Transactions on Fuzzy Sets and Systems (TFSS)
URL: https://tfss.journals.iau.ir/
Online ISSN: 2821-0131
Vol.2, No.1, (2023), 101-132
DOI: http://doi.org/10.30495/tfss.2023.1971760.1053

Article type: Original Research Article

Evolving Fuzzy Systems in Taxi Demand Forecasting and Classification

Luis Fernando Linhares, **Alisson Marques Silva***

Abstract. This work presents an approach to taxi demand forecasting and classification. The proposed approach uses historical data from taxi rides and meteorological data. The Kruskal-Wallis variable ranking method is used to identify the most relevant variables. The selected variables are used as input to an evolving fuzzy system to perform the prediction. Once the forecast is made, the demand results are classified by value ranges. Those ranges are also identified by colors that compose a heatmap, displayed at each time interval. In this work, to perform the prediction, four evolving systems are evaluated: Autonomous Learning Multi-Model (ALMMo); evolving Multivariable Gaussian Fuzzy Modeling System (eMG); evolving Fuzzy with Multivariable Gaussian Participatory Learning and Recursive Maximum Correntropy eFCE and; evolving Neo-Fuzzy Neuron (eNFN). Computational experiments were carried out to evaluate the evolving systems in predicting Pick-Up and Drop-Off, at intervals of 15 and 30 minutes, for 86 zones in New York, covering the period from 01/01/2018 to 31/ 10/2018. The results obtained by the evolving systems are compared with each other and state of the art. Among the evolving models, ALMMo presented the best results compared to the state of the art and other evolving models. Performance obtained by the evolving models suggests that the proposed approach is promising an alternative to forecasting and classifying passenger demand.

AMS Subject Classification 2020: 03E72; 68T05; 68T27
Keywords and Phrases: Forecast, Classification, Fuzzy Systems, Evolving Systems, Taxi Demand.

1 Introduction

Problems related to urban mobility in large centers have constantly been increasing. The popularization of private vehicles, added to the inefficiency and precariousness of public transport (buses, subways, etc.), has contributed to the increase in urban traffic, negatively impacting the lives of millions of people [13]. One of the main challenges for transport agencies in these cities is managing strategies to mitigate these problems [59]. In this context, taxi rides are viable alternatives to traditional public transport. They provide personalized services and easy access to passengers, allowing a more comfortable journey and less dependence on pre-established schedules [19]. However, a pertinent and constant problem for this transport system still needs to be resolved: passenger supply and demand.

Passenger supply and demand lead to the following problem: reducing an existing gap between the pool of available vehicles (taxis) and passenger demand in a given region and time [38]. While taxi drivers spend time looking for passengers, some passengers do not find taxis available, increasing customer dissatisfaction with the services provided and the excessive fuel expenditure in taxi vehicles [37]. However, once the demand of their customers is known in advance, taxi companies can logistically organize their fleet, increasing or decreasing their offer, reducing the waiting time of passengers, reducing the idle time of taxi drivers without

*Corresponding author: Alisson Marques Silva, Email: alissonmarques@gmail.com, ORCID: 0000-0002-1023-6514
Received: 31 October 2022; **Revised**: 4 January 2023; **Accepted**: 4 January 2023; **Published Online**: 7 May 2023.

How to cite: L. F. Linhares and A. M. Silva, Evolving Fuzzy Systems in Taxi Demand Forecasting and Classification, *Trans. Fuzzy Sets Syst.*, 2(1) (2023), 101-132.

customers, and serving more passengers, and consequently, obtaining higher profits [52]. In addition, the cost of fuel, vehicle depreciation, and time in transit also decrease, reducing the costs of a taxi fleet[54].

According to [37], it is possible to extract knowledge from several available data sources. Using urban traffic data is the primary way to aggregate demand information. Taxi companies' applications can use information from the tracking technology known as Global Positioning System (GPS) to obtain information such as boarding location, average vehicle speed, trajectory and time traveled, and disembarkation location, among others. Several companies use these technologies to control their fleet and have consistent records. With this, through computational techniques, it is possible to transform this data into useful information, which will serve as support to understand the flow of demand in a given region.

Several machine learning techniques have been used to solve or minimize the problem of passenger supply and demand. Autoregressive Integrated Moving Average (ARIMA) [37, 36], Convolutional Neural Networks (CNN) [64, 67], Long Short-Term Memory (LSTM) [21, 55], and Support Vector Machine (SVM) [19, 58] are examples of some of the main techniques used. Most of these techniques use offline learning to train and validate their models. Offline learning, also known as a batch, represents the training of traditional machine learning models done by samples accumulated over a period of time. That is, their training is done in batches. According to [69], the main characteristics of this type of learning include:

- needs to have the entire dataset previously available;
- the training of the models is done using several iterations in the data to find the best set of weights and/or parameters;
- systems, once trained, are no longer updated.

In offline learning, the dataset is usually divided into training and testing for this type of learning. Training data is run in batches on models to update parameters. The test data are executed with the model already trained to evaluate its performance [11]. In some approaches, the division of data sets is done by training, validation, and testing, with validation being a necessary period to assess whether the model training was satisfactory.

However, these techniques, for the most part, have certain limitations inherent to their nature, namely: they need periodic revisions of the models, training them and adapting them again to the reality of the data; they do not deal with situations in which there are changes in environmental conditions, typical of online configurations with non-stationary data and; they rarely manage to handle complex systems where there are multiple modes of operation [4]. These limitations create methodological gaps that make it challenging to predict taxi demands accurately. This is mainly because demand is simultaneously related to several exogenous factors with high volatility. Temperature, time of day, important events in the region, and traffic are some examples of these factors [31]. Given this context, it is desirable that the techniques used to forecast taxi demand have continuous and incremental learning, adapting to external factors present in traffic. Additionally, these systems must preferably supply the restrictions inherent to the previously mentioned traditional techniques.

The concept of online learning deals with data processing in a continuous flow. Therefore, the models that use this learning process are the samples incrementally, as they are presented. Furthermore, the training of these models is constant. That is, there is no interruption in learning while new data samples are available [45]. The processing of continuous data streams brings some characteristics inherent to its nature, namely:

- samples arrive continuously, and data streams can be infinite [48];
- systems have no control over how or when data will arrive [47];
- after processing, a sample should ideally be discarded to avoid scalability issues [16].

Online learning is also advantageous in computational terms, as it uses low memory, processor, and disk resources [45]. Evolving Fuzzy Systems (EFS) are a class of algorithms that use this type of learning. EFS can simultaneously adapt its structure and parameters as new samples are made available [42]. This type of system is suitable for situations where data is non-stationary, arrives as a datastream, and the environment is subject to change.

Taxi demands have characteristics that identify them as a data stream, as the ride information is generated uninterruptedly and may undergo changes in its pattern due to an accident, road works, weather, etc. Therefore, the application of EFS is feasible for the problem of forecasting and classifying passenger demand.

In this context, this work aims to present an approach to forecasting the demand for taxi passengers. In this approach, the forecast is performed by an evolving fuzzy system using historical information from taxi rides and weather data as input. Computational experiments evaluate four evolving systems forecasting Pick-Up and Drop-Off. In addition to the forecast, the classification of taxi demands is performed considering four classes.

The rest of the paper is organized as follows. Section 2 presents the concepts of evolving fuzzy systems, and the models used in this work are described. The preliminary concepts necessary for a better understanding of this work are presented in Section 3. Section 4 introduces a literature review of works focused on taxi demand. The proposed approach is introduced and detailed in Section 5. The computational experiments and their respective results and analyzes are presented in Section 6. Finally, Section 7 illustrates the final considerations and proposals for future studies.

