t+1,i} = \alpha d_{t,i} + \alpha(1-\alpha)d_{t-1,i} + \alpha(1-\alpha)^2 d_{t-2,i} + \alpha(1-\alpha)^3 d_{t-3,i} + \alpha(1-\alpha)^4 \\ d_{t-4,i} + ... + \alpha(1-\alpha)^k d_{t-k,i}\end{aligned} \tag{8}$$

Thus, the prediction for demand at time $\hat{d}_{t+1,i}$ is expressed as the weighted average of the past observations with the weights decreasing exponentially. Let us define $l_t = \hat{d}_{t+1,i}$. Hence:

$$l_{t,i} = \alpha d_{t,i} + (1-\alpha)l_{t-1,i} \tag{9}$$

In this case $l_{t,i}$ is a measurement of the level of the times series at time t in zone i. This representation allows generalizing the exponential model to allow for seasonality.

5.2 DOUBLE SEASONALITY HOLT-WINTERS METHOD (DSHW)

Figure 3 shows the total demand for taxi and TNC services for June 2018 in New York City. Aggregation is applied using 30 minutes intervals. The results show a clear pattern and repetitive behavior per day and week. The peak hours are shown on weekdays reflected on commuting patterns. On the other hand, Sundays tend to have lower demand as compared to the other days. The traditional Holt-Winters method (HWSM) is suitable for time series with one seasonal pattern. The governing equations for the single season HWSM are defined as:

$$\begin{aligned} Level &: l_{t,i} = \alpha(d_{t,i} - s_{t-m}) + (1-\alpha)(l_{t-1,i} + b_{t-1,i}) \\ Growth &: b_{t,i} = \beta(l_{t,i} - l_{t-1,i}) + (1-\beta)b_{t-1,i} \\ Seasonal &: s_{t,i} = \gamma(d_{t,i} - l_{t-1,i} + b_{t-1,i}) + (1-\gamma)s_{t-m,i} \\ Forecast &: \hat{d}_{t+h,i|t,i} = l_{t,i} + b_{t,i}h + s_{t-m+h_m} \end{aligned} \tag{10}$$

Where $s_{t,i}$ is the seasonal component, $l_{t,i}$ is the estimate of the level at time t, $b_{t,i}$ is the estimate of the growth at time t and h_m is [(h-1) mod m]. The parameters (α, β, γ) are restricted between 0 and 1.

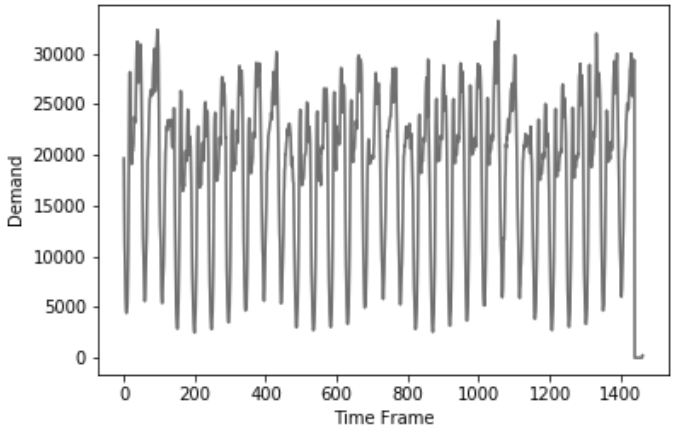

Figure 3: Seasonality TNC and Taxi Demand

Following the method applied in [Taylor, 2003] , the formulation of Double Seasonality Holt-Winters method (DSHW) that can accommodate more than one seasonal pattern follows:

$$\begin{aligned}
Level : l_{t,i} &= \alpha(d_{t,i} - s^1_{t-m1,i} - s^2_{t-m2,i}) + (1-\alpha)(l_{t-1,i} + b_{t-1,i}) \\
Growth : b_{t,i} &= \beta(l_{t,i} - l_{t-1,i}) + (1-\beta)b_{t-1,i} \\
Season\ 1 : s^1_{t,i} &= \gamma(d_{t,i} - l_{t,i} - s^2_{t-m2}) + (1-\gamma)s^1_{t-1,i} \\
Season\ 2 : s^2_{t,i} &= \delta(d_{t,i} - l_{t,i} - s^1_{t-m1}) + (1-\delta)s^2_{t-1,i} \\
Forecast : \hat{d}_{t+h,i|t,i} &= l_{t,i} + b_{t,i}h + s^1_{t-m1+h} + s^2_{t-m2+h}
\end{aligned} \tag{11}$$

Based on Figure 3, two seasonalities are included in the model in regards to the repetitive behavior of travel demand in a daily and weekly basis.

6 Least Squares Support Vector Machine (LS-SVM)

Using the concepts introduced in the class, LS-SVM provides multiple advantages and good prediction for demand estimation [Jiang et al., 2019]. The framework introduced in class provides theoretical background for supervised learning models applied into classification. A regression problem is introduced in the following report to extend the applications for SVM. The difference between applying the standard SVM and Least Squares SVM relies on changing the inequality constraint in the original method with an equality constraint, helping in the estimation of the

Lagrange Multiplier. In this section, the primal and dual optimization problems are provided to better guide implementation of the model . In the feature space, the model LS-SVM is characterized based on the following expression:

$$y = \omega^T \chi(d) + b \tag{12}$$

where ω is the model coefficient and $\chi(.)$ is a column vector characterizing the nonlinear mapping function. Let us define the input and output vectors:

$$d = [d_{t-1,i}, d_{t-2,i}, ..., d_{t-K,i}], y = d_{t,i} \tag{13}$$

Being $d \in R^K$ as input vector and $d_{t,i} \in R$ as output. Given training sets D and Y, the corresponding optimization problem of LS-SVM parameter estimation is as follows:

$$\min J(\omega, e) = \frac{1}{2}\omega^T\omega + \gamma\frac{1}{2}\sum_{k=1}^{N-K} e_k^2 \tag{14}$$

The equality constraints of the optimization problems are:

$$y_k = \omega^T \chi(d_k) + b + e_k, k = 1, ..., N-K \tag{15}$$

Where

$$\begin{aligned} y_k &= d(k+K, i), \\ d_k &= [d(k,i), d(k+1,i), ..., d(k+K-1,i)] \end{aligned} \tag{16}$$

The Lagrange dual problem is specified as:

$$L(\omega, b, e, \alpha) = J(\omega, e) - \sum_{k=1}^{N-K} \alpha_k \omega^T \phi(d_k) + b + e_k + y - k \tag{17}$$

