# Using Neural Network for Predicting Hourly Origin-Destination Matrices from Trip Data and Environmental Information

Ehsan Hassanzadeh[1], Zahra Amini[*,1]

[1] Department of Civil Engineering, Sharif University of Technology, Tehran, Iran

E-mail Addresses: ehsan.hassanzadeh@student.sharif.edu (E. Hassanzadeh), zahra.amini@sharif.edu (Z. Amini)

Submitted for review and possible publication in
Scientia Iranica

Corresponding author:
Zahra Amini
Azadi Avenue
Tehran
Iran
Email: zahra.amini@sharif.edu
Phone:

## Biography

**Ehsan Hassanzadeh** is a graduated Master’s student in transportation engineering at the Sharif University of Technology, Tehran, Iran. In his master’s studies, he has focused on transportation network modeling and traffic percolation under the supervision of Dr. Zahra Amini. During his Master’s career, he has also focused on applications of Machine Learning methods in Transportation Engineering, specifically on the application of Neural Networks in estimating travel demands. He holds a bachelor’s degree in Civil Engineering from the Ferdowsi University of Mashhad and he has started his Ph.D. studies in Transportation Engineering at the University of British Columbia (UBC).

**Zahra Amini** is currently an Assistant Professor at the Department of Civil Engineering, Sharif University of Technology. She completed her bachelor’s degree in 2014 and her master’s degree in 2015, in Civil Engineering at University of California, Berkeley. She obtained her Ph.D. in Highway and Traffic Engineering, in 2018 at University of California, Berkeley. Her research interests are Intelligent Transportation System (ITS), traffic theory and control strategies, and transportation system operation and management.


## Abstract

Predicting Origin-Destination demand has always been a challenging problem in transportation. Conventional demand prediction methods mainly propose procedures for forecasting aggregated temporal Origin-Destination (OD) flows. In other words, they are primarily unable to predict short-term demands. Another limitation of these models is that they do not consider the impact of environmental conditions on trip patterns. Furthermore, OD demand prediction requires two individual steps of modeling: trip generation and trip distribution. This article presents a framework for predicting hourly OD flows using the Neural Network. The proposed method utilizes trip patterns and environmental conditions for predicting demands in single-step modeling. A case study on New York City Green Taxi 2018 trip data is done to evaluate the method, and the results demonstrate that the network has reasonably accurate OD flows predictions.



**Keywords:** O/D demand prediction; Short-term prediction; Neural Network; Machine Learning, Trip Generation


## Highlights

- Short-time O/D flow prediction is proposed to be obtained by Neural Network models
- The proposed Neural Network model uses past trip and environmental condition data
- The proposed model may replace aggregate distribution models for short-time predictions

## 1. Introduction

In recent years, demand for public transport has increased significantly due to urban development and population growth. One option to meet this demand is public transportation network expansion, which is expensive and has many limitations [1]. A more appropriate solution is network management with the available facilities. Network management includes strategies and policies to fulfill demand in the system and utilize the facilities more effectively [2]. A preliminary requirement if devising such management plans is predicting and modeling users' travel behavior.

One of the most popular models in demand modeling is the Four-Step Model (FSM), which contains the following steps: 1. trip generation, 2. trip distribution, 3. mode choice, and 4. route choice [3]. Although this paper mainly focuses on the first two steps and assumes predicting O/D demands for a single mode, some dynamic factors are considered that could alter users' mode choices. The second step, trip distribution, distributes generated trips to match destinations. In the third step, mode choice and trip modes proportions are specified, and the user behavior is modeled using the Consumer Choice theory. Based on this theory, consumers' preference is affected by utility functions, which are not deterministic [4–6]. By assuming a uniform distribution for the random component of the utility function across the modes, it can be concluded that the difference in utilities is only due to the difference in the systematic part. However, it should be noted that researchers have assumed that the random components are independent but non-identically distributed [4].

Nonetheless, reviewing the random component's distributions is beyond the scope of this paper, and since it will not be used in this research's modeling process, the distribution is assumed to be uniform to explain the impact of the systematic component. As a result, users' mode choice behavior can be modeled and predicted by considering systematic utility components, including consumers' socio-economic attributes, the vehicle's operational characteristics, and the trip distribution table [4]. Moreover, previous studies show that environmental conditions (e.g., weather data, land use, and other related parameters) have substantial impacts on travel times, public transportation systems quality, and daily travel behaviors [7–10]. As a result, these parameters can be considered in the consumer's utility function. The relationship of these parameters with the utility function (e.g., linear or nonlinear) would be determined by the Neural Network in this project by creating dummy variables in the hidden layers; hence, this paper will not discuss the possible relationships with the utility function. Given that the other parameters, such as the socio-economic attributes (e.g., the impact of income on the mode choice), do not vary over short periods, this paper disregards such parameters as they are assumed to be unchanged over the study period.

This paper aims to predict hourly OD flows for a single specific mode using the Neural Network (NN) without users' information (e.g., income, car ownership). To predict OD flows, input parameters reflecting the consumer's utility and other parameters regarding trip information are used. Then to evaluate the proposed method, this study uses New York City Green Taxi 2018 trip data and New York City 2018 weather data. The trip data consists of 8.81 million trip information, including trips' origin and destination zones number, trip distance, and other related trip information. Another dataset that is used for training the model includes hourly weather data of the desired location.

The rest of this article is organized as follows. Section 2 reviews related works on OD prediction and studies using a similar dataset. Section 3 discusses the proposed framework to predict OD flows in detail. Then, section 4 describes datasets used to evaluate the proposed model and data verification by investigating existing trip patterns. Section 5 discusses model results, and the final section provides the conclusion.

## 2. Literature review

Demand modeling has been a prominent research area in transportation for years, and the FSM has been one of the most comprehensive approaches for demand modeling. This approach underlies methods to predict mode-specific demand [11]. The Gravity model is widely used for trip distribution in the trip generation step of the FSM. This model distributes trips between zones based on the zones' relative attraction and a function of distances between zones [12]. The model is calibrated on a single OD table, including aggregated trip data. Thus, applying the Gravity model for estimating hourly OD flows is not practical. Moreover, the Gravity model considers a limited number of parameters to calibrate the model. Studies have shown that the Gravity model output has little similarity with the observed data [13]. It can be inferred from the Gravity model that the origin and destination zones and a function of distances between the zones (either temporal or spatial) should be considered in the modeling.

The third step of the FSM involves using discrete choice models to understand user behavior when selecting transportation modes. The Logit and Probit models are commonly used in this step, which use utility functions to determine choice probabilities [14,15]. However, collecting user information to define these utility functions can be challenging. These models have limitations, such as assuming a constant relationship between dependent and independent variables, making them inflexible and unable to adapt dynamically. They also perform poorly when input variables are multicollinear [16–18]. This paper focuses on parameters that can affect user choices based on environmental conditions, rather than explicitly modeling mode choice.

Recent studies have explored the use of Neural Networks (NN) as an alternative to traditional methods for predicting OD matrices and mode choices [19–23]. Researchers have compared the accuracy and performance of NN models with other statistical methods, such as the Multinomial Logit Model (MNL) [24], mode choice modeling [25], and Bayesian Model Tree [26]. Xiong et al. [27], proposed a framework that used Graph Neural Networks (GNN) and Kalman filters to predict OD flows based on historical link flows. Yaldi et al. [28] used NN models with three input parameters to predict trip flows. However, these approaches have limitations in considering the factors impacting OD flows and user behavior. The current paper proposes a new approach that uses NN models to predict trip flows based on trip patterns, environmental conditions, and consumer preferences. Like the method proposed by Xiong et al. [27], the framework used by Yaldi et al. [28] limits input parameters to trip interchange attributes, ignoring environmental attributes affecting users' behaviors. In contrast, the current paper implements the NN to predict trip flows using trip patterns and environmental conditions considering consumer preferences.