2 Evolving Fuzzy Systems - EFS

Evolving intelligent systems can be seen as a combination of systems with expandable structures (in this specific case, fuzzy rules) and online machine learning methods [1, 32]. EFS is a type of evolving intelligent system which shares the following characteristics:

- its structure is not fixed, that is, it naturally expands and retracts as the system evolves [24];
- its parameters are continuously adapted as the system evolves [2];
- its operation is uninterrupted, allowing learning in online mode and, if necessary, in real-time [42].

Due to the online and incremental learning, the EFS demonstrate fast data processing since the samples are processed and discarded soon after they are discarded [32]. This feature allows the execution of massive data sets without incurring high computational costs, which would not occur in offline [20] processing models. Thus, EFS are a viable alternative for uninterrupted data processing and continuous learning.

Once the concepts of evolving fuzzy systems have been discussed, the subsequent sections deal with the four systems used as tools in this work: ALMMo [5], eMG [25], eNFN [46], and eFCE [41].

2.1 ALMMo - Autonomous Learning Multi-Model System

ALMMo, proposed in [5], is a multi-model evolving fuzzy system with fuzzy rules of the AnYa [3] type. The antecedent of the AnYa rules is based on the concept of data clouds. The principle of data clouds is similar to that of clustering algorithms, that is, each cloud represents a set of samples with similar characteristics. The shape of each cloud is formed by the data that constitute it, and focal points represent its centers. Each new sample is classified in a data cloud, and this classification is performed using an empirical analysis method (EDA) [6]. The EDA assigns, for each sample, a discrete density level based on the distance of this sample from the others, classifying them closer or further away from the focal points of each cloud.

ALMMo fuzzy rules are as follows:

$$\text{IF}\left(j^* = \underset{i=1,2,\ldots,G}{\text{argmin}}\left(\|x_t - c_t^i\|\right)\right)\text{THEN}\left(\Xi_{j*} \leftarrow x_t\right),$$

where $x_t = [x_1, x_2, \ldots, x_E]^T$ a data sample in Euclidean space defined as $\boldsymbol{R}^M$, *argmin* the method that searches for the nearest data cloud, Ξ_i the ith data cloud and c_t^i its corresponding focal point. The number of data clouds is represented by G in the observed universe $\boldsymbol{R}^M$, which also represents the number of rules.

To change its structure, ALMMo evolves its data clouds, modifying, creating, or deleting them as new samples are presented to the model. At each new sample, the scalar average of the products $\overline{X}_t$ and the global average μ_t at the instant t are computed recursively by

$$\overline{X}_t = \tfrac{t-1}{t}\overline{X}_{t-1} + \tfrac{1}{t}\|x_t\|^2, \tag{1}$$

and

$$\mu_t = \tfrac{t-1}{t}\mu_{t-1} + \tfrac{1}{t}x_t. \tag{2}$$

Subsequently, the unimodal density $\overline{D}$ is calculated between the current sample x_t and all identified focal points of the i-th class by

$$\overline{D}_t(x_t) = \frac{1}{1+\frac{|x_t-\mu_t\|^2}{\overline{X}_t-\|\mu_t\|^2}}. \tag{3}$$

ALMMo checks whether to create, merge, modify or delete rules for each new sample x in time t presented to the algorithm. Condition 1 verifies whether a new rule should be created.

$$\begin{aligned}\textit{Condition 1: IF }&\left(\overline{D}_{t+1}(x_{t+1}) > \max_{i=1,2,\ldots,G_t}\left(\overline{D}_{t+1}\left(c_t^i\right)\right)\right)\\ \textit{OR }&\left(\overline{D}_{t+1}(x_{t+1}) < \min_{i=1,2,\ldots,G_t}\left(\overline{D}_{t+1}\left(c_t^i\right)\right)\right).\end{aligned}$$

If Condition 1 is not accepted, the sample is assigned to the closest data cloud that has its parameters updated. Otherwise, a new rule is created with the focal point at x_t, and Condition 2 is checked. This condition tests whether the created cloud overlaps with an existing cloud.

$$\textit{Condition 2}: \quad IF \quad (\overline{D}_{t+1,i} \geq (\frac{1}{1+n^2})).$$

If Condition 2 is satisfied, the overlapping cloud will be replaced by the newly created one, and the new cloud will inherit the consequent parameters of the overlapping cloud.

Subsequently, the quality of the rule base is verified by a utility measure, aiming to exclude rules that are little activated. This measure is calculated based on the accumulated sum of the rule's contribution to the calculation of the output from the moment of its creation to the current sample, that is, it is the measure of the importance of a given fuzzy rule concerning other rules ($i = 1, 2, ..., G_{t+1}$). Its calculation is obtained by

$$\eta_{t+1}^{i} = \frac{1}{t+1-activated^{i}} \sum_{l=activated^{i}}^{t+1} \delta_{l}^{i}, \tag{4}$$

where $activated^i$ represents the time the rule/data cloud was activated and δ_l^i is the activation level of the ith rule/data cloud at the instant l. The activation level δ_l^i is calculated for each rule $(i = 1, 2, ..., G_l)$ in:

$$\delta_{l}^{i} = \frac{\overline{D}_{l}^{i}(x_l)}{\sum_{j=1}^{G_l} \overline{D}_{l}^{i}(x_l)}. \tag{5}$$

The variable $\eta 0$ is a constant that defines a threshold for excluding rules with a low activation rate. A rule is excluded by Condition 3. If Condition 3 it's met, the jth rule/data cloud is excluded, along with the consequent parameters.

$$Condition\ 3: \ IF \quad (\eta_t^j + 1 < \eta 0).$$

In addition to excluding rules, ALMMo develops a method to exclude variables with a high correlation index to avoid redundancy, improve processing time and improve the algorithm's overall performance.

2.2 eMG - evolving Gaussian Multivariable Fuzzy System

eMG [25] is an evolving fuzzy system based on first-order Takagi-Sugeno rules. Its rule base is built through a clustering algorithm with participatory learning. Participatory learning refers to using what has already been learned to verify the impact of including a new sample in the model, that is, a new sample is evaluated based on the knowledge already acquired [56].

The antecedent of the rules is represented by multivariable Gaussian membership functions H, defined by

$$H(x) = exp\left[-\frac{1}{2}(x-c)\Sigma^{-1}(x-c)^{T}\right], \tag{6}$$

where x is an input vector, Σ is a positive-definitive symmetric matrix, c is a vector with modal values (centers) and represents an element present in $H(x)$ and Σ represents the dispersion level of $H(x)$. Both c and Σ are parameters that are part of the membership function and are associated with the center and dispersion of the function, respectively [25].

eMG evolves its structure by creating, updating, or merging clusters. The model output is obtained by the weighted average of the contributions of each rule. Each rule belongs to a cluster, and for each new sample presented to the model, eMG updates its structure through a compatibility measure calculated per cluster. The compatibility measure $p_t^i \in [0, 1]$ is calculated by the distance between the current input x_t and the centers of the existing clusters c_t^i, that is, p_t^i is obtained by

$$p_t^i = \exp\left[-\frac{1}{2}M\left(x_t, c_t^i\right)\right], \tag{7}$$

where the distance M can be calculated by

$$M\left(x_t, c_t^i\right) = \left(x_t - c_t^i\right)\left(\sum_i^t\right)^{-1}\left(x_t - c_t^i\right)^T, \tag{8}$$

where x_t is the current sample and c_t^i the center of the i-th cluster. An alert threshold is defined for the compatibility measure T_p, which is obtained by

$$T_p = \exp\left[-\frac{1}{2}\chi^2_{m,\alpha}\right], \tag{9}$$

where $\chi^2_{m,a}$ is a $Chi-Square$ distribution with m degrees of freedom and α the one-sided confidence interval. After calculating the compatibility measure and the new sample for all clusters, the index of the cluster with the highest degree of compatibility i^* is chosen.