The optimization conditions are as follows:

$$\begin{aligned} &\frac{\delta L}{\delta \omega} = 0 \implies \omega = \sum_{k=1}^{N-K} \alpha_k \phi(d_k) \\ &\frac{\delta L}{\delta b} = 0 \implies \omega = \sum_{k=1}^{N-K} \alpha_k = 0 \\ &\frac{\delta L}{\delta e_k} = 0 \implies \alpha = \gamma e_k, k = 1, ..., N-K \\ &\frac{\delta L}{\delta \alpha_k} = 0 \implies \omega^T \phi(d_k) + b + e_k - y_k = 0, k = 1, ..., N-K \end{aligned} \tag{18}$$

Hence, the following linear equations can be solved:

$$\begin{bmatrix} 0 & \vec{1} \\ \vec{1}^T & \Omega + \gamma^{-1} I \end{bmatrix} \begin{bmatrix} b \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ y \end{bmatrix} \tag{19}$$

Where $y = [y_1, ..., y_{N-k}]^T, \vec{1} = [1, ..., 1]$, and $\alpha = [\alpha_1, ..., \alpha_{N-K}]^T$. By applying Mercer conditions:

$$\Omega_{kl} = \phi(d_k)^T \phi(d_l) = K(d_k, d_l), \qquad k, l = 1, ..., N - K \tag{20}$$

The training process of the model involves estimation of the dual optimization problem. The parameters are obtained by solving the linear equations specified in (19). After finding parameters α and β in the optimization model, given a new data point, the model estimate prediction as:

$$y = \sum_{k=1}^{N-K} \alpha_k K(d_k, d) + b \tag{21}$$

6.1 KERNEL FUNCTION

The kernel function is specified as the function k use to map the non-linear input feature space into a high dimensional linear space that better resembles the non-linear function. Among the most applied nonlinear kernel functions, the research team found the polynomial kernel function, the radial basis kernel function, and the sigmoid kernel function. This report adopts the Radial Basis kernel function (RBF) characterized in the expression:

$$K(x_i, x_j) = \exp\left(\frac{-||x_i - x_j||^2}{(2\sigma^2)}\right) \tag{22}$$

In order to obtain the LS-SVM model with RBF kernel, an addition two extra tuning parameters are required, including the regularization parameter γ defining the trade-off between the training error minimization and smoothness of the function. In addition, σ^2 (sig2) is the Kernel function parameter. In this case, nested cross-validation is applied for estimation of these parameters as explained in the previous section.

7 LONG-SHORT TERM MEMORY (LSTM) - RECURRENT NEURAL NETWORK (RNN)

Recurrent neural network, and especially LSTM, are among the most well-known models to process sequential data. The definition of RNNs as recurrent is based on the process of performing

the same computation on every element of a sequence, with the output conditioned on previous computations. RNN uses input x, stores hidden state h and outputs d at each time-step t. Estimation of the weights can be obtained by unrolling the network for a finite number of steps as shown in Figure 4

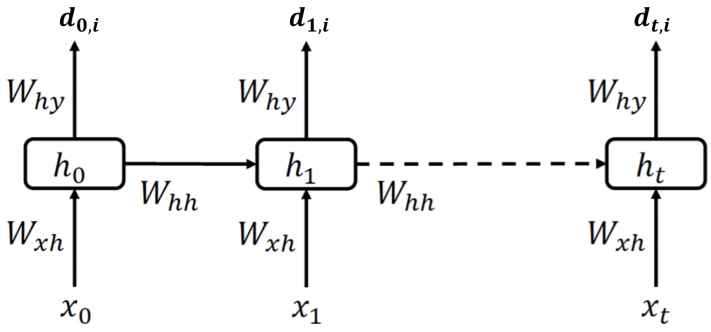

Figure 4: Recurrent Neural Network α

Based on Figure 4, the computational elements applied on each step can be described as follows:

1. x_t is the input at time-step t,
2. h_t is the non-linear activation function at time-step t. The non-linear activation function in this case is usually a hyperbolic tangent: $h_t = tanh(W_{xh}x_t + W_{hh}h_{t-1} + b_h)$,
3. d_t is the output at time step t.

LSTM, a derivation of RNN, is applied in this case considering its capability of learning long-term dependencies due to its gathering mechanism. This means that LSTM cell remembers its previous status from previous time observation, selectively forgets part of the past, and updates its current status given cell state and the input. This helps to forget information which is less important and remember information with high importance even over longer period of time. That information for instance might be the demand in a zone 24 hour ago [Gers et al., 1999].

For training purposes, the model is trained with normalized data using min/max scaling in a range of $[-1, 1]$. This is done because the time zone demands are non-linear in nature and highly dynamic in behaviour. Studies show that the Neural Networks adapt for such data when scaled [Wittman, 2002].

8 Model Evaluation and Results

Real demand values from multiple zones are shown in Figure 5. Based on Figure 5, there is a considerable variation of demand depicted across zones. There are zones with high demand for