Researchers have realized that environmental conditions may impact traffic patterns in various ways. Liu et al. claimed that weather parameters, including temperature, snowfall, and precipitation, substantially impact travel behaviors [7]. They showed that these weather parameters affect all travel modes, including pedestrian walking, bicycle, private car, and public transport. In another study, Rudloff et al. evaluated relations between weather conditions and trip patterns using mode choice models. They estimated choice models' parameters based on household survey data from Vienna, employing the maximum likelihood approach. Their results showed that weather conditions significantly influence transport choice and travel behavior [8]. Hyland et al. investigated the effects of weather conditions on travel mode choice using a stated preference (SP) survey in Chicago and realized that commute choice patterns differ vastly in various weather conditions. Furthermore, they claimed that the impacts of weather on mode choices vary across the community [29]. Thus, the present paper considers weather conditions as effective environmental parameters while training the network for predicting OD flows.

The NN is mainly trained on the existing trip patterns to learn future predictions. Consequently, it is essential to create parameters considering different trip patterns to have a more accurate estimate of the future. Studies have shown that different trip behaviors are observed on weekends and holidays compared to workdays. Dong et al. [30] used trajectory data collected from ride-hailing services in Beijing, China, to investigate urban trip patterns. Their results showed tangible differences in trip distributions between particular zones. Specific hourly patterns also justify considering the hour of the day as an effective parameter in network training. They observed a notable difference between workday and non-workday trip patterns for various trip purposes. Other researchers have also shown the importance of the time of the day in predicting OD flows [31–33]. These results reveal the importance of considering weekends, holidays, and hour of the day in model training.

Another aspect of this research is dealing with big data for transportation analysis. In this regard, numerous research uses big data for various types of analysis. To name a few related research, [34,35] use big data for analyzing a specific mode of transportation. The latter also focuses on the impacts of COVID-19 on bike-sharing systems. Another similar approach to dynamically predict trip patterns using the NN is to apply agent-based day-to-day models. In this field, many papers focus on trip-related information and how it can impact traffic conditions by applying agent-based models [36,37]. Based on the nature of the problem, which includes various parameters impacting traffic patterns and mode choices, this paper opts to utilize the NN for predicting OD flows.

This article uses 2018 New York City taxi data from NYC Open Data to evaluate the proposed framework's performance. Related works on similar datasets are as follows. Deri et al. used similar 2010-2013 New York City taxi data and presented a solution for estimating taxi trajectories using Dijkstra's algorithm with a significantly reduced computation time [38]. In another study, Freire et al. discussed cleaning Spatio-temporal data. They used 2008-2012 New York City taxi data to observe the anomalies in the dataset. Results showed that data exploration needs users' assistance, and the lack of adequate information about events prevents the system from discerning anomalies [39]. Patel et al. proposed an approach to visually explore big OD data and determine average hourly drivers' revenue. They used 2014 New York City taxi data to evaluate their method. Unlike related works using a similar dataset [40], this paper aims to predict OD flows considering the abovementioned parameters.

## 3. Materials and Methods

This section describes data cleaning procedures and obtaining various input and output parameters required for modeling and the network structure. Input parameters (independent parameters in modeling) indicate parameters used as the network's input to predict output parameters (dependent parameter in modeling), the OD flow per hour. In other words, hourly input data are used to predict the hourly OD flows. This section divides the proposed algorithm's procedure into four major steps, as summarized below:

- *Step 1:* Obtaining Input Parameters (Independent Parameters) data for training and testing the NN (See Section 3.1). This step consists of the following minor steps:
    - *1A:* Obtaining OD zone IDs and hour of the day (See Section 3.1.1)
    - *1B:* Obtaining interzonal travel times (See Section 3.1.2)
    - *1C:* Obtaining binary parameters (See Section 3.1.3)
- *Step 2:* Obtaining output parameters data (number of trips for each pair of OD zones at each time step) for training and testing the NN (See Section 3.2).
- *Step 3:* Cleaning obtained data to remove any outliers that may deteriorate the NN accuracy (See Section 3.3).
- *Step 4:* Reshaping input and output data matrices to be fed into the NN for training (See Section 3.4).

- *Step 5:* Standardize data to avoid bias in training the NN (See Section 3.5).
- *Step 6:* Building the NN's structure and training the network (See Section 3.6).

According to the possible factors affecting trips described in earlier sections, this paper considers the network's input parameters as follows:

- Interzonal travel times, including calculated hourly travel times for all possible OD pairs
- Origin zone, defined by a unique ID
- Destination zone, defined by a unique ID
- Hour of the day, specified by a number within the range zero to 23
- Weekend/ weekday binary classification
- Holiday/ nonholiday binary classification
- Temperature, including hourly resolution records
- Precipitation, including hourly resolution records
- Snow depth, including hourly resolution records

This paper assumes that certain parameters such as passenger count and fare amount cannot be determined without access to corresponding demand data or algorithms used to calculate these parameters. Therefore, these parameters are not used as inputs in the model. Additionally, input parameters with intercorrelation are omitted, such as distance between origin and destination zones. The procedures for obtaining each parameter are explained in detail below.

### *3.1. Obtaining input data*

As described earlier, each hourly input record comprises nine factors. Three of these nine parameters, which are temperature, precipitation, and snow depth, can be obtained directly from datasets for each time step. Obtaining the remaining six requires additional steps, which are described below.

#### *3.1.1. Obtaining OD zone IDs and hour of the day*

In this research, the "hour of the day" variable is defined as the departure hour for each trip. This definition may raise an error since trips are not necessarily finished in the same hour as they started. Since it is assumed that the desired trips only consist of urban trips, trip durations would be reasonably short; hence the error is negligible. Zone IDs can be defined as the assigned IDs for each Traffic Analysis Zones (TAZs). Therefore, each trip's destination and origin can be determined with two IDs demonstrating its origin and destination. Traffic Analysis Zones can be specified using the available datasets for the research area or by defining the TAZs using the available methods [41].

#### *3.1.2. Obtaining interzonal travel times*

The Gravity model examines how distances between zones impact OD flows, but this article suggests using hourly travel times between zones to consider the impact of traffic flows. Hourly travel times can be obtained from various services like Google Maps, but the calculation of shortest paths requires real-time traffic data. Since this research aims to predict hourly OD flows, all input and output data should be aggregated into hourly records. The article proposes calculating average travel times between OD pairs in each hour after removing outliers to represent the hourly travel time for all the trips between the OD zones. Section 3.3 provides more details on the removal of outliers. Then, interzonal hourly travel times are obtained as a linear matrix, $TT_k$, according to Equation (1).

$$TT_k = \left[ tt_{1,1,k}, ..., tt_{1,n,k}, ...., tt_{i,j,k}, ..., tt_{n,n,k} \right] \tag{1}$$

$TT_k$ = linear travel time matrix in the hour *k*, $tt_{i,j,k}$ = travel time for the *ij* OD pair for the hour *k*.

Depending on the dataset used, travel times for the OD pairs may be obtained using a particular method available. As will be discussed in Section 4.1, the dataset used in this study includes the start time and end time for each trip record. Therefore, each trip's travel time can be simply calculated by computing the in-time vehicle for each trip.

### *3.1.3. Obtaining binary parameters*

As described in section 2, users have different traffic behaviors on weekends and holidays than on regular weekdays. Two binary parameters are defined to address this variation: *"Weekend"* and *"Holidays,"* parameters which indicate whether the trip was on the weekend or holiday or not, respectively. These parameters' values are equal to zero if the desired day is not a holiday or a weekend. It is worth mentioning that the holidays can be specified using the national holidays' list for the desired database. After obtaining all parameters as discussed, linear matrices of hourly attributes, $att_k$ can be created according to Equation (2). The data source for obtaining these parameters will be discussed in Section 4.2.

$$att_k = \left[ hr_k, weekend_k, holiday_k, temp_k, PCP_k, SD_k \right] \tag{2}$$

$att_k$ = linear attributes matrix of the hour *k*, and for the hour *k*: $hr_k$ = the hour *k* of the day, $Temp_k$ = the hour *k* hourly temperature, $PCP_k$ = the hour *k* hourly precipitation, $SD_k$ = the hour *k* snow depth.