An alert mechanism is used to find clusters whose structure does not correctly represent the current knowledge of the system. Therefore, this cluster must undergo a review [25]. For each new sample x_t inserted into the model, the alert index $a_t^i \in [0,1]$ is calculated for all clusters and estimated through the cumulative probability of V_t, in

$$a_t^i = \overline{p}(V_t < \overline{z}), \tag{10}$$

where $\overline{p}(V_t = \overline{z})$ is a binomial distribution. A threshold of the alert index T_a is also defined, which is used in creating clusters. The threshold T_a is calculated by

$$T_a = 1 - \frac{\alpha}{\omega}, \tag{11}$$

where α is a parameter that defines the significance level and ω is the observation window for calculating the alert index.

A cluster is created when the compatibility measure p_t^i is less than the compatibility threshold T_p for all clusters and the alert index for the cluster with the highest degree of compatibility $a_t^{i^*}$ is greater than its respective threshold T_a. In other words, a cluster will be created if

$$p_t^i < T_p, \forall i = 1, \quad , c_i^t \text{ and } a_t^{i^*} > T_a \text{ for } i^* = max_i(p_t^i).$$

Otherwise, if

$$p_t^i > T_p, \forall i = 1, \quad , c_t^i \text{ and } a_t^{i^*} < T_a \text{ for } i^* = max_i(p_t^i),$$

the sample is inserted into the cluster with the most similarity. Soon after, the center of this cluster is updated in

$$c_{t+1}^{i^*} = c_t^{i^*} + \lambda(p_t^{i^*})^{1-a_t^{i^*}}(x_t - c_t^{i^*}), \tag{12}$$

where $\lambda \in [0,1]$ represents the learning rate. The eMG consequent parameters are updated by a weighted least squares recursive algorithm.

2.3 eNFN - evolving Neo-Fuzzy Neuron

The eNFN [46] is an evolving fuzzy system built upon a structure composed of a set zero-order Takagi-Sugeno models, one for each input variable. The eNFN structure uses triangular and complementary membership functions and evolves based on the modeling error calculated recursively. A gradient descent algorithm with an optimal learning rate updates the rules' consequent parameters.

The learning algorithm starts with two membership functions for each input variable. With each new sample, new functions can be created and included using the global average error of the model and the average local error of the most active membership function. The mean value $\hat{\mu}_t$ and the global error variance $\hat{\sigma}_t^2$ are recursively computed by

$$\hat{\mu}_t = \hat{\mu}_{t-1} - \beta\left(\hat{\mu}_{t-1} - e_t\right), \tag{13}$$

and

$$\hat{\sigma}_t^2 = (1-\beta)(\hat{\sigma}_{t-1}^2 + \beta(\hat{\mu}_t - e_t)^2), \tag{14}$$

for input x_t at time t, where error in t is defined by $e_t = y_t - \hat{y}_t$ and β is the learning rate.

The average value of the local error to the most active membership function $\mu_{b_{ti}^*}$ is calculated recursively by

$$\mu_{b_{ti}^*} = \mu_{b_{ti-1}^*} - \beta(\mu_{b_{ti-1}^*} - e_t). \tag{15}$$

A threshold τ is used to limit the number of rules, avoiding complex models and overfitting. This threshold τ is compared to the smallest distance ($dist$) allowed between the modal value of the function to be created and adjacent functions. A new membership function will be created and included if

$$\mu_{b_{ti}^*} > \hat{\mu}_t + \hat{\sigma}_t^2 \text{ and } dist > \tau. \tag{16}$$

The procedure to the exclusion of membership functions is based on the concept of age [33]. A membership function will be excluded if it remains inactive for a long time. The age of a j-th membership function is calculated by

$$age_j = t - activated_j, \tag{17}$$

in which t is the current time interval and $activated_j$ represents the interval of activation of the j-th membership function. Let b_i^- be the index of the least active membership function at time t. Then, the membership function indexed by b_i^- will be excluded if $b_i^- > \omega$, where ω is a parameter that indicates the time limit for deleting a function.

2.4 eFCE - evolving Fuzzy with Multivariable Gaussian Participatory Learning and Recursive Maximum Correntropy

The eFCE [41], as the eMG, is an evolving system that builds its structure by including, merging, or excluding clusters and rules based on a recursive clustering algorithm with participatory learning and multivariable Gaussian membership functions. The clustering algorithm uses the compatibility measure computed recursively to add a new cluster. The compatibility measure is calculated by Euclidean and Mahalanobis distances. The age and population concepts are used to exclude inactive clusters and rules, and the merge procedure is based on the remarkably overlapping of a cluster pair. The consequent parameters are updated based on the Recursive Maximum Correntropy.

The distance measure is defined as follows:

- If $n_t^i < N_{max}$, uses the Euclidean distance computed by:

$$D = (x_t - \mu_t^i)^T I_{mxm}(x_t - \mu_t^i). \tag{18}$$

- If $n_t^t > N_{max}$, uses the Mahalanobis distance obtained by

$$D = (x_t - \mu_t^i)^T (\sum)^{-1} (x_t - \mu_t^i), \tag{19}$$

in which n_t^i is the number of samples from the i-th cluster, N_{max} is the limit for creating microclusters or clusters, I_{mxm} is an identity matrix m x m, $\sum$ is a scatter matrix (Mahalanobis distance) or an identity matrix (Euclidean distance).

The cluster's age defines the inactivity time of the cluster, and the population represents the number of samples attributed to a cluster. Thus, the least active cluster (indexed by $\bar{i}$) is found for each sample. The cluster indexed by $\bar{i}$ is excluded if

$$age_t^{\bar{i}} > \omega \ \ and \ \ \frac{n_t^{\bar{i}}}{t} < 0.01.$$

in which t is the number of samples.

In the merge procedure, if two clusters i^* and i have the norm of the difference in distance between their centers, μ_{i^*} and μ_i, less than or equal to a predefined threshold, then they are merged, i.e.,

$$||\mu_t^{i^*} - \mu_t^i|| \leq \rho \ \ (i = 1...c_t \ and \ i \neq i^*), \tag{20}$$

in which ρ is the threshold for merging clusters. The center of the new cluster is defined by the weighted average as follows:

$$\mu_t^{i^* \cup i} = \mu_t^{i^*} - \frac{n_t^i}{n_t^{i^*} + n_t^i} (\mu_t^{i^*} - \mu_t^i), \tag{21}$$

in which n is the number of samples of the clusters. The new cluster depends on the number of samples of the merged clusters. The dispersion matrix of the resulting cluster is updated to the average value of the dispersion matrices of the merged clusters by

$$\sum_t^{i^* \cup i} = \frac{\sum_t^{i^*} + \sum_t^i}{2}. \tag{22}$$

3 Preliminary Concepts

This section introduces some concepts and terminology necessary to understand the context of passenger demand forecasting. First, Section 3.1 presents the definition of passenger demand forecast and illustrates some terminology on the subject. Then, Section 3.2 describes the main techniques used to divide regions. Next, the relationship between passenger demand forecasting and the classification process is introduced in Section 3.3. Finally, the concepts of heatmaps are detailed in Section 3.3.

3.1 Passenger Demand Forecasting

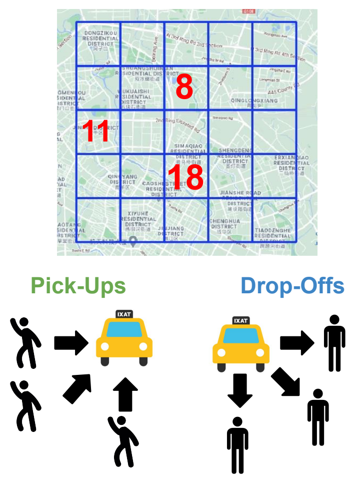
Passenger demand forecasting is considered a time series problem [64, 67, 49]. A time series can be described as a set of observations organized sequentially over time. Assuming that we have a set of observed variables $(x_t, x_{t-1}, x_{t-2}, ..., x_{t-P})$ of P past observations up to the instant t, we want to predict the next value (x_{t+1}) [11]. To understand how demand forecasting is performed, it is first necessary to introduce some terminologies:

- **Taxi Zone**: a taxi zone can be defined as the partition of a L region into smaller K regions, defined as zones. Each zone is defined by an index z, where $1 \leq z \leq K$;

- **Time Interval**: continuous time partitioned into identical sequential intervals. Consider the observed period of 30 days and an interval of 60 minutes. This period would then represent 720 (24 x 30) observations;

- **Taxi Ride**: a taxi ride represents an event where the driver pick-up the passenger in a certain region and at a certain time (Pick-Up) and drops him off at his destination also in a certain region and time (Drop-Off) [67];

- **Taxi Passenger Demand**: literature works, such as [67, 21, 8], classify passenger demands as the sum of Pick-Ups and/or Drop-Offs in a given time interval t, in a given zone z. Figure 1 shows an example of the demand for Pick-Up and Drop-Off for taxis in zones 8, 11, and 18 between 08:00 and 9:00 am.