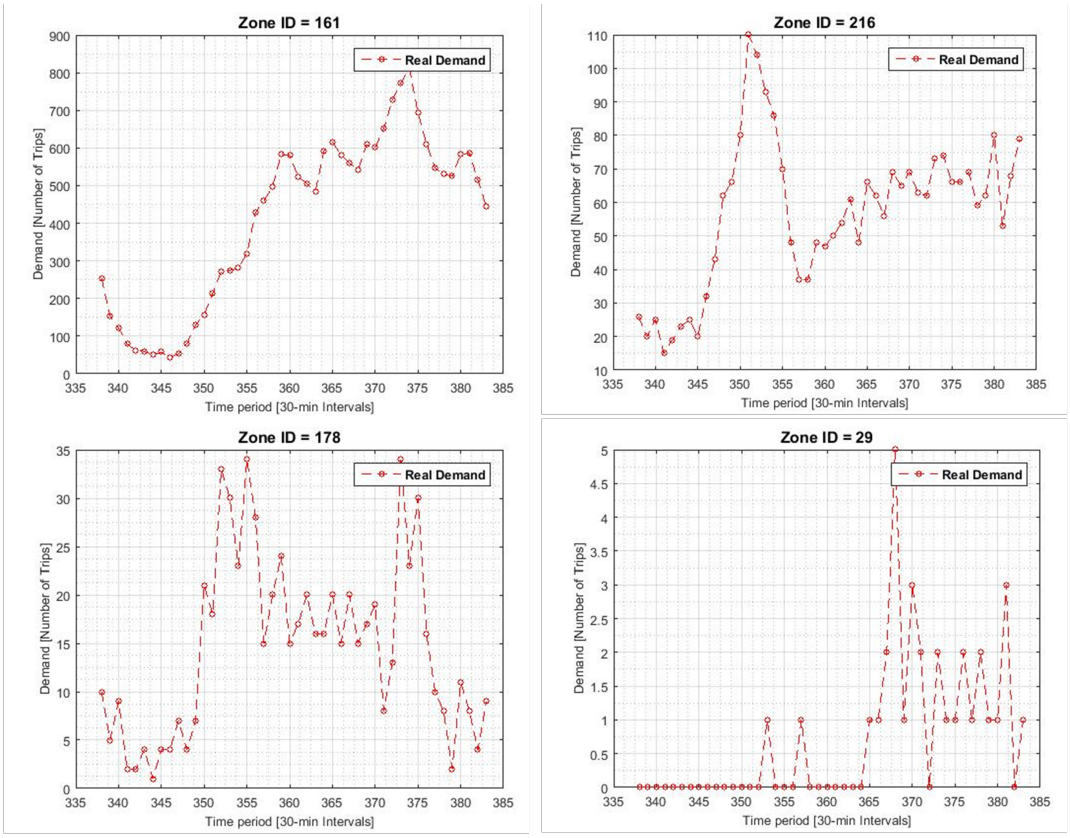

Figure 5: Real demand in multiple zones June 7th, 2018 α, γ and δ

taxi and TNC services with up to 1000 trips requests in a 30-minute time interval. On the other hand, additional zones can refer low or even null values in the number of trips. We will refer to these zones to see how well our models perform.

8.1 Exponential Smoothing (ES)

8.1.1 Simple Exponential Smoothing (SES)

Based on the expression (8), it is important to estimate the parameter α associated to the decaying function. There are two different approaches for estimation of this parameter. The first approach includes estimation of individuals α_i per region. However, it is desirable to have a simpler, a more general model for the entire sample of zones. Therefore, we estimated a global α that yields the lowest mean squared error (MSE) across all districts.

The estimation of the parameter α was obtained via cross-validation. Daily observations were trained for forecasting of time periods $d_{t+2,i}$ and $d_{t+3,i}$ representing 30 minutes and 1 hour

Table 1: Parameter Definition

Method	Definition of Parameters
Simple Exponential Smoothing	'Parameter Alpha'=[0,1]; 'Parameter Training Window'=48 (24 Hours); 'Parameter Forecast Window'=3 (90 minutes after training - 30 mins time period)
Double Seasonality Holt-Winters Method	'Parameter Alpha'=[0,1]; 'Parameter Beta'=0 (No growth); 'Parameter Gamma'=[0,1]; 'Parameter Delta'=[0,1]; 'Parameter Training Window'=672 (2 Weeks); 'Parameter Forecast Window'=3 (90 minutes after training - 30 mins time period); 'Parameter Window Season 1'=48 (24 hours); 'Parameter Window Season 2'=336 (1 week);

after the model training. The prediction was applied for these periods considering the objective towards dispatching taxis in high demand zones. Applying $t+2$ and $t+3$ time instants allow a time window of 30 minutes for taxi drivers to drive to these specific high demand zones. The Mean Squared Error is reported as the average of these two values as follows:

$$MSE = \frac{1}{T*I}\sum_{i=1}^{I}\sum_{t=48}^{T}[\frac{\hat{d}_{t+2,i}+\hat{d}_{t+3,i}}{2} - \frac{d_{t+2,i}+d_{t+3,i}}{2}]^2 \tag{23}$$

Where I are all the zones and T are all the time periods for prediction after 24 hours for training. Model parameters are shown in Table 1. Selection of the proper parameter α is applied via cross validation. Figure 6 shows sensitivity of the MSE with respect to the range of α from 0.1 to 1. The algorithm supports the minimum MSE when the parameter of α is equal to 1. These results are not encouraging considering the high value of MSE as well as the result of finding an optimal α in a frontier point. A plausible explanation includes regions with significant number of zeros in which SES cannot characterize their trends.

8.1.2 Double Seasonality Holt-Winters Method (DSHW)

The estimation of the DSHW was implemented using the parameters shown in Table 1. The estimation includes cross validation for three parameters including α, γ, and δ. Parameter β was set equal to zero since prediction is implemented in a time frame of one month. In general, it is expected a significant growth in demand in longer periods spanning forecasts within years. Figure 7 provides the results for MSE with respect to parameters α, γ and δ. Parameter α ranges from [0.5,0.85] while the ranges for parameters γ and δ are within [0.05,0.9]

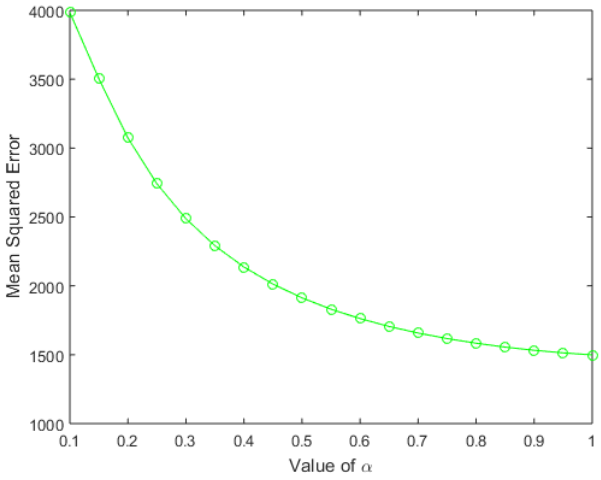

Figure 6: Sensitivity of MSE w.r.t. parameter α

The training results for the Exponential Smoothing with Double Seasonality are encouraging. The observed MSE is significantly lower in comparison to the obtained one from the SES method. The minimum MSE obtained from training data in this case was reduced from 1498 using the SES to 513 when parameters α, γ and δ were set at 0.55, 0.1, and 0.1, respectively. In addition, it is observed an decrease in the obtained error when values γ and δ are closer to zero.

The testing error is measured using $\alpha = 0.55$, $\gamma = 0.1$, and $\delta = 0.1$, as we have obtained from our training. The measured MSE for all regions is 688. Figure 8 represents our predictions with DSHW on few regions. Although it is not perfectly accurate, it does capture the seasonality of the data, and shows reasonable predictions.