## *3.2. Obtaining output data*

The network's output parameter for each hour is an OD flow matrix showing trip counts between each OD pair. The OD flow matrices at each hour are created by counting the trips between each pair of OD in the trips database. In this study, origins and destinations are considered Traffic Analysis Zones (TAZs), which can be specified by assigning zone IDs. The hourly OD matrix is created as shown in Equation (3).

$$OD_k = \begin{bmatrix} T_{1,1} & \cdots & T_{1,n} \\ \vdots & T_{i,j} & \vdots \\ T_{n,1} & \cdots & T_{n,n} \end{bmatrix} \tag{3}$$

$T_{i,j,k}$ = trip counts from zone *i* to zone *j*, for the hour *k*. *k* = the data record index representing the hour *k*

It should be noted that hourly trips are counted based on their start time (i.e., departure time). These matrices are then reshaped to linear matrices, as shown in Equation (4), to simplify the network's training process since it would be less baffling to acquire one row of data per record when feeding the input data to the network.

$$OD_k = \left[ T_{1,1,k} \ldots\ldots T_{1,n,k} \; T_{i,j,k} \ldots\ldots T_{n,n,k} \right] \tag{4}$$

These matrices are then added to the final output matrix, $T$, according to the occurrence time, starting from the first hour of the initial day (*k = 0*) to the last hour of the last day in the period (*k = t*). Thus, the output matrix, $T$, would be a $(t, n \times n)$ dimensional matrix according to Equation (5). Each row of the output matrix (i.e., dependent variable) indicates hourly trip counts for an OD pair. The predicted results after training the model will also follow the same format.

$$T = \begin{bmatrix} OD_0 \\ \vdots \\ OD_t \end{bmatrix} \tag{5}$$

### *3.3. Data Cleaning*

The existence of errors in data will result in bias in the network's training process. As a result, possible errors should be omitted from the data before using it for training. This section discusses removing outliers and possible errors from input and output data. To do so, the Z-score is calculated for each record to identify outliers in the data. The Z-score indicates the distance between the observed value and the sample's mean in the standard deviation units [42]. The Z-score can be calculated using Equation (6).

$$z = \frac{x - \mu}{\sigma} \tag{6}$$

$z$ = the standard score, $x$ = the observed value, $\mu$ = the mean of the sample, $\sigma$ = the standard deviation of the sample

After calculating the Z-score, records with $|z| \geq 3$ are considered outliers (preserving 99.8% of the data range). It should be noted that the threshold for removing the records is calculated after investigating the results by examining different thresholds. This procedure should be done for all input and output parameters with possible errors. Besides, constraints should be set for each parameter to ensure that all remaining data are valid. For example, travel time values should be positive and over 60 seconds. Values exceeding these ranges should be omitted based on the parameter range.

### *3.4. Reshaping data matrices*

So far, the origin and destination zone IDs have not been determined in the input and output matrices. As discussed before, each pair of OD has a unique traffic pattern. So, it is crucial to consider origin and destination zones as parameters for training the model. Then, input and output data are reshaped so that each row of data matrices represents dependent and independent values for a specific OD pair in a specific hour. The final output matrix, $T$, would be as shown in Equation (7).

$$T = \begin{bmatrix} OD_0 \\ \vdots \\ OD_k \\ \vdots \\ OD_t \end{bmatrix} \rightarrow \begin{bmatrix} T_{1,1,0} \\ \vdots \\ T_{n,n,0} \\ \vdots \\ T_{i,j,k} \\ \vdots \\ T_{n,n,t} \end{bmatrix} \tag{7}$$

To create the final input matrix, $X$, the travel times matrices, $TT_k$, should be first reshaped similar to the output matrix. Then, trip attributes parameters can be appended, including origin and destination zone IDs for each record, duplicating common attributes for all the trips in the desired hour. As a result, the final input matrix, $X$, is according to Equation (8).

$$X = \begin{bmatrix} tt_{1,1,0} & i & j & hour_0 & weekend_0 & holiday_0 & temp_0 & PCP_0 & SD_0 \\ & & & & \vdots & & & & \\ tt_{i,j,k} & i & j & hour_k & weekend_k & holiday_k & temp_k & PCP_k & SD_k \\ & & & & \vdots & & & & \\ tt_{i,j,t} & i & j & hour_t & weekend_t & holiday_t & temp_t & PCP_t & SD_t \end{bmatrix} \tag{8}$$

$tt_{i,j,k}$ = calculated travel time for the origin zone *i* and the destination zone *j* in the hour *k*

$i$, $j$ = origin and destination zone IDs for each record

### *3.5. Data Standardization*

Due to the significant variances between parameter values (either the difference between values of one parameter or the diversity between the data range of various parameters), the network's training process may be biased. As a result, large trip counts in the OD matrix, which are vital for modeling, could be recognized as outliers. All parameters' values are standardized in their category to address this issue, making the mean of each parameter zero and the standard deviation of parameter one. Standard values are calculated using Equation (9) [43].

$$x' = \frac{x-\mu}{\sigma} \tag{9}$$

$x'$ = the standardized value, $x$ = the observed value, $\mu$ = the mean of the sample, $\sigma$ = the standard deviation of the sample

It should be noted that the network's predicted data will be calculated in the normalized format and must be converted to the original format for evaluating the model.

### *3.6. The Neural Network's structure*

The Neural Network is a supervised machine learning method in which the network is trained first using a set of data with pre-defined outputs. The network tries to minimize the defined objective function to achieve the most desirable results by finding connections between input and output nodes. The NN is composed of multiple layers including input, hidden, and output layers. Hidden layers identify possible relations between parameters and provide a representation of data with multiple layers of abstraction. Each layer has a specific activation function to transmit the data format for the next layer. The NN optimizer updates weights based on the gradients computed in each iteration through an iterative backpropagation process [44].

The NN used in this paper has two hidden layers, as illustrated in Figure 1. The input layer is provided with the input parameters and transmits input data directly to the next layer via the neurons. As mentioned earlier, the number of nodes in this layer equals the number of input parameters, which is nine. The input and output layers' dimensions are 9 and 1, implying the input feature vector's dimensionality and prediction value. The predicted value here is the hourly OD flow, and each predicted value defines the predicted flow for a specific OD pair in a specific hour of the desired period. To determine the number of nodes in the hidden layer, different numbers of nodes can be chosen, and then the output results of the network can be compared. As a general experimental rule, the number of nodes in the hidden layers is chosen close to the average input and output number of nodes. After investigating the results with different counts of nodes for the hidden layers, 7 and 5 nodes are finally considered for the hidden layers, respectively, as shown in Figure 1.

Activation functions are added to the NN to convert the previous layer's output values into desirable input values for the next layer. The ReLU (Rectified Linear Unit) activation function is used for hidden layers in this network as it offers better performance and generalization than the other activation functions used for predicting a numerical value [45]. The ReLU function can be written as Equation (10).

$$ReLU(x) = \max(0, x) = \begin{cases} x, & if \;\; x \geq 0 \\ 0, & if \;\; x < 0 \end{cases} \tag{10}$$

As illustrated in Figure 1, the output layer connects the last hidden layer to the output values. This layer has only one node, which is the network output value. Activation functions like the Sigmoid function result in output values between 0 and 1 that predict categorical values. The output values in this work are numerical; thus, the Linear function is used as an activation function for the output layer, expressed as Equation (11).

$$f(x)=ax \tag{11}$$

The Mean Squared Error (MSE) loss function is used in this study to calculate the difference between the actual and the prediction value. This function computes the average squared difference between the actual and predicted values using Equation (12) [43]. The MSE is also used as the metric parameter for keeping track of performance measures (i.e., the objective function of the NN).

$$MSE=\frac{1}{n}\sum_{i=1}^{n}\left(y_i-\hat{y}_i\right)^2 \tag{12}$$

$y_i$ = the actual value, $\hat{y}_i$ = the predicted value, $n$ = number of predicted values

The "learning rate" is a crucial hyperparameter in NN that determines the step size for weight updates during the optimization process. Overfitting is a common issue when data is similar to each other in time, and to avoid it, the "shuffle" parameter should be used while training the model. The Adam optimizer is used for updating weights in the network, and it can adjust the learning rate with the "decay" parameter. The network's weights are updated iteratively using batches, with each batch representative of the dataset, and the batch size should be large enough to include non-zero values. The dataset is divided into three splits of training, testing, and validation, with each containing a different percentage of the data. The optimal values for the hyperparameters require evaluating different values on the dataset.