08:00 to 09:00		
Regions	Pick-Up	Drop-Off
8	30	20
11	10	50
18	80	30

Figure 1: Pick-Up and Drop-Off demand in a certain region and time.

3.2 Region Division Techniques

Efficiently dividing a region on the map can improve the accuracy of predictor models by shaping specific formats for each region based on regional empirical studies. Some works in the literature, such as those by [64, 67, 51] perform the division as follows: choosing a given region L, delimiting its maximum latitudes and longitudes, and dividing it into K similar polygons, rectangles, or hexagons being the commonly used shapes. A set of latitudes and longitudes defines each polygon threshold. Therefore, a taxi zone z can be described as L_z, where $1 \leq z \leq K$. You can find several tools to assist in the process of delimiting and dividing zones, such as *QGIS*[2], *ArcGIS*[3] and the *GeoPandas*[4] library.

Other division forms include identifying high-demand density centers through clustering algorithms such as K-Means. These densities are determined with delimitation techniques, such as the Voronoi diagrams [14]. The work of [38] proposes an evolving origin and destination matrix, dividing the city of Porto, Portugal, into regions shaped by the density of taxi requests. The algorithm that generates these regions works in an evolving way, shaping the format of each region in the matrix according to the seasonality of the city.

[2]https://qgis.org/
[3]https://www.arcgis.com
[4]https://geopandas.org

Additionally, the selection of tourist sites (Stadiums, Museums, etc.) is also used to delimit the regions [40, 50]. For this, a predefined distance radius is determined starting from the central latitude/longitude of the location. There is also the spatial division by predefined neighborhoods or irregular shapes, for example, by the city hall [67]. Figure 2 illustrates three techniques for dividing zones, namely: (A) division by polygons; (B) division by Voronoi diagrams, and; (C) irregular division in New York City (United States of America).

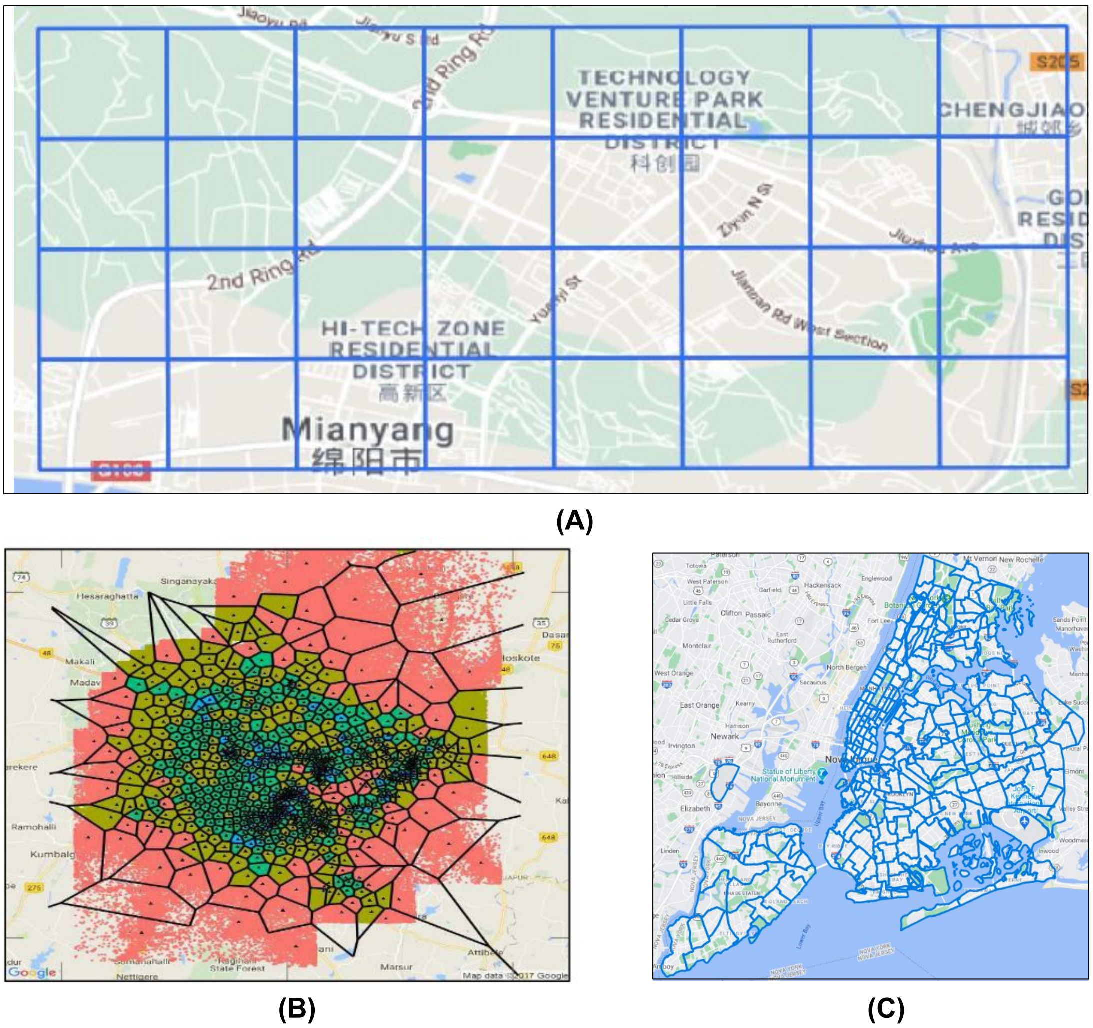


Figure 2: Examples of zone divisions: (A) rectangular division in Mianyiang City (China); (B) division based on Voronoi diagrams and; (C) division into irregular zones in New York City (United States of America). Source: Figure (B) adapted from [14].

3.3 Forecasting and Classification

Introduced the aforementioned concepts, given a number P of observations of past demands (D_t, D_{t-1}, D_{t-2}, D_{t-3}, ..., D_{t-P}), the objective is to forecast passenger demand F steps ahead (D_{t+F}). The demand forecast can be set for short periods (short-term) and long periods (long-term). The number of steps ahead in which the prediction is performed differentiates a demand from short to long periods. Long-term demands increase the step window, predicting F demands ahead, where $F > 1$. In the short-term, demands ahead are defined as $F = 1$, that is, one step ahead [64].

Classification is one of the most common tasks in machine learning. The classification consists of recognizing and categorizing patterns, given certain characteristics that are predetermined in different classes [39]. In the context of passenger demand, one of the ways to classify demand is to define a range of values in which the discrete value of demand fits. Each range would represent a class, which nomenclatures would define, thus becoming a multi-class problem [61].

3.4 Heatmaps

Heatmaps are two-dimensional graphical representations of data where colors represent the values of a given variable. Its way of presentation minimizes the amount of learning needed to understand it. For example, in a heatmap, it is easy to identify that the proportion of a specific color is relative to the level of the represented variable. In addition, heatmaps show information directly about the stimulus, making its visualization and interpretation easy [9].