8.2 Least Squares-Support Vector Machine (LS-SVM)

The LS-SVM model is evaluated using the same performance metrics specified for exponential smoothing. Nested cross-validation was applied for estimation of the three SVM parameters γ, σ^2 and the time window k. Grid search method was conducted for estimation of the parameters. The results in [Jiang et al., 2019] obtained the values of $\gamma = 10$ and $\sigma^2 = 1$ as the ones minimizing cross-validation MSE in Shanghai, China. Using these values as reference, the ranges for parameters γ and σ^2 were conducted within the ranges [2, 20] and [1,10], respectively. The discrepancy across demands of regions and high errors reported in the DSHW method provides support arguments towards estimation of regional models. Hence, individual SVM models per region are estimated using the same hyper-parameters for testing MSE.

The definition of a homogeneous time window k across all regions is desirable for consistency while structuring the data. Parameter k should not be too long to make sure training and validation can be implemented in real time. The range included in this project span the values of k between 1 to 24. It means that approximately 12 hours of observations will be utilized to

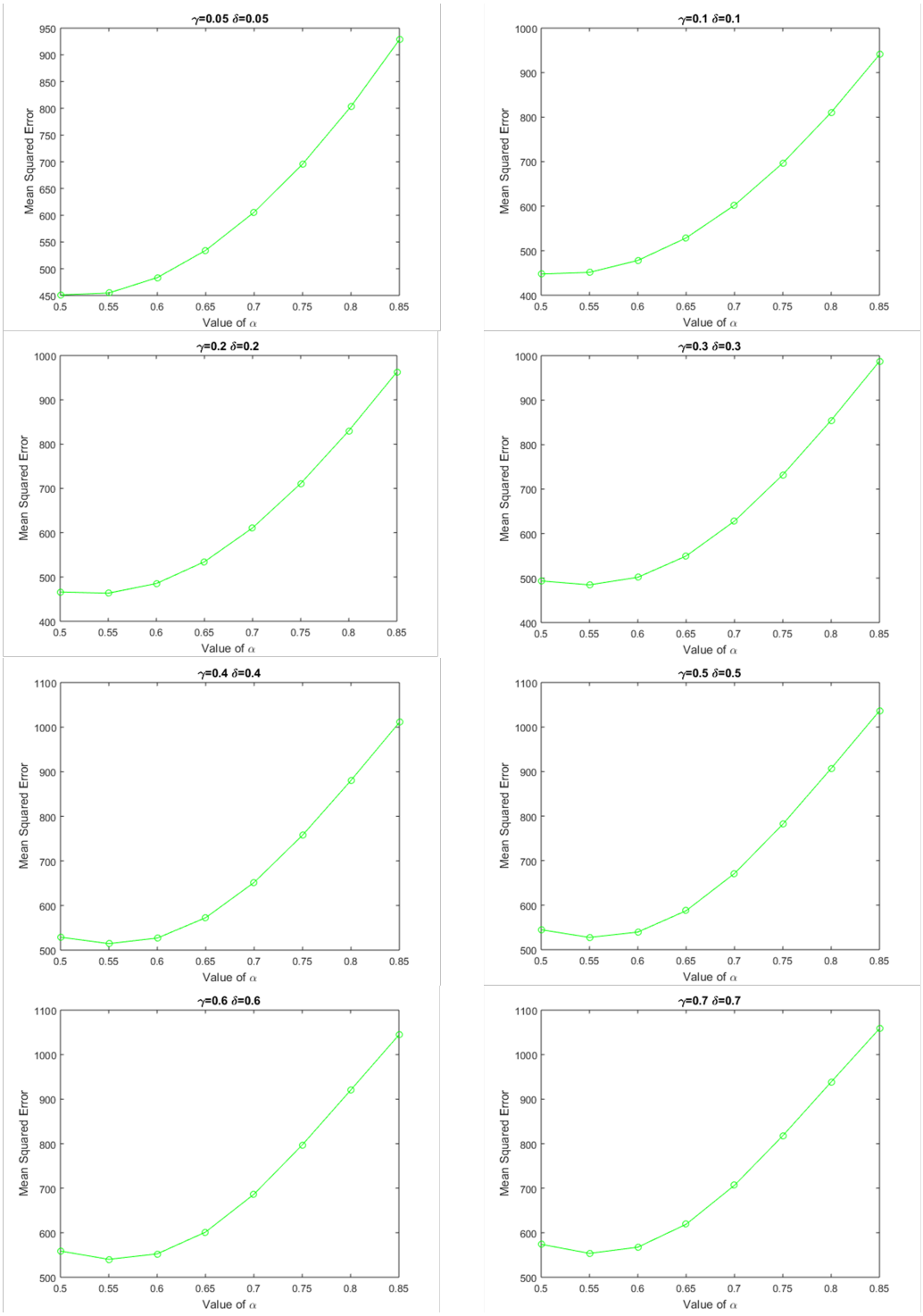

Figure 7: Sensitivity of MSE w.r.t. parameters α, γ and δ

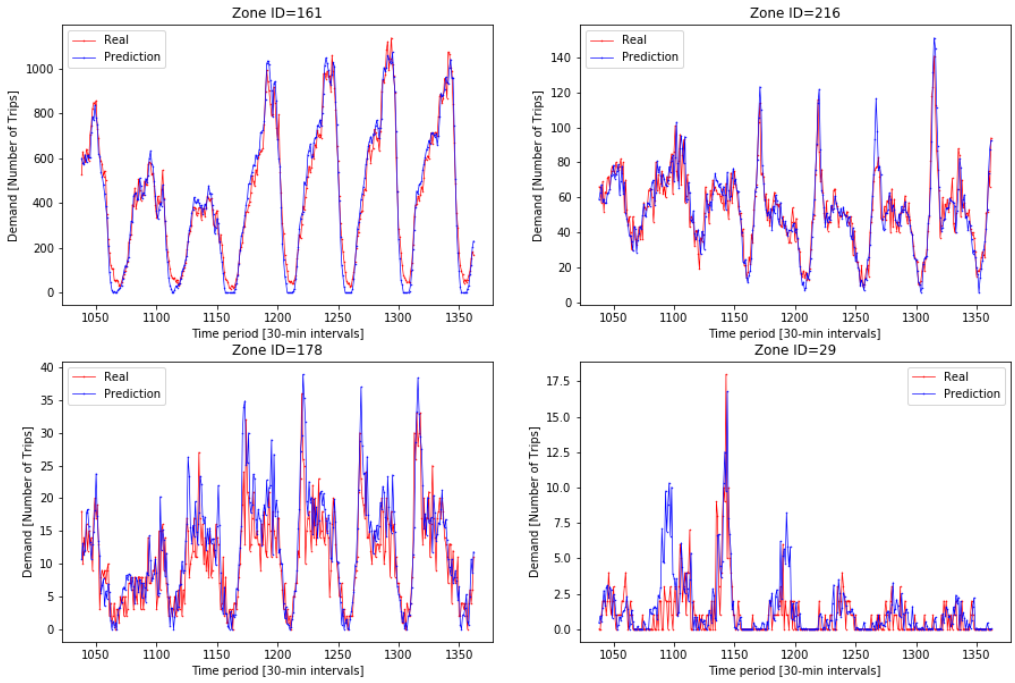

Figure 8: Real vs predicted Demand Testing June 21st to June 30th. Multiple Zones

forecast the time period 30 to 90 minutes ahead.

The results for training and validation MSE across multiple values for γ, σ^2 and k are shown in Figure 9 and Figure 10. The results for training MSE behave as expected. Greater values of γ provide lower training MSE since it penalizes for errors committed in the training set. In addition, longer time windows reflected in higher k also provides lower error considering that more information is included for training. In regards to parameter σ^2, the training error supports the value of $\sigma^2 = 1$ as the one providing the lowest training MSE.

The results for validation MSE in figure 10 show a different behaviour as compared to training MSE. In this case the results show a reduction in the obtained error in the range of γ between [2,6] and then an increasing trend in the range [6,20]. Hence, the adopted value for parameter γ is 6. The results for validation MSE w.r.t. parameter k support lower MSE values when increasing the number of periods. The figure also shows an stabilization of validation MSE after $k = 10$ with an increase pattern after $k = 20$. Based on these results,$k = 18$ is adopted as the parameter that minimizing cross-validation error. Finally, the results showed the lowest validation MSE w.r.t. to σ^2 when $\sigma^2 = 8$.

Testing error is estimated using parameters $\gamma = 6$, $\sigma^2 = 8$, and $k = 18$. Figure 11 provides a graphical comparison of real vs predicted demand for the same regions introduced in Figure 5.