The process of updating the weights of a neural network is iterative and occurs over many epochs. An epoch consists of one forward and backward pass of the entire dataset, which is usually too large to be fed into the network at once, so it is divided into batches. It is important to choose a batch size that is representative of the dataset to prevent errors. The batch size parameter should be large enough to include non-zero values in a batch since multiple output values are often zero. The correct values for the batch size and epoch parameters should be chosen by assessing different data values. Finally, the dataset is split into three sets: training, testing, and validation, with percentages of 56%, 30%, and 14%, respectively.

## 4. Case Study

A case study was done using open-source trip and weather datasets to evaluate the proposed framework. These datasets are reviewed in detail in the following sections.

### *4.1. Trip Data*

This study uses open-source data provided by the Taxi and Limousine Commission (TLC) available on the NYC Open Data website [46]. The dataset consists of 8.81 million New York

Green Taxi 2018 trip records. Each record includes pickup date and time, drop-off date and time, trip's origin and destination zones ID, and other fields shown in Table 1 (redundant fields for modeling such as tax and distance are ditched).

According to the City Zones dataset available on the NYC Open Data, zone IDs denoted in Table 1 represent specific taxi zones defined by the Department of City Planning [47]. As shown in Figure 2, there are 265 zones numbered from 1 to 265. Table 2 shows samples of this dataset. This study uses this definition of zones for the case study instead of TAZs for specifying origins and destinations.

### *4.2. Weather Data*

Section 2 discussed that the weather conditions substantially impact daily travel behaviors. This study uses an open-source weather dataset from the National Climatic Data Center (NCDC) [48]. This dataset contains hourly recorded weather information, including date and time, temperature, wind speed, wind direction, 1-hour liquid precipitation, 6-hour liquid precipitation, and snow depth, as shown in Table 3.

### *4.3. Data Preparation*

Now that the datasets are explained, procedures done on the datasets described in section 3 are summarized here. After setting constraints for each parameter, values exceeding these constraints are omitted. Then, examining the remaining data reveals that there are apparently no outliers remaining in the dataset, as they were possibly removed in the last step. However, there would still be outliers while calculating average travel times, which will be removed based on the Z-score. Moreover, additional constraints should be set for some parameters. For example, calculated travel times with minimal positive values (e.g., less than a minute) may not be omitted based on the initial constraint of being positive and also may not be detected as outliers. After cleaning the datasets from possible errors, only the data for the first quarter of the year (90 days) is used in this study to prevent inundating the network while training. So, indices of data rows considering the available 265 zones are calculated as 265*265*90*24 = 151,686,000 indices.

### *4.4. Data Verification*

It is necessary to verify the validity of the acquired data to prevent possible errors before training the network. Data verification can be done by examining the conformity of trip patterns with previous studies. As a result, average trip patterns are investigated in this section to verify the validity of the dataset. To investigate the trip patterns, average hourly trip counts were computed for all weekends and weekdays in February 2018. To show the contrast between the trip patterns during the holidays and non-holidays, hourly trip counts of 1$^{st}$ January 2018 (New Year's Day) are computed as a sample. These patterns are plotted in Figure 3.

As illustrated in Figure 3, there are two peak hours in the morning and the evening for trip counts in the weekday average trip pattern, which are 8 a.m. and 6 p.m., respectively. The weekend average trip pattern shows that the morning peak hour is vanished (since there are no work-based trips in the morning) and the midnight trip is increased significantly compared to the average weekdays. The trip patterns from previous studies can be compared to similar hourly patterns of taxi trips on weekdays and weekends to validate the results [30,49]. The main difference between the holidays and non-holiday trip patterns (including non-holiday weekends) is that there are numerous holidays, and generally, traffic patterns are changed based on the holiday and related celebrations or rituals of the day. The reasons mentioned above substantially impact trip patterns (e.g., travel destinations change vastly). For instance, the hourly trip pattern on 1st January 2020 plotted in Figure 3 shows substantial differences compared to weekday and weekend trip patterns.

Trip patterns between OD zones can also be inspected to see differences in the patterns between weekends, workdays, and holidays. The weekday average trip pattern shows that trips between zones 74 and 75, East Harlem North and East Harlem South neighborhoods, have the most frequency at different day hours, including the morning and evening peak hours. According to the Office of the New York State Comptroller report, East Harlem is mainly a residential neighborhood with concentrated small businesses [50]. Weekends average trip pattern exposes that trips between zones 41 and 42 (Central Harlem and Central Harlem North neighborhoods) and internal trips of zone 7 (Astoria zone of borough Queens) have the highest frequency at different hours of the day. Inspecting New York City's Zoning and Land Use Map [51] indicates that Central Harlem and Astoria are commercial neighborhoods, including numerous recreational places, specifically the Astoria. Internal trips of the Astoria neighborhood also showed the highest frequency at different hours in the selected holiday. According to the City Zones dataset, these paths are illustrated in Figure 4.

These results gave us good insights into the differences between weekdays, weekends, and holiday trip patterns and the need to use binary parameters to address these variances. It can also be derived that the demand in the origin and destination zones is a function of land use. Therefore, it can be verified that the input data have rational patterns and can be used for training the network.

## 5. Results

### *5.1 Network Results*

"Keras is a deep learning API written in Python, running on top of the machine learning platform TensorFlow. It was developed with a focus on enabling fast experimentation." [52] This package provides the required functions for training the network in Python. After evaluating different values for the network parameters, a summary of the chosen values for the network's parameters is given in Table 4. After manually inspecting the network's prediction accuracy, these values are chosen by evaluating different values for each parameter.

As shown in Figure 5, it is observed that the error value converges to a relatively constant value after performing several epochs. Consequently, the number of epochs is chosen to be 10. The NN training results indicate the presence of MSE = 0.5798 after ten epochs, which is reasonable. It should be noted that these results cannot be compared directly to results from the trip distribution models, including the Gravity Model and the Fratar model, since these models predict aggregated trips for a period of time. On the contrary, this paper proposed a method to predict hourly trip counts. However, the hourly prediction may cause an increase in error, which comes from numerous possible scenarios in each hour. The output results of each epoch can be seen in Figure 5, and the loss reduction trend in Figure 6. The middle oscillations in Figure 6 indicate the beginning of a new epoch. The error reduction trend verifies that the network is working correctly. As shown in Figure 5, loss values decrease rapidly at first and then slowly after the third epoch.

It can be inferred from the results that the loss value is minimized after a limited number of iterations, and there is no need to increase the number of epochs. The network performance is then investigated on one million random samples from the test dataset. It is worth mentioning that the predicted results are rescaled to the original values and rounded to the nearest integer since they represent trip counts. Test results are given in Table 5.

Test results show that the number of predicted zero values in the OD matrix and the total predicted trips perfectly match the actual values, and the Mean Squared Error of 0.0348 confirms this. R Squared value of 0.453 shows the model's acceptable fit, but possible reasons for the R Squared's relatively small value are discussed here. One million samples are approximately equivalent to 14 hours of trips since there are 70,225 possible paths (a path is a possible route between OD pair) between zones in an hour, and the results show that there are 15,100 trips in one million random samples of the trips. It can be deduced that there is an average of 1,100 trips for the available 70,225 paths in one hour, which means the average hourly trip count for each path is a small value. It can also be derived from the results that the average value for non-zero trip counts is approximately equivalent to 1.7 trips. Hence, slight deviations from the actual value can be due to the rounded predicted values (e.g., the predicted value of 3 for the actual value of 2), decreasing the Coefficient of Determination (R Squared) vastly. As a result, R Squared's small value can not necessarily represent the model's inadequate goodness of fit, and the MSE is a better quantifier to evaluate the model's goodness of fit. Suggested solutions to reduce the existing errors are given in the discussion.