Choropleth heatmaps are maps generally used to represent information in a given geographic region. The main difference between the choropleth maps and conventional ones is the limit of representation of colors in the image. While traditional heatmaps represent colors without a boundary, choropleth maps present information in colors limited by a predefined format, whether irregular or not [7]. Among its applications, we can mention: traffic flow; vehicle routes and density; population density; spread of disease; to name just a few [12]. Figure 3 illustrates the example of a choropleth heatmap, representing the number of taxi requests from 263 predefined regions of the boroughs of New York City. The black circles show the regions with the highest density of requests.

4 Literature Review

This section presents a summary of the main works related to the context of passenger supply and demand. First, Section 4.1 discusses studies that used consolidated statistical techniques for time series and historical data of the races to forecast demand. Next, in Section 4.2, works that use exogenous variables (time of day, weather conditions, regional events, etc.) to help improve the results produced by the models are presented. Then, in Section 4.3, approaches that extract knowledge through maps (spatial) combined with historical data of races (temporal) are discussed. Finally, Section 4.4 presents a compendium of the 30 main works researched in this study and their main objectives.

4.1 Forecast with Temporal Analysis

To address the problem of passenger supply and demand works in the literature focus on understanding and possible modeling solutions. Data mining techniques are used to create knowledge from diverse databases. Initial results indicated that time slots better distributed the quantification of passengers and that stratifying a city or region into micro-regions (zones) helped to understand the peculiarities in each [26]. In addition, the influence of traffic at certain times of the day, weekends, and days of the week were identified as influential factors in taxi drivers' strategies to get passengers [65].

With the advancement of research, the concepts of aggregation of passenger requests by time interval are consolidated, and new approaches emerge to perform demand forecasting. For example, in [36], a hybrid statistical model is created to predict passenger demand for the next 30 minutes at 63 booths in the city of Porto (Portugal). Additionally, the model of [36] is modified in [37] to simulate the demand forecast in real-time. Furthermore, [61] forecasts taxi demand in Stockholm city (Sweden) using Artificial Neural Networks. Finally, in [19], models based on Support Vector Machine (SVM) are created to forecast demands in the next 5, 10, and 15 minutes in China.

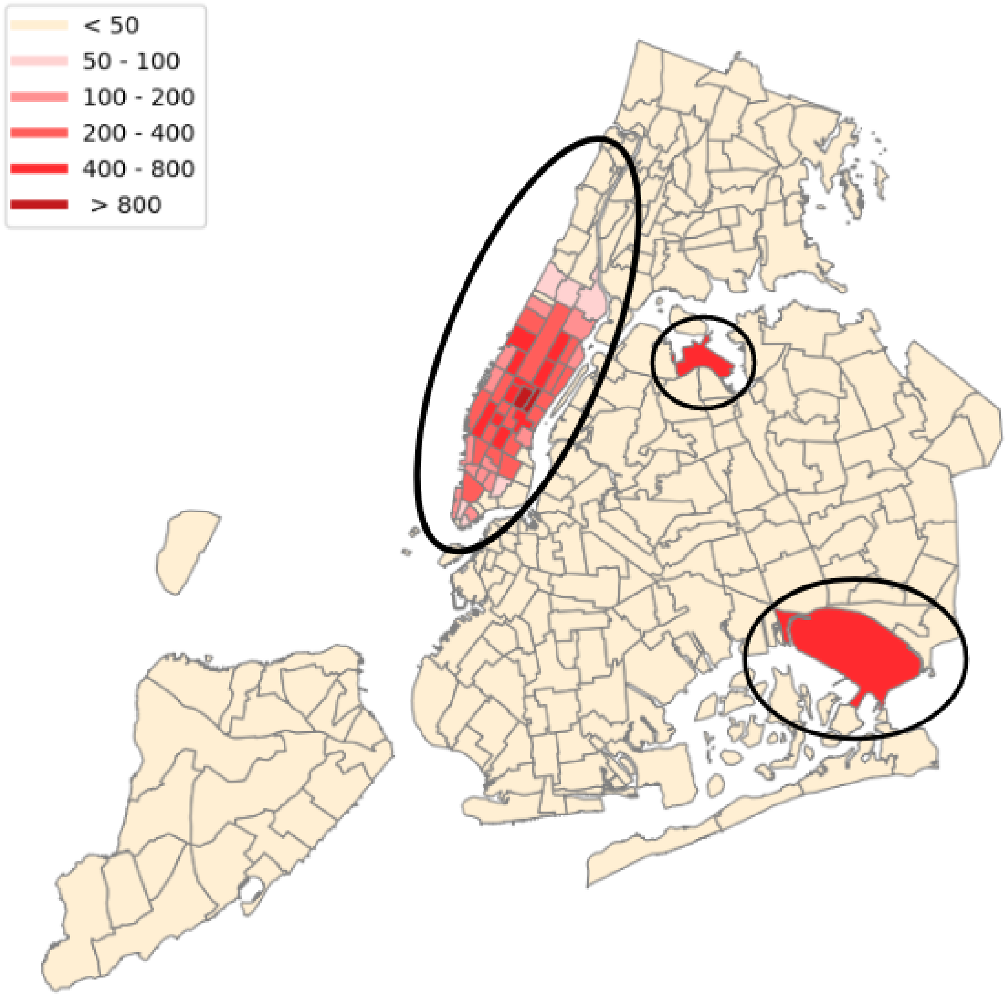


Figure 3: Example heatmap of taxi requests in New York.

Intuitively, it is assumed that the demand forecast aims to predict the adequate number of Pick-Up at a given time. However, it was found that understanding the pattern of Drop-Off also adds knowledge. Once it is known how many taxis have arrived in a given region, it is possible to know in advance the number of vehicles that will be available, allowing taxi companies to dispatch them to places where services are needed [57]. Therefore, it is understood that both Pick-Up and Drop-Off predictions are relevant.

4.2 Forecast with Inclusion of Exogenous Variables

In some works, it is possible to find authors researching how external factors affect the result of demand. Temperature, weather conditions, air humidity, and time of day are examples of these factors [21]. In the work of [27], information about wind speed, amount of snow, and temperature are used to forecast demand in New York (United States of America), testing the performance of two models: Fusion Convolutional Long Short-Term Memory Network (FCL-Net) and Spatio-Temporal Residual Network (ST-ResNet). In [55], different combinations of characteristics (weather data, Pick-Up, Drop-Off, day of the week) are tested to forecast demand within a 60-minute interval. A model dubbed Multi-Level Recurrent Neural Networks (MLRNN) based on Long Short-Term Memory (LSTM) is the subject of work by [66]. The model receives, as input, meteorological information and historical data to forecast demands in the next 30 minutes in New York.

Other authors analyze the impact of points of great interest (stadiums, hospitals, etc.) and events (concerts, presentations, games) on the region's traffic. For example, the study by [35] gathers information from events held in two popular areas of New York (Terminal 5 and Barclays Center). It includes these events as input variables in their models, performing demand prediction in these two areas. Still in this context, other approaches explore the forecast of taxis in regions of great interest, such as airports, hospitals, tourist sites, subway, etc. [50]. [30] created a model dubbed Context-Aware Attention-Based Convolutional Recurrent Neural Network (CACRNN), combining points of interest (Points Of Interest - POI) with meteorological data to forecast demands in the next 15 minutes in Chengdu (China) and New York. In [68] studies, a framework is proposed combining Multilayer Perceptron (MLP) and Long Short-Term Memory (LSTM) to capture the temporal dependencies of runs. At the same time, convolutional layers are used to interpret event data and convert it into useful information to forecast demand in New York in two centers where major events occur in the city (Terminal 5 and Barclays Center). In [29], a model is developed with the combination of Random Forest and Ridge Regression techniques to predict passenger demand at points of great interest in Xian (China) for the interval of 1 hour.