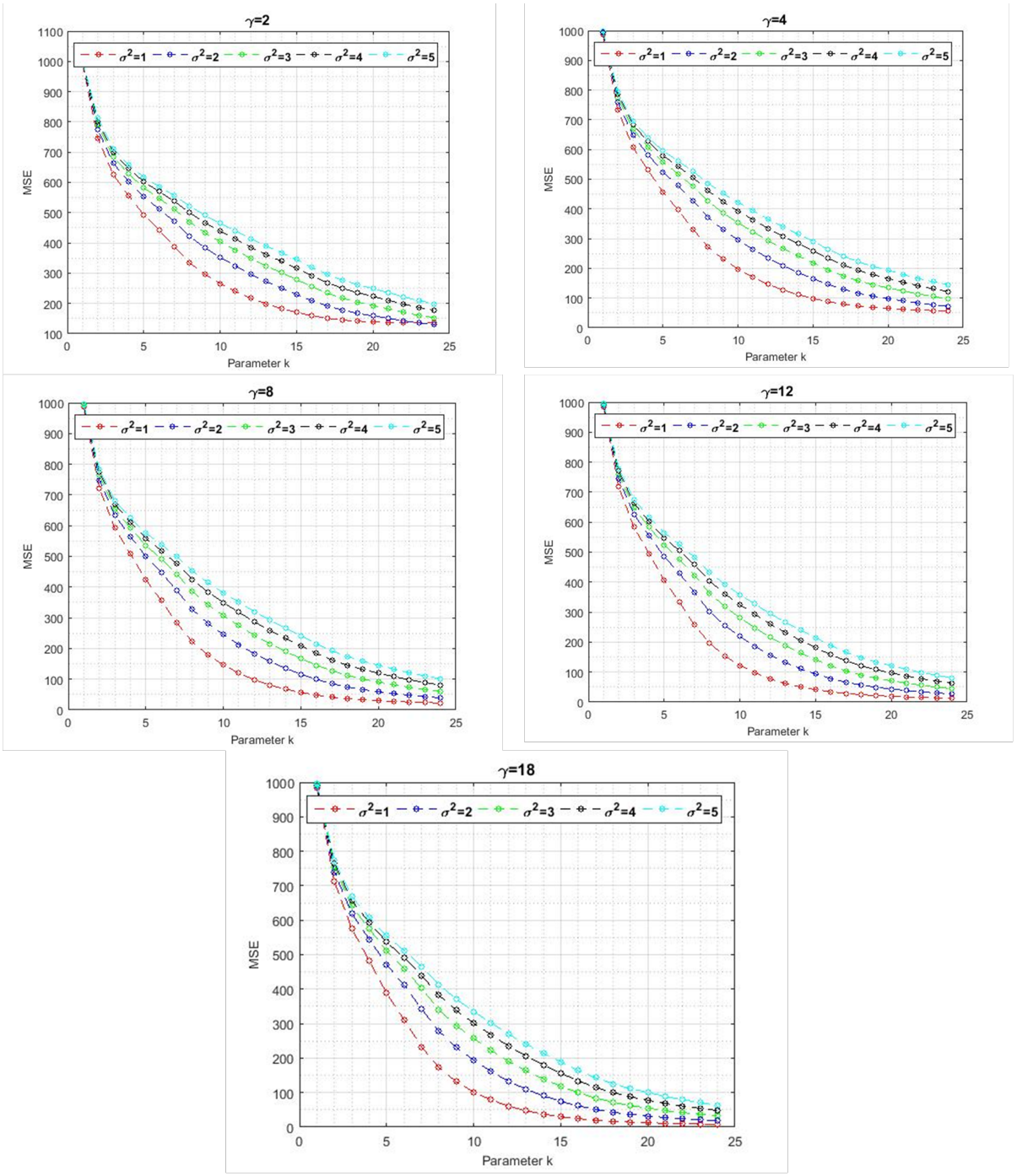

Figure 9: Training MSE w.r.t. to parameters γ,σ^2 and k

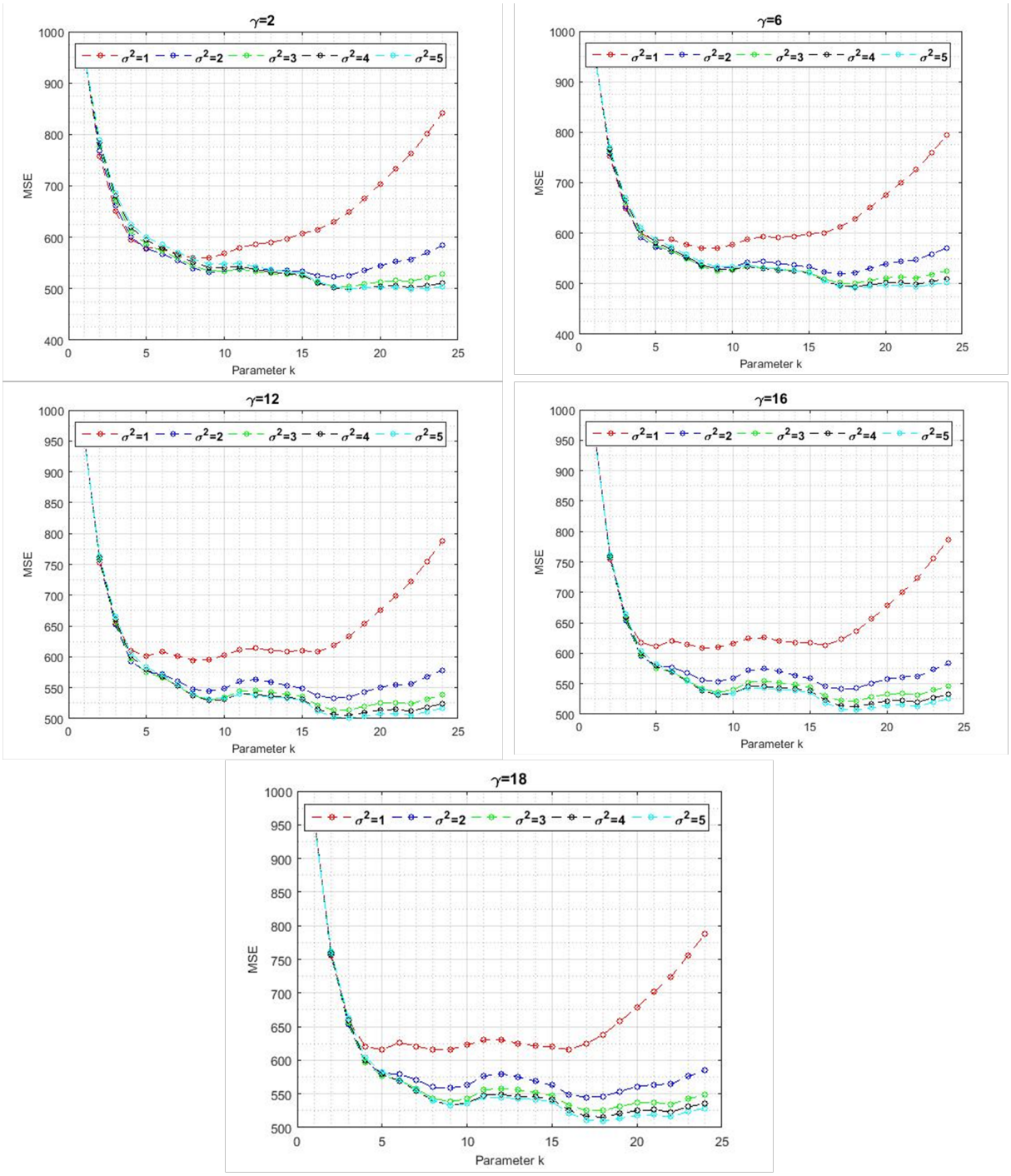

Figure 10: Validation MSE w.r.t. to parameters γ,σ^2 and k

The results are encouraging. The results follow a similar trend as the real one with challenging conditions on those regions with a low number of trips. The final testing MSE across all regions was 465.