### *5.2 Validating Network Results*

In this section, the NN's results are compared to the Gravity model to validate the results. As mentioned before, the NN predicted hourly trips, and the Gravity model generates aggregated trip predictions; therefore, these results cannot be compared directly. So, the results of the Gravity model should be compared to the aggregated results of the NN. Although this study aims to predict hourly flows, comparing the aggregated form of the results with the traditional models is compulsory for validation. The Gravity model's general form can be expressed in Equation (13) [53,54].

$$T_{ij} = \frac{P_i A_j F_i K_j \times f\left(c_{ij}\right)}{\sum_v A_v F_i K_v \times f\left(c_{ij}\right)} \tag{13}$$

$T_{ij}$ = total trips between zones *i* and *j*

$P_i$ = total trips produced by zone *i*

$A_j$ = total number of trips attracted to zone *j*

$v$ = set of 265 zones

$f\left(c_{ij}\right)$ = decreasing function of the travel cost $c_{ij}$

$F_i$ = balancing factor ensuring $\sum_j T_{ij} = P_i$ (14)

$K_j$ = balancing factor ensuring $\sum_i T_{ij} = A_j$ (15)

The travel cost function in Equation (13) (friction function) is any decreasing function of the travel cost (which is assumed to be the travel time in this study). Hence, the friction function can be considered as the power function shown in Equation (16) [55].

$$f\left(c_{ij}\right)=\frac{1}{(c_{ij})^{n}} \tag{16}$$

$c_{ij}$ = average travel time between zones $i$ and $j$
$n$ = power variable

Since the Gravity model requires total productions and attractions for the prediction period (i.e., the NN test data), they should be estimated using the trip generation models. The trip generation models require access to socio-economic data, which are assumed to be unavailable in this study. As a result, zones' productions and attractions are estimated by applying linear regression to the train data to predict the number of attracted ( $A_j$ ) and produced ( $P_i$ ) trips for the test set. The training and test data in this section cannot be the same as before since the Gravity model predicts aggregated trips for the prediction period. Consequently, hourly records of the dataset are aggregated into daily records. The test data includes 27 daily productions and attractions records for all zones (30% of the whole period), and the training data includes 63 daily records. Although the trip generation estimation should be for the desired 27 days, the training and the test data are aggregated into data points of 9 days to increase accuracy. In other words, the test data is aggregated into three data points (each one including aggregated productions and attractions of all zones for nine days period), and the training data is aggregated into seven accumulated data points.

Then, linear regression is applied to each zone's seven data points to predict future productions and attractions. It is worth mentioning that the linear regression equation is calibrated for the productions and attractions of each zone separately. Hence, $265\times 2=530$ linear regression equations are calibrated, predicting three data points for future periods. Zones' production and attraction values are then compared to the actual values. Table 6 shows R Squared values of predictions for the zones' productions and attractions.

Linear regression results show a reasonable fit of the predicted productions and attractions with an R Squared of 0.99 and a reasonable error in predicting the total trips. The three predicted data points for each zone's attractions and productions are then aggregated to calibrate the Gravity model. As shown in Equation (14), the Gravity model requires average travel times for the forecasting period. The required travel times are calculated by averaging the non-zero travel times after removing the outliers, as described in section 3.3. It should be noted that the average travel times are calculated using the data from the 63-day training dataset. Since the zero average travel times cannot be used in Equation (16), the zone pairs with an average travel time of zero are assumed to have no trip interchanges. However, this assumption may increase the accuracy of the results since zone pairs with no trip interchanges in the 63 days training period are forced to have no trips in the future.

After implying the travel times in Equation (13), the $F_i$ and $K_j$ balancing factors are calibrated through an iterative process to ensure that the conditions expressed in Equation (14) and Equation (15) are met. The stop condition of this iterative process is as follows:

$$\max\left\{\max_{i\in v}\left(\left|1-\frac{P_i}{\sum_j T_{ij}}\right|\right),\ \max_{j\in v}\left(\left|1-\frac{A_j}{\sum_i T_{ij}}\right|\right)\right\}<0.05 \tag{17}$$

$T_{ij}$ = total trips between zones *i* and *j*

$P_i$ = total trips produced by zone *i*

$A_j$ = total number of trips attracted to zone *j*

$v$ = set of 265 zones

Then, the calibration process is done with different power variables, *n*, in Equation (16) to minimize the error. Table 7 shows the Gravity model results with different values for the power variable, including the iterations needed to meet the convergence condition expressed in Equation (17).

As shown in Table 7, *n* = 2 had the lowest error in prediction with an MSE of 2340 and an R Squared of 0.84. However, the aggregated NN results for the same period showed an MSE of less than 25. One point worth mentioning from the above table is that the R Squared value for the power variable of 6 is negative. While the R Squared name suggests that it may always range from 0 to 1, some exceptions may also be negative. In cases where the model predictions are not being compared to the observation that were used for calibrating the model, the Total Sum of Squared Errors component ($SS_{res}$) is not included in the Total Sum of Squares ($SS_{tot}$) [56].

$$R^2 = 1-\frac{SS_{res}}{SS_{tot}} \tag{18}$$

$SS_{res}$ = Total Sum of Squares of Residuals

$SS_{tot}$ = Total Sum of Squares

Hence, as Equation (18) suggests, the R Squared value could also be negative in such cases. It should be noted that R Squared has been criticized for the lack of reliability as a measure of predictive accuracy [57]. Therefore, a more suitable measure of accuracy should be used to compare the prediction results, which is MSE in this case. Moreover, the NN uses MSE as the metric to optimize the learning process, and as a result, the network tries to reduce MSE in each iteration. The results showed that the NN had a clear superiority to the Gravity model in this case, although the purpose of this study was to predict hourly OD flows, and the Gravity model is unable to predict the flows on an hourly basis. Besides, forecasting the future trip distribution using the Gravity model required additional steps to estimate the future trip generations (i.e., the second step of the FSM), while the NN can predict trip distributions more precisely without requiring further steps. Another point worth mentioning is that, as mentioned before, the output results of the gravity may not be directly compared to those of the NN. The reason is that the Gravity model predicts aggregated trips for a period of time, while the NN in this study aims to predict hourly trips. Although it can be argued that the aggregated trips of the NN output results can be compared to the Gravity model results, given that the NN is optimized to predict the hourly hours, this study is not focusing on such comparisons using visual descriptions (e.g., comparison plots)

## 6. Conclusion and Discussion

OD matrix prediction for a specific transit mode using traditional methods has always faced numerous problems, including data collection. Using traditional methods requires data collection for the trip generation and trip distribution steps of the FSM. This data includes users' socio-economic characteristics and travel expenses information, which requires time-consuming and costly collection methods, such as filling out questionnaires. Finally, due to the nature of these collecting data methods, there is a significant error in the collected data, and it is also challenging to update them periodically.

This paper aimed to facilitate this procedure using the data from data-driven transportation systems. Prediction results showed proper fit and the logical dependence of the output data on the input data. Other advantages of predicting trips using the NN compared to the traditional modeling methods are considering more scenarios (weekends/holidays and more), quickly updating the network with recent changes, and adequately forecasting OD flows on an hourly basis.

It can be inferred from the results that there are considerable differences in the number of trips between zones. As a result, some output values, which are numerically significant and essential to be included in predictions, are detected as outliers and have insignificant impacts on model training. As a potential research extension, paths between zones can be classified based on their traffic volume (e.g., low traffic, medium traffic, and high traffic) and then modeled for each category separately. Creating dummy variables indicating each category can also be done instead.

It should also be noted that the results represent the predicted part of the demand that taxi drivers could handle. In other words, there would be other trip demands exceeding the taxi service supply; therefore, as there are no data for the unanswered demands in the dataset, the calibrated model disregards such demands. A potential research extension includes datasets containing users' requests to consider the drivers' unhandled trip requests, especially during peak hours.

As described in section 4.4, holiday trips showed various patterns depending on the occasion. A potential research direction is to model each type of trip discussed in this article separately (e.g., separate modeling for weekdays and weekends) to increase accuracy. Using algorithms to detect abnormal trip patterns (e.g., gatherings, special occasions that are not officially registered, and social events) and separating them from other data used for training the network can also improve results. Trip data used in this study included origin and destination zones for each trip, including precise longitude and latitude of origins and destinations, resulting in more accurate travel times and improved results.

## Conflict of Interest

All the authors have no conflict of interest with the funding entity and any organization mentioned in this article in the past three years that may have influenced the conduct of this research and the findings.