In the literature, it is possible to find works that gather customer information through mobile applications. These applications inform you by GPS triangulation of your location. These data are used as input to increase the accuracy of the models. For example, in [18], an experiment is conducted using data from mobile customer applications to forecast demand for the next 30 minutes in the city of Tokyo (Japan). Taxi drivers who participated in the experiment achieved an average profit of 3.9% higher than those who did not.

4.3 Combinations of Spatial and Temporal Techniques for Forecasting

Some researchers suggest creating models that unify temporal and spatial information, determined as Spatio-Temporal models. Map analysis (spatial) combined with historical ride data (temporal) creates more robust hybrid models capable of predicting passenger demand more accurately. Some models, such as [59] and [31], aggregate taxi demand into images converted into heatmaps and process them with Convolutional Neural Networks (CNN). CNN's output is input to a second model that predicts passenger demand. The work by [27] also presents a similar proposal, aggregating map analysis, meteorological data, and historical information in a framework to test various models in the literature to forecast demand in New York. The work of [28] uses weather data, moving averages, and historical data from races to feed a model that uses three *deep learning* techniques: Convolutional Neural Networks (CNN), Long Short-Term Memory (LSTM), and Gated Recurrent Unit (GRU). This study predicts demands on points of interest (site of tourist attractions, hospitals, etc.) in the city of Kaohsiung (Taiwan). Other works share approaches similar to those mentioned above, such as those by [15] and [22].

The relationship between regions is also part of studies in the literature. Closer regions tend to share similar demand patterns. For example, [64] proposes a feature selector named Spatio-Temporal Dynamic Time Warping - ST-DTW, based on the Euclidean distance between the zones. This method is compared to other feature selectors available in the literature. [51] proposes adding weights to each region, in which regions closer to each other gain greater weight and, consequently, are classified differently by the model. Unlike other works, the demand forecast in this approach is done using an origin-destination tuple, that is, the number of rides that left a given region X and had their destination in a given region Y. On the other hand, the study by [8] measures similarities between zones by analyzing user preferences. Therefore, places with similar themes tend to have similar demand patterns. In [34], a variable selector based on a statistical method known as Augmented Dickey-Fuller (ADF) is used. After selection, the Multi-Task Deep Learning (MTDL) model forecasts demand in the next 10 minutes in New York.

A combination of classification and regression models can be seen in the work of [67], which presents two approaches: one based on classification and the other on time series prediction. The authors understand that passenger demand can be transformed into a classification problem. Each demand is classified by a label (demand from 00:00 to 02:00 is classified as 1, the demand from 02:00 to 04:00 as 2, etc.). This classifier is an input variable to a regression model that predicts taxi demands at 30-minute intervals in New York City.

Data from competing companies can help to increase the accuracy of the models. This can be seen in the study by [62], in which they proposed a *framework* named Unified Spatial-Temporal Network (USTN), using maps processed in convolutional layers, together with historical data from Uber [5] and information from New York yellow taxis to forecast demand at 1-hour intervals.

4.4 Summary of Works

In a complementary way, this section summarizes 30 of the main studies analyzed in this work related to forecasting passenger demand. Table 1 presents the authors, the technologies used, the main objectives, and the cities where the forecast was made.

Table 1: Abstracts of literature works.

Autor	Models e Techniques	Objectives	City
[21]	FCL-net	Demand forecast - 1-hour intervals	Hangzhou - China
[49]	LinUOTD	Demand forecast - 1-hour intervals	Beijing and Hangzhou - China
[57]	Poisson Regression	Identify demand and supply patterns	New York - USA
[14]	Voronoi Tesselation, *Geohash* e *K-Means*	Optimize demand forecast	New York - USA
[55]	LSTM	Demand forecast - 10, 20 and 60-minutes intervals	New York - USA
[35]	Gaussian Process	Demand forecast - 1-hour intervals	New York - USA
[27]	*Residual Neural Networks* - ResNet	Demand forecast - 1-hour intervals	New York - USA
[50]	Combined Model with GRU	Demand forecast - 1-hour intervals	Bangkok - Thailand
[59]	LSTM and CNN	Demand forecast - 30-minute intervals	Guangzhou - China
[18]	*Autoencoders*	Demand forecast 30-minute intervals	Tokyo - Japan
[19]	SVM	Demand forecast 10-minute intervals	City not informed - China
[8]	CRNN	Demand forecast 1-hour intervals	Shenyang - China
[51]	GCN	Demand forecast (origin-destination) - 1-hour intervals	Chengdu and Beijing - China
[63]	ST-Ann - *Encoders e Decoders*	Demand forecast - 30 and 60-minute intervals	Beijing - China and New York - USA
[60]	DBSCAN	Demand forecast - 1-hour intervals	Beijing - China
[31]	Convolutional LSTM	Demand forecast - 30-minute intervals	New York - USA
[15]	Vector Auto Regression and Least Absolute Shrinkage Selection Operator	Demand forecast - 15-minute intervals	New York - USA
[43]	LSTM	Trajectory forecast of possible taxi clients	San Francisco/New York - USA and Porto - Portugal
[64]	RNN and Dynamic Time Warping	Demand forecast - 15-minute intervals	Chengdu - China and New York - USA
[22]	TBI2Flow	Passenger and traffic forecast - 30-minute interval	Shanghai, China
[53]	None	Survey - traffic and demand forecast	None
[67]	LSTM	Demand forecast - 30-minute intervals	New York - USA
[17]	LSTM and Fuzzy Neural Networks	Traffic forecast = 5-minute intervals	San Diego - USA and Beijing - China
[62]	USTN	Demand forecast - 1-hour intervals	New York - USA
[30]	Context-Aware Attention-Based and CACRNN	Demand forecast - 15-minute intervals	Chengdu - China and New York - USA
[29]	Random Forest and Ridge Regression	Demand forecast - 1-hour intervals	Xian - China
[34]	*Multi-Task Deep Learning*	Demand forecast - 10-minute intervals	New York - USA
[66]	MLRNN	Demand forecast 30-minute intervals	New York - USA
[68]	MLP and LSTM	Demand forecast - 30-minute intervals	New York - USA
[28]	CNN and LSTM	Demand forecast - 30-minute intervals	Kaohsiung - Taiwan

5 Proposed Approach

This section details the approach to forecasting and classifying taxi passenger demand. First, the method to demand forecast is detailed in Section 5.1. Next, Section 5.2 describes the methodology for classifying demand.

5.1 Demand Forecasting

The demand forecast process consists of 4 steps, as shown in Figure 4. Obtaining and extracting the rides database is the first step described in Section 5.1.1. Later, Section 5.1.2 details the input variables and how

[5]Private ride company (https://www.uber.com).

the dataset is constructed. Next, the feature selection is illustrated in Section 5.1.3. Finally, Section 5.1.4 shows how an evolving fuzzy system makes forecasting.