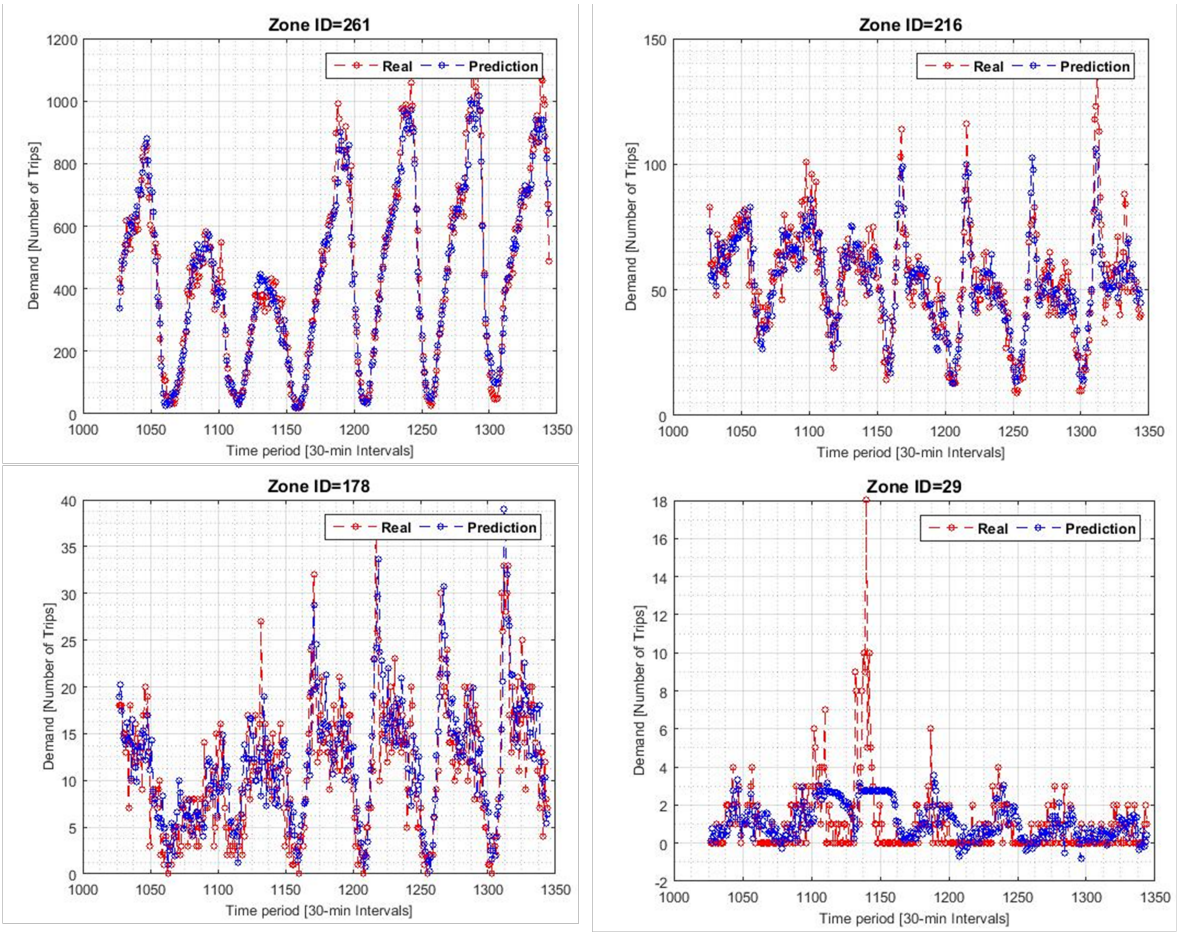

Figure 11: Real vs predicted Demand Testing June 21st to June 30th. Multiple Zones

Table 2: Parameter Definition

Method	Definition of Parameters
LSTM	'Hidden Layer Size'=[50,125]; 'Parameter Training Window'=[30,60] (15 – 30 hours); 'Parameter Forecast Window'=2 (60 minutes after training - 30 mins time period); 'Learning Rate'=[0.001,0.0001]; 'Number of Learning Epochs'=[70,150]

8.3 Long Short Term Memory Neural Network (LSTM)

The model is trained individually for each zone with the training, validation and testing set split as discussed in chapter (4.2) with the exception that instead of taking into account all previous data, only a sliding window is set (i.e. in order to make a prediction only last k results are taken into account).

The LSTM neural network itself is built using PyTorch library. The model consists of one hidden layer consisting of x neurons and a linear layer that makes the final prediction. Some experiments were done with deep neural network of two or more layers. However, not having the abundance of training data proved to be fatal for such direction in training.

Cross-Entropy loss function is used for training the model as the objective is to minimize the difference between model's predicted probability distribution given the dataset and distribution of training dataset's probabilities.

As optimizer function, Adam optimizer is used. This algorithm stands for adaptive moment estimation, which unlike stochastic gradient descent maintains a per-parameter learning rate that are adapted based on the average of recent magnitudes of gradients.

Aside from this setting, several hyper-parameters were tested in order to find the best combination that on average works the best for all zones. As a simplification we select few zones as representatives of high, medium and low demand zones. This is done in same fashion as with LS-SVM (chapter 8.2)

The values of hyper-parameters were tuned one by one. This means that once optimal value for one parameter was found, it was used during tuning of other parameters. Initial setup for LSTM is 'Hidden Layer Size' = 100 'Parameter Training Window' = 48, 'Learning Rate' = 0.0005, 'Number of Learning Epochs' = 100