## Acknowledgments and Declarations

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

## References

1. Mathew, T. V and Sharma, S., "Capacity expansion problem for large urban transportation networks", *J. Transp. Eng.*, **135**(7) (2009).
2. Halvorsen, A., Koutsopoulos, H. N., Ma, Z., and Zhao, J., "Demand management of congested public transport systems: a conceptual framework and application using smart card data", *Transportation*, **47**(5) (2020).
3. McNally, M. G., "The Four-Step Model", In *Handbook of Transport Modelling*, D. A. Hensher and K. J. Button, Eds., Emerald Group Publishing Limited, **1**, pp. 35–53 (January 1, 2007).
4. Bhat, C. R., "Random utility-based discrete choice models for travel demand analysis", *Transp. Syst. Plan. Methods Appl.*, **10**(1), pp. 1–30 (2003).
5. Manski, C. F., "The structure of random utility models", *Theory Decis.*, **8**(3) (1977).
6. Walker, J. and Ben-Akiva, M., "Generalized random utility model", *Math. Soc. Sci.*, **43**(3), pp. 303–343 (2002).
7. Liu, C., Susilo, Y. O., and Karlström, A., "Weather variability and travel behaviour – what we know and what we do not know", *Transp. Rev.*, **37**(6), pp. 715–741 (2017).
8. Rudloff, C., Leodolter, M., Bauer, D., Auer, R., Brög, W., and Kehnscherper, K., "Influence of Weather on Transport Demand: Case Study from the Vienna, Austria, Region", *Transp. Res. Rec.*, **2482**(1), pp. 110–116 (2015).
9. Tsapakis, I., Cheng, T., and Bolbol, A., "Impact of weather conditions on macroscopic urban travel times", *J. Transp. Geogr.*, **28**, pp. 204–211 (2013).
10. Litman, T., "Land Use Impacts on Transport How Land Use Factors Affect Travel Behavior" (2008).
11. McNally, M. G., *The Four-Step Model*, Emerald Group Publishing Limited (2007).
12. Bouchard, R. J. and Pyers, C. E., "Use of Gravity Model for Describing Urban Travel", *Highw. Res. Rec.*, **88** (1965).
13. Long, G. D., *An Evaluation of the Gravity Model Trip Distribution*, Texas Transportation Institute (1968).
14. McFadden, D. L., "Conditional Logit Analysis of Qualitative Choice Behavior", *Front. Econom.* (1974).
15. Louviere, J., Street, D., Carson, R., Ainslie, A., Deshazo, J. R., Cameron, T., Hensher, D., Kohn, R., and Marley, T., "Dissecting the Random Component of Utility", *Mark. Lett.*, **13**(3), pp. 177–193 (2002).

16. Ranganathan, P., Pramesh, C. S., and Aggarwal, R., “Common pitfalls in statistical analysis: Intention-to-treat versus per-protocol analysis”, *Perspect. Clin. Res.*, **7**(3), pp. 144–146 (2016).
17. Cramer, J. S., “The Origins of Logistic Regression”, *SSRN Electron. J.* (2005).
18. Lindner, A., Pitombo, C. S., and Cunha, A. L., “Estimating motorized travel mode choice using classifiers: An application for high-dimensional multicollinear data”, *Travel Behav. Soc.*, **6**, pp. 100–109 (2017).
19. Duan, Z., Zhang, K., Chen, Z., Liu, Z., Tang, L., Yang, Y., and Ni, Y., “Prediction of city-scale dynamic taxi origin-destination flows using a hybrid deep neural network combined with travel time”, *IEEE Access*, **7**, pp. 127816–127832 (2019).
20. Zhang, J., Che, H., Chen, F., Ma, W., and He, Z., “Short-term origin-destination demand prediction in urban rail transit systems: A channel-wise attentive split-convolutional neural network method”, *Transp. Res. Part C Emerg. Technol.*, **124**, p. 102928 (2021).
21. Chu, K. F., Lam, A. Y. S., and Li, V. O. K., “Deep Multi-Scale Convolutional LSTM Network for Travel Demand and Origin-Destination Predictions”, *IEEE Trans. Intell. Transp. Syst.*, **21**(8), pp. 3219–3232 (2020).
22. Krishnakumari, P., van Lint, H., Djukic, T., and Cats, O., “A data driven method for OD matrix estimation”, *Transp. Res. Part C Emerg. Technol.*, **113**, pp. 38–56 (2020).
23. Ke, J., Qin, X., Yang, H., Zheng, Z., Zhu, Z., and Ye, J., “Predicting origin-destination ride-sourcing demand with a spatio-temporal encoder-decoder residual multi-graph convolutional network”, *Transp. Res. Part C Emerg. Technol.*, **122**, p. 102858 (2021).
24. Lee, D., Derrible, S., and Pereira, F. C., “Comparison of Four Types of Artificial Neural Network and a Multinomial Logit Model for Travel Mode Choice Modeling”, *Transp. Res. Rec.*, **2672**(49) (2018).
25. Golshani, N., Shabanpour, R., Mahmoudifard, S. M., Derrible, S., and Mohammadian, A., “Modeling travel mode and timing decisions: Comparison of artificial neural networks and copula-based joint model”, *Travel Behav. Soc.*, **10** (2018).
26. Brathwaite, T., Vij, A., and Walker, J. L., “Machine Learning Meets Microeconomics: The Case of Decision Trees and Discrete Choice” (2017).
27. Xiong, X., Ozbay, K., Jin, L., and Feng, C., “Dynamic Origin–Destination Matrix Prediction with Line Graph Neural Networks and Kalman Filter”, *Transp. Res. Rec.*, **2674**(8), pp. 491–503 (2020).
28. Yaldi, G., Taylor, M. a. P., and Yue, W. L., “Using Artificial Neural Network in Passenger Trip Distribution Modelling (A Case Study in Padang, Indonesia)”, *Proc. East. Asia Soc. Transp. Stud.*, **2009**, pp. 105–105 (2009).
29. Hyland, M., Frei, C., Frei, A., and Mahmassani, H. S., “Riders on the storm: Exploring weather and seasonality effects on commute mode choice in Chicago”, *Travel Behav. Soc.*, **13**, pp. 44–60 (2018).
30. Dong, X., Wang, L., and Hu, B., “Analysis of spatio-temporal distribution characteristics of passenger travel behaviour based on online ride-sharing trajectory data”, *J. Phys. Conf. Ser.*, **1187**(5), p. 052055 (2019).
31. Regehr, J. D., Montufar, J., and Hernandez-Vega, H., “Traffic Pattern Groups Based on Hourly Traffic Variations in Urban Areas”, *J. Transp. Inst. Transp. Eng.*, **7**(1), pp. 1–16 (2015).