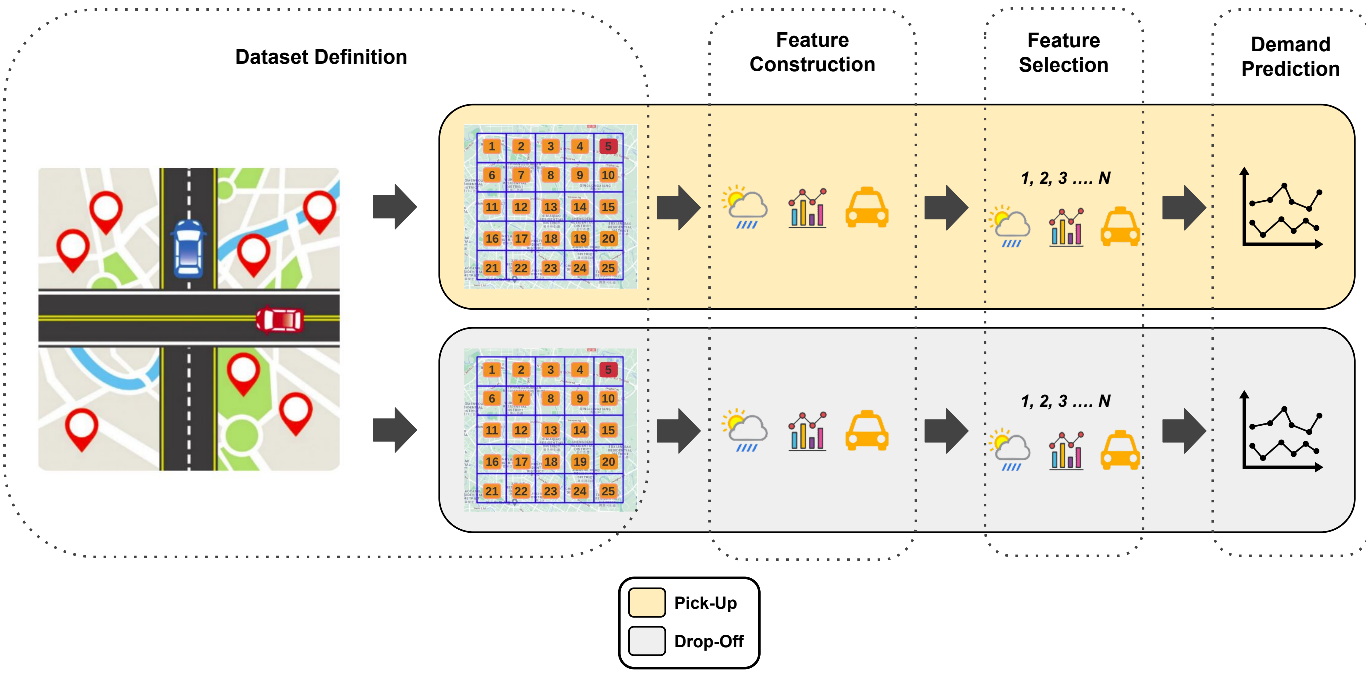


Figure 4: Steps of the approach for demand forecasting.

5.1.1 Dataset Definition

The extraction of relevant information that will serve as input for the evolving models starts with the rides database. These bases are data from the neighborhood, district, or city rides. In these bases, the following information is usually found: date and time of Pick-Up, date and time of Drop-Off, and Pick-Up and Drop-Off regions. Additional information can also be found on these bases, such as the amount charged, number of passengers, probability of cancellation, payment method, etc.

Once the ride's base is determined, the study region is chosen. This region is then delimited and divided into smaller regions, defined as zones (z). This division aims to frame the study region in predefined limits and cluster the ride locations. A summary of the main region division techniques available in the literature is detailed in Section 3.2. After delimiting the region into zones, the locations of Pick-Up and Drop-Off are identified and converted into their respective zone.

After the zones have been delimited and the locations converted, the next step is the generation of datasets. A dataset for Pick-Up and another for Drop-Off is built in each of the k zones, delimiting each demand D for a taxi by time interval t. For a better understanding of the concepts mentioned above, see Section 3.1.

5.1.2 Feature Construction

As described in Section 3.1, passenger demand forecasting is considered a time series problem. The concept of time series can be defined by a set of observations generated sequentially over time [10]. In this work, the values passed to be used to model the series are characterized by the following:

- **Historical Demand of Intervals (*dhi*)**: taxi demand of the last 8 time intervals. The historical demand of the z zone intervals is represented by

$$dhi_{z,t} = dhi_{z,t-1}, dhi_{z,t-2}, ..., dhi_{z,t-8}. \tag{23}$$

- **Historical Demand of the Days (*dhd*)**: demand from the previous 8 days. The historical demand for days in the z zone can be described as

$$dhd_{z,tr} = dhd_{z,tr-1}, dhd_{z,tr-2}, ..., dhd_{z,tr-8}, \tag{24}$$

  for the same relative time interval, where tr represents the previous day's relative time interval of the demand to be forecast.

- **Average Historical Demand Intervals (*mdhi*)**: the simple average of the last 8 lags of historical demand (dhi). The $mdhi_z$ is obtained for the z zone by

$$mdhi_{z,t} = 1/8 \sum_{i=1}^{8} dhi_{z,t-i}. \tag{25}$$

- **Average of Historical Demand of Days (*mdhd*)**: simple average of the last 8 lags of previous days' historical demand (dhd_z). Its calculation is represented as

$$mdhd_{z,tr} = 1/8 \sum_{i=1}^{8} dhd_{z,tr-i}. \tag{26}$$

Then, 18 variables extracted from historical values of rides (series lags) are generated. Furthermore, using information extracted from taxi rides, the following 3 variables are obtained by

- **Day of Week (*ds*)**: Sunday (1), Monday (2), ..., Saturday (7).
- **Weekend (*fs*)**: day of week (0) weekend (1).
- **Time of Day (*hd*)**: divided into three periods: rest time (0), normal time (1), and peak time (2). Rest time is from 20:00 to 06:00. Peak hours include the following periods: [06:00 - 09:00, 12:00 - 15:00, 17:00 - 20:00]. The remaining times of the day are determined as normal times.

The exogenous variables are obtained from meteorological information, as suggested in [21, 49]. The meteorological data were extracted from the platform *Wheater Underground*[6]. On this platform, meteorological data are made available every hour, being inserted into the dataset in the interval before the demand is forecast. The following list details the 5 variables used in this work:

- **Temperature (*te*)**: measure in Fahrenheit.
- **Relative humidity (*ur*)**: measure in percentage.
- **Precipitacion (*pr*)**: measurement in millimeters.
- **Wind speed (*vv*)**: measurement in miles per hour (mph).

[6] https://www.wunderground.com/

- **Time Condition** ($\boldsymbol{ct}$): textual description that was represented by indices of 1 (*Fog*) to 50 (*Windy*).

Each data set has 26 input variables, 21 extracted from information in the rides database, and 5 exogenous variables obtained from meteorological information. Thus, at the end of this stage, each zone has a set of data containing the indexes of the intervals, the 26 variables, and their respective taxi demands.

5.1.3 Feature Selection

The next step after constructing the datasets is the selection of input variables. This step aims to identify which variables are the most relevant among the 26 created in Section 5.1.2. In this work, Kruskal-Wallis [23] is used, a statistical method that compares variables among themselves and ranks them by their degree of relevance.

Once sorted, the most relevant N variables that will be part of the new datasets are chosen. As the Kruskal-Wallis method only ranks the variables in order of relevance, the value of N must be found by an alternative method or empirically. Finally, datasets for each zone are built with the most relevant N variables and the respective taxi demands.

5.1.4 Demand forecasting

In this approach, the taxi demand forecast is done by an evolving fuzzy system. This forecast is made for $t+1$, that is, for the next time interval. The forecast is made by zone for each set of Pick-Up and/or Drop-Off, using the N variables selected in the selection step as input.

5.2 Demand Classification

The demand classification process is illustrated in Figure 5. The steps for building the dataset are, *mutatis mutandis*, the same as described in Section 5.1, so it will not be necessary to describe them again. Next, Section 5.2.1 describes the definition of classes. Then, Section 5.2.2 deals with how demand is classified. In this work, demand classification is performed using the outputs obtained in the forecast, transforming the forecasting task into a classification problem, as in [44]. Finally, Section 5.2.3 describes the generation of heatmaps.

5.2.1 Classes Definition

To start the classification process, the number of classes is defined. In this work, four classes of demand were defined: (i) Very Low; (ii) Low; (iii) Medium, and; (iv) High. Then, the ranges of values that will identify each class are determined. These ranges are limited to their upper and lower limit. Finally, the demands of all zones are used to identify the boundaries by class. For simplification purposes, in this work, the boundaries of all classes are always multiples of 5. The limits identification can be performed with the aid of a histogram.