After several trials and averaging error between zones the best validation results were achieved with following parameters: 'Hidden Layer Size' = 90 'Parameter Training Window' = 48, 'Learning Rate' = 0.0004, 'Number of Learning Epochs' = 90.

The accuracy of predictions is shown using zone with the highest density of rides (Zone ID=161). The model for this zone also has the highest mean squared error. How the predictions of next 30–60 minutes interval stands against the actual demand is shown in (Figure 12). For this zone the MSE is 2963. Across all the zones the MSE is averaging 663.

Experimentally this seemingly high value can be further lowered by merging training batches

for zones with similar curves of demand. This leads to a hypothesis that currently we do not have enough data to train LSTM, if only data from one zone are used. On the other hand, it is objectively hard to define which zones are similar or not and thus each model uses data only from its dedicated zone.

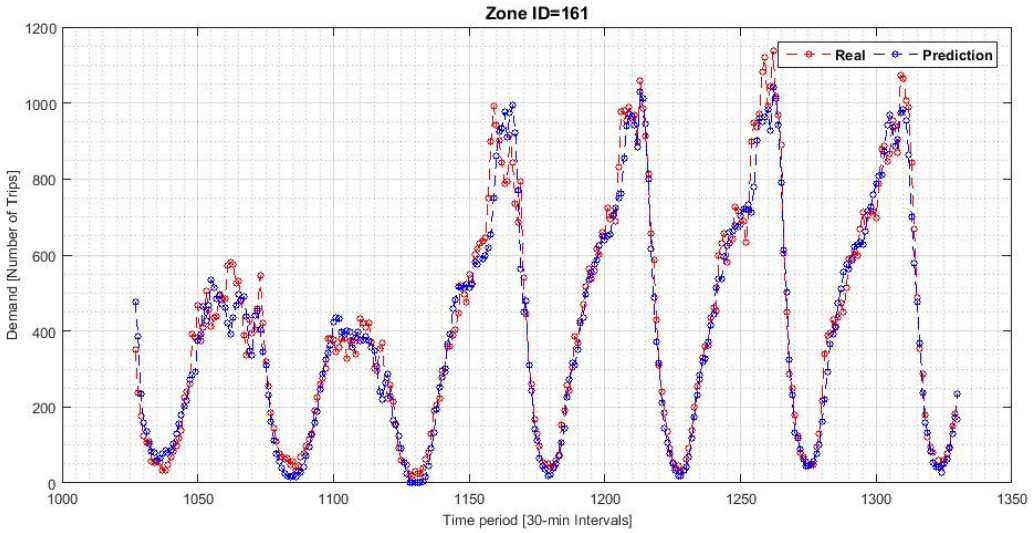

Figure 12: Prediction of demand vs actual demand in the last week of June

9 Conclusion & Future Research Direction

Table 3 is the summary of our models and their testing errors. While DSHW and LSTM are showing MSE of around high 600s, LS-SVM shows testing MSE of 465, which is significantly lower than the other two models. From the results we have obtained from our experiments, we conclude that out of the three models, LS-SVM performs the best in predicting the demands of taxi and TNC.

Models	MSE for testing data
DSHW	688
LS-SVM	465
LSTM	663

Table 3: Model Comparison

However, it is to be noted that these results still have quite large MSEs. Also, they are collected from only a small amount of data (one month), and the experiments were not extensive, due to time constraints. Therefore, experimenting with more data and different hyper-parameter is crucial to make a more solid conclusion of the research.