32. Fujita, M., Yamada, S., and Murakami, S., “Time Coefficient Estimation for Hourly Origin-Destination Demand from Observed Link Flow Based on Semidynamic Traffic Assignment”, *J. Adv. Transp.*, **2017** (2017).
33. Venkatanarayana, R., Smith, B. L., and Demetsky, M. J., “Quantum-Frequency Algorithm for Automated Identification of Traffic Patterns”, *Transp. Res. Rec.*, **2024**(1), pp. 8–17 (2007).
34. Yu, Q., Xie, Y., Li, W., Zhang, H., Liu, X., Shang, W.-L., Chen, J., Yang, D., and Yan, J., “GPS data in urban bicycle-sharing: Dynamic electric fence planning with assessment of resource-saving and potential energy consumption increasement”, *Appl. Energy*, **322**, p. 119533 (2022).
35. Shang, W.-L., Chen, J., Bi, H., Sui, Y., Chen, Y., and Yu, H., “Impacts of COVID-19 pandemic on user behaviors and environmental benefits of bike sharing: A big-data analysis”, *Appl. Energy*, **285**, p. 116429 (2021).
36. Shang, W.-L., Chen, Y., and Ochieng, W. Y., “Resilience Analysis of Transport Networks by Combining Variable Message Signs With Agent-Based Day-to-Day Dynamic Learning”, *IEEE Access*, **8**, pp. 104458–104468 (2020).
37. Shang, W., Han, K., Ochieng, W., and Angeloudis, P., “Agent-based day-to-day traffic network model with information percolation”, *Transp. Transp. Sci.*, **13**(1), pp. 38–66 (2017).
38. Deri, J. A., Franchetti, F., and Moura, J. M. F., “Big data computation of taxi movement in New York City”, *2016 IEEE Int. Conf. Big Data Big Data*, pp. 2616–2625 (2016).
39. Freire, J., Bessa, A., Chirigati, F., Vo, H., and Zhao, K., “Exploring What not to Clean in Urban Data: A Study Using New York City Taxi Trips”, *IEEE Data Eng. Bull.*, **39**(2), pp. 63–77 (2016).
40. Patel, U., *NYC Taxi Trip and Fare Data Analytics Using BigData* (2015).
41. Martínez, L. M., Viegas, J. M., and Silva, E. A., “A traffic analysis zone definition: a new methodology and algorithm”, *Transportation*, **36**(5), pp. 581–599 (2009).
42. Rousseeuw, P. J. and Hubert, M., “Robust statistics for outlier detection”, *WIREs Data Min. Knowl. Discov.*, **1**(1), pp. 73–79 (2011).
43. Shanker, M., Hu, M. Y., and Hung, M. S., “Effect of data standardization on neural network training”, *Omega*, **24**(4), pp. 385–397 (1996).
44. LeCun, Y., Bengio, Y., and Hinton, G., “Deep learning”, *Nature*, **521**(7553), pp. 436–444 (2015).
45. Nwankpa, C., Ijomah, W., Gachagan, A., and Marshall, S., “Activation Functions: Comparison of trends in Practice and Research for Deep Learning” (2018).
46. Taxi, N. Y. and (TLC), L. C., “New York City Green Taxi Trip Data”, Available: https://www.nyc.gov/site/tlc/about/tlc-trip-record-data.page (2018).
47. Department of City Planning’s Neighborhood Tabulation Areas (NTAs), N. Y. C. D., “NYC Taxi Zones”, Available: https://catalog.data.gov/dataset/nyc-taxi-zones/resource/79e158e7-d158-4ace-b9d3-41ede473c76c (2019).
48. (NCDC), N. C. D. C., “NYC 2018 Hourly Surface Data”, Available: https://www.ncei.noaa.gov/products/land-based-station/integrated-surface-database (2018).
49. Tang, J., Liu, F., Wang, Y., and Wang, H., “Uncovering urban human mobility from large scale taxi GPS data”, *Phys. Stat. Mech. Its Appl.*, **438**, pp. 140–153 (2015).

50. Office of the New York State Comptroller, O., "An Economic Snapshot of the East Harlem Neighborhood", Available: https://www.osc.state.ny.us/files/reports/osdc/pdf/report-9-2018.pdf (2018).
51. Office of City Planning, N. Y. C. D., "New York City's Zoning & Land Use Map", p. 5, Available: https://zola.planning.nyc.gov/ (2020).
52. Chollet, F., "Keras", Available: https://github.com/fchollet/keras (2015).
53. Wilson, A. G., "Advances and problems in distribution modelling", *Transp. Res.*, **4**(1), pp. 1–18 (1970).
54. Duffus, L. N., Sule Alfa, A., and Soliman, A. H., "The reliability of using the gravity model for forecasting trip distribution", *Transportation*, **14**(3), pp. 175–192 (1987).
55. Celik, H. M., "Sample size needed for calibrating trip distribution and behavior of the gravity model", *J. Transp. Geogr.*, **18**(1), pp. 183–190 (2010).
56. Willmott, C. J., "On the Validation of Models", *Phys. Geogr.*, **2**(2), pp. 184–194 (1981).
57. Schemper, M., "Predictive accuracy and explained variation", *Stat. Med.*, **22**(14), pp. 2299–2308 (2003).

**Captions**

## Figures and Tables

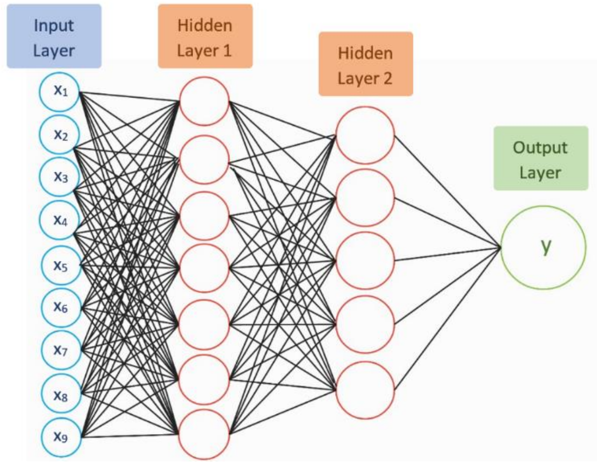


Figure 2. Structure of the Neural Network in this study

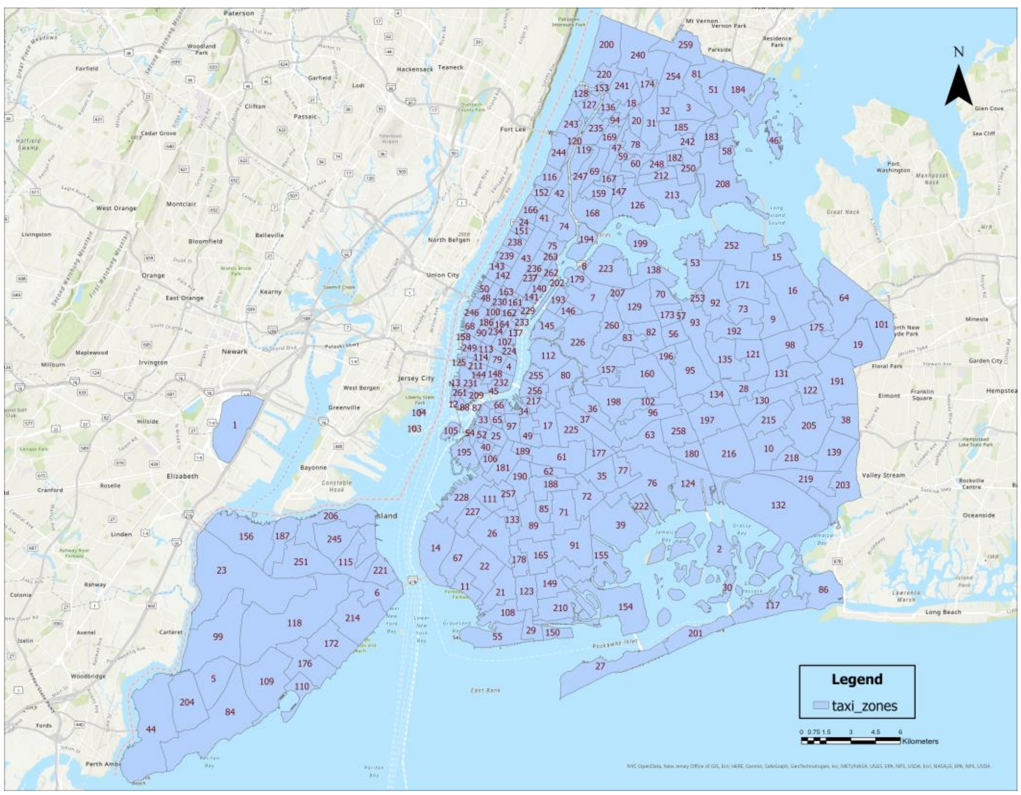


Figure 2. New York City taxi zones

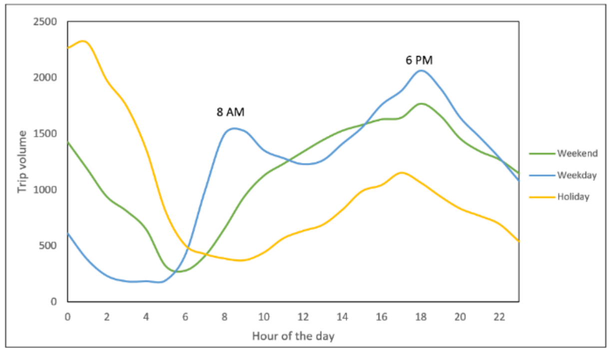


Figure 3. Weekday, Weekend and Holiday hourly trip patterns
Holiday trip data as of 1st January 2018, Weekend and Weekday trip data are average trip counts of all weekends and all weekdays of February 2018 respectively.

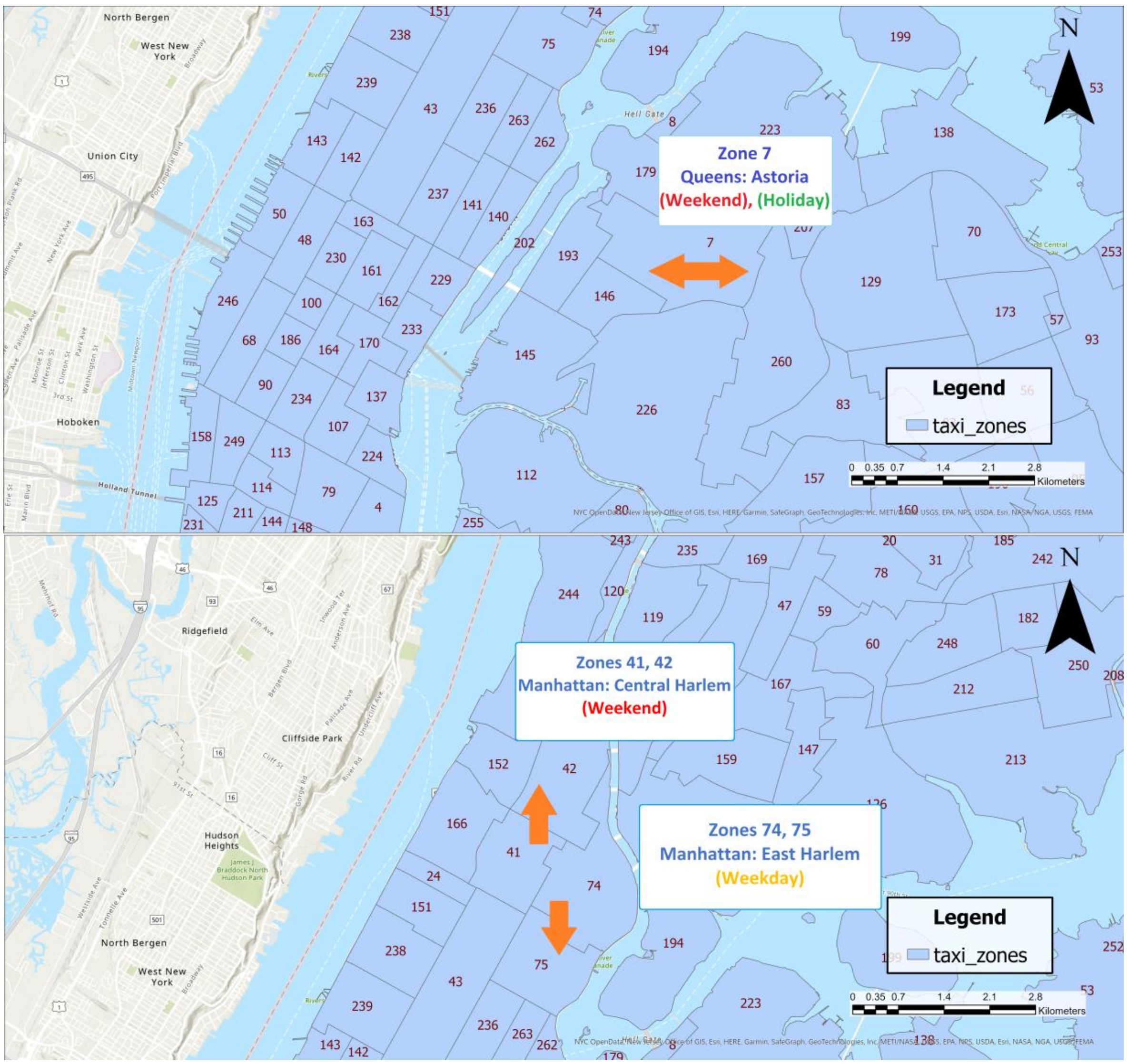


Figure 4. Paths with highest trip counts

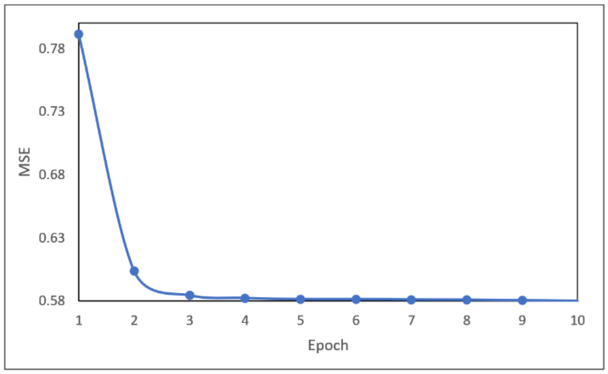


Figure 5. Optimum loss values in each epoch

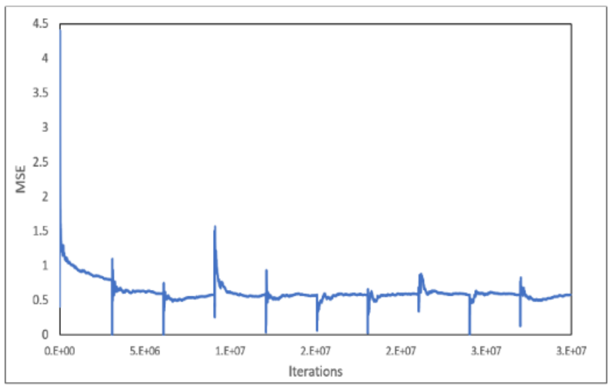


Figure 6. Loss values in each iteration

Table 8 - Trip record fields sample

| **Field** | **Pickup Date Time** | **Dropoff Date Time** | **PU Location ID** | **DO Location ID** | **Passenger Count** | **Trip Distance** |
|---|---|---|---|---|---|---|
| **Description** | Passenger pick-up date and time | Passenger drop-off date and time | Destination zone ID | Origin zone ID | Number of passengers | Distance traveled from origin to destination |
| **Sample** | 01/01/2018 12:18:50 AM | 01/01/2018 12:24:39 AM | 43 | 75 | 2 | 3.5 mi |

Table 9 – New York City zones dataset sample

| **Zone ID** | **12** | **46** | **94** | **165** |
|---|---|---|---|---|
| **Location** | Manhattan-Battery Park | Manhattan-Chinatown | The Bronx-Fordham South | Brooklyn-Midwood |

Table 10 - Weather record sample

| Field | YR--MODAHRMN | DIR | SPD | SKC | TEMP | PCP01 | SD |
|---|---|---|---|---|---|---|---|
| **Description** | Year-Month-Day-Hour-Minute in GMT | Wind direction in compass degrees | Wind speed (mph) | Sky cover (Nominal classification) | Temperature in Fahrenheit | 1-hour liquid precipitation in inches | Snow depth in inches |
| **Sample** | 201801281151 | 990 | 6 | SCT | 49 | 0.01 | 0.0 |

Table 11. Network parameters values

| Parameter | Learning rate | Decay | Batch size | Number of epochs |
|---|---|---|---|---|
| **Value** | 0.0005 | $1*10^{-6}$ | 256 | 10 |

Table 12. Test results on one million random samples

| | Total Trips | Number of zero values | R Squared | MSE |
|---|---|---|---|---|
| **Predicted Values** | 14721 | 991384 | 0.453 | 0.0348 |
| **Actual Values** | 15127 | 991324 | | |

Table 13 - Trip generation linear regression results

| | Total Trips | | R Squared |
|---|---|---|---|
| | **Predicted** | **Actual** | |
| **Zones Productions** | 692585 | 693364 | 0.9921 |
| **Zones Attractions** | 692565 | 693364 | 0.9922 |

Table 14 - The Gravity model results

| Power Variable (n) | Number of zero values | | R Squared | MSE | Iterations |
|---|---|---|---|---|---|
| | **Actual** | **Predicted** | | | |
| **1** | 19112 | 18057 | 0.5933 | 6098.768 | 9 |
| **2** | | 17780 | 0.8439 | 2340.261 | 11 |
| **3** | | 16584 | 0.7714 | 3248.935 | 13 |
| **4** | | 13502 | 0.4755 | 7866.803 | 16 |
| **5** | | 10164 | 0.1162 | 13254.1566 | 19 |
| **6** | | 7950 | -0.2248 | 18370.005 | 22 |