In future research, we will continue to improve the performance of the three models, as they show some promise to predict the demands of taxis and TNC. In general, we will need to use more data, possibly an entire year or more, to acquire better results. For DSHW, a more efficient optimization method needs to be used for acquiring optimal parameters α, γ, and δ, as we estimated the said parameters using simple loops. Also, more experiments should be conducted using different time windows, instead of one fixed time window of two weeks, for better performance of the model. As for LS-SVM, we suspect that it is overfitting to the data. Moreover, we are using a more complex model due to estimation of individual parameters per zone, instead of a more general model. By having more data to experiment with, we believe these problems can be overcome. We did not have too much time to experiment with LSTM, as the training takes a lot more time than the other two models. Moreover, we suspect that we are lacking data for the model to perform at an optimal level. Therefore, training with more data would be the next task for improving LSTM.

A “widening" of the data input is inevitable for future research. For example, we have discussed using the land usage or weather as part of the data input in the beginning. However, due to lack of time, we decided to start simple. If such data is taken into consideration, we believe the performance of our models will increase, as we believe such data have direct correlation with the demands.

Finally, although we have some promising results from our short project, we believe it is not ready to be implemented in real life to help the NYC taxis survive in the storm of TNCs. It seems like the taxi companies will have to fight the battle on their own a little while longer.

REFERENCES

[Cochrane, 2018] Cochrane, C. (2018). Time series nested cross-validation.

[Davis et al., 2018] Davis, N., Raina, G., and Jagannathan, K. (2018). Taxi demand forecasting: A hedge-based tessellation strategy for improved accuracy. *IEEE Transactions on Intelligent Transportation Systems*, 19(11):3686–3697.

[Deri and Moura, 2015] Deri, J. A. and Moura, J. M. (2015). Taxi data in new york city: A network perspective. In *2015 49th Asilomar Conference on Signals, Systems and Computers*, pages 1829–1833. IEEE.

[Donovan and Work, 2014] Donovan, B. and Work, D. (2014). New york city taxi trip data (2010-2013).

[Dudley et al., 2017] Dudley, G., Banister, D., and Schwanen, T. (2017). The rise of uber and regulating the disruptive innovator. *The political quarterly*, 88(3):492–499.

[Gers et al., 1999] Gers, F. A., Schmidhuber, J., and Cummins, F. (1999). Learning to forget: Continual prediction with lstm.

[Hyndman et al., 2008] Hyndman, R., Koehler, A. B., Ord, J. K., and Snyder, R. D. (2008). *Forecasting with exponential smoothing: the state space approach*. Springer Science & Business Media.

[Jiang et al., 2019] Jiang, S., Chen, W., Li, Z., and Yu, H. (2019). Short-term demand prediction method for online car-hailing services based on a least squares support vector machine. *IEEE Access*, 7:11882–11891.

[Jonas, 2015] Jonas, A. (2015). Share and share dislike: The rise of uber and airbnb and how new york city should play nice. *JL & Pol'y*, 24:205.

[Ke et al., 2017] Ke, J., Zheng, H., Yang, H., and Chen, X. M. (2017). Short-term forecasting of passenger demand under on-demand ride services: A spatio-temporal deep learning approach. *Transportation Research Part C: Emerging Technologies*, 85:591–608.

[Laptev et al., 2017] Laptev, N., Yosinski, J., Li, L. E., and Smyl, S. (2017). Time-series extreme event forecasting with neural networks at uber. In *International Conference on Machine Learning*, volume 34, pages 1–5.

[Li et al., 2012] Li, X., Pan, G., Wu, Z., Qi, G., Li, S., Zhang, D., Zhang, W., and Wang, Z. (2012). Prediction of urban human mobility using large-scale taxi traces and its applications. *Frontiers of Computer Science*, 6(1):111–121.

[Makridakis et al., 2018] Makridakis, S., Spiliotis, E., and Assimakopoulos, V. (2018). The m4 competition: Results, findings, conclusion and way forward. *International Journal of Forecasting*, 34(4):802–808.

[Moreira-Matias et al., 2013] Moreira-Matias, L., Gama, J., Ferreira, M., Mendes-Moreira, J., and Damas, L. (2013). Predicting taxi–passenger demand using streaming data. *IEEE Transactions on Intelligent Transportation Systems*, 14(3):1393–1402.

[Mukai and Yoden, 2012] Mukai, N. and Yoden, N. (2012). Taxi demand forecasting based on taxi probe data by neural network. In *Intelligent Interactive Multimedia: Systems and Services*, pages 589–597. Springer.

[Offenhuber and Ratti, 2014] Offenhuber, D. and Ratti, C. (2014). *Decoding the city: Urbanism in the age of big data*. Birkhäuser.

[Rutledge et al., 2006] Rutledge, G. K., Alpert, J., and Ebisuzaki, W. (2006). Nomads: A climate and weather model archive at the national oceanic and atmospheric administration. *Bulletin of the American Meteorological Society*, 87(3):327–342.

[Smith, 2018] Smith, M. (2018). Can we predict when and where a crime will take place?

[Taylor, 2003] Taylor, J. W. (2003). Short-term electricity demand forecasting using double seasonal exponential smoothing. *Journal of the Operational Research Society*, 54(8):799–805.

[Wittman, 2002] Wittman, T. (2002). Time-series clustering and association analysis of financial data. *University of Texas, Austin*.

[Xu et al., 2017] Xu, J., Rahmatizadeh, R., Bölöni, L., and Turgut, D. (2017). Real-time prediction of taxi demand using recurrent neural networks. *IEEE Transactions on Intelligent Transportation Systems*, 19(8):2572–2581.

[Yao et al., 2018] Yao, H., Wu, F., Ke, J., Tang, X., Jia, Y., Lu, S., Gong, P., Ye, J., and Li, Z. (2018). Deep multi-view spatial-temporal network for taxi demand prediction. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

[Zhang et al., 2016] Zhang, K., Feng, Z., Chen, S., Huang, K., and Wang, G. (2016). A framework for passengers demand prediction and recommendation. In *2016 IEEE International Conference on Services Computing (SCC)*, pages 340–347. IEEE.

[Zhao et al., 2016] Zhao, K., Khryashchev, D., Freire, J., Silva, C., and Vo, H. (2016). Predicting taxi demand at high spatial resolution: Approaching the limit of predictability. In *2016 IEEE International Conference on Big Data (Big Data)*, pages 833–842. IEEE.