5.2.2 Demand Classification

In this step, the transformation of the prediction task into a classification problem is performed. First, the values predicted by the evolving model are treated and converted into one of the classes. In other words, it checks which lower/upper bound (domain) the predicted value ($\hat{y}_t$) fits. The conversion is carried out as follows:

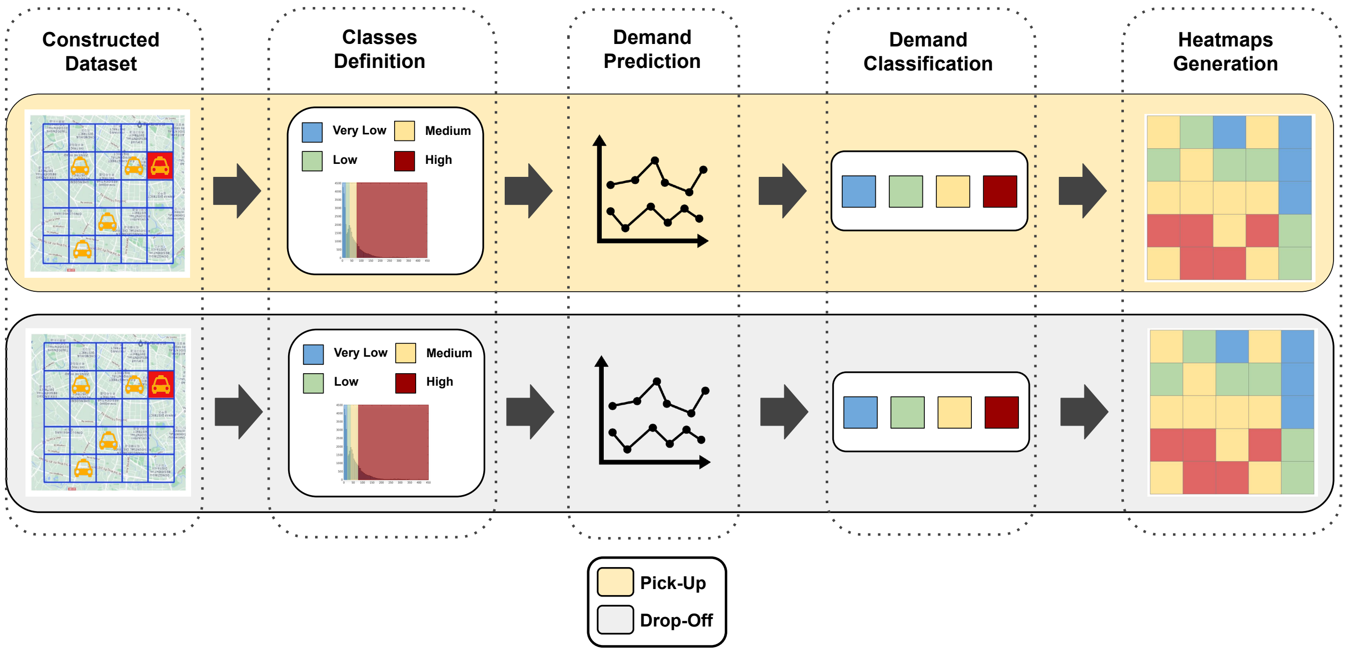


Figure 5: Steps for classification.

$$Class = \begin{cases} Very\ Low, & \text{If } 0 \leq \hat{y}_t \leq UB_{\text{Very Low}}, \\ Low, & \text{If } LB_{\text{Low}} \leq \hat{y}_t \leq UP_{\text{Low}}, \\ Medium & \text{If } LB_{\text{Medium}} \leq \hat{y}_t \leq UP_{\text{Medium}}, \\ High & \text{If } LB_{\text{High}} \leq \hat{y}_t \leq UP_{\text{High}}, \end{cases} \tag{27}$$

where LB and UP are the lower and upper bounds of the class, respectively.

5.2.3 Heatmaps generation

In this last step, the demands already converted into classes will be part of a heatmap (for more details on heatmaps, see Section 3.4). The heatmap will identify each zone by its geographic format and color, with the zones with higher demands represented by warmer colors and those with lower demands by cooler colors.

6 Experiments and Results

This section presents the computational experiments to evaluate the proposed approach in forecast and classification contexts, both for Pick-Up and Drop-Off demands. In this study, we chose to use a dataset derived from the running database of the city of New York. The database is provided by the NYC Taxi and Limousine Commission[7]. The experiments were conducted in a subset of 86 zones (out of 263) from 01/01/2018 to 10/31/2018. The datasets were divided into 60% (01/01/2018 to 06/30/2018) for selecting variables by Kruskal-Wallis and defining the classes. The base's other 40% (07/01/2018 to 10/31/2018) was used to validate the models. The datasets were normalized between 0 and 1. To compare results, the settings described above were the same used in work by [67]. In their work, Zhang et al. [67] evaluated demand

[7] Available at: https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page

forecasting performance by comparing 5 models with offline learning: ARIMA, MLP, sLSTM, mLSTM, and pmLSTM. The best result was obtained by pmlLSTM, which will be used as a benchmark.

The time intervals used were 15 and 30 minutes, as in [37, 36, 64, 67]. To carry out the experiments, 4 datasets were constructed for each z zone, namely: (i) 15-minute Pick-Up; (ii) 30-minute Pick-Up; (iii) 15-minute Drop-Off and; (iv) 30-minute Drop-Off. The feature selection was conducted by Kruskal-Wallis, considering the 5, 10, and 20 most relevant variables. The dataset with 26 variables is also considered.

In carrying out the forecasting experiments, the evolving models ALMMo [5], eMG [25], eFCE [41], and eNFN [46] were used. The four models are deterministic; their results do not change with each new iteration. Therefore, each algorithm was executed only once. The processing of the samples was carried out online; that is, the models evolved their structure and parameters for all samples. It is noteworthy that no procedure was performed for fine-tuning the models' hyperparameters. It was decided to use the same parameters in all experiments to maintain the generalist approach. The hyperparameter values for each evolving model followed the metric limits defined by their respective authors, listed as follows:

- **ALMMo:** $forgettingfactor = 0{,}1$; $densitythreshold = 0{,}8$; $\omega = 10$.
- **eMG:** $\alpha = 0{,}01$; $\lambda = 0{,}05$; $w = 40$, $\Sigma = 10^{-1}$.
- **eFCE:** $\alpha = 0.01$; $\lambda = 0.05$; $w = 40$, $\Sigma = 10^{-1}$, $Nmax = 3$, $\rho = 0.12$.
- **eNFN:** $\eta = 10$; $\beta = 0{,}01$; $w = 100$.

To measure the performance of the proposed approach, forecasting experiments were performed and compared with alternative approaches through the RMSE (Root Mean Square Error) [64, 67]. The results are presented as the average of the errors of the zones. As for the classification experiments, Accuracy was used, a metric already consolidated to evaluate this type of problem.

Section 6.1 deals with the results obtained in the prediction task. Then, Section 6.2 presents the results of demand classification and heatmaps generation. Finally, Section 6.3 presents a detailed analysis of the input variables selected by the Kruskal-Wallis method.

6.1 Demand Forecasting

Table 2 demonstrates the performance of predictors for the 15-minute interval. The table presents the best results obtained by each of the models, with the best performance obtained by ALMMo (with 20 input variables) followed by eFCE (with 10 input variables), eMG (with 10 input variables), and eNFN (with 5 input variables). Figure 6 illustrates the forecast by ALMMo (with 20 input variables) for the 15-minute interval from 10/24/2018 to 10/30/2018 in Times Square (zone 230).

Table 2: Performance in forecasting Pick-Up and Drop-Off demand for the 15-minute interval.

Model	**Pick-Up**	**Drop-Off**
	RMSE	**RMSE**
ALMMo {N = 20}	**13.329**	**12.460**
eFCE {N = 10}	14.592	13.887
eMG {N = 10}	14.793	15.466
eNFN {N = 5}	15.464	14.580
pmlLSTM [67][8]	——	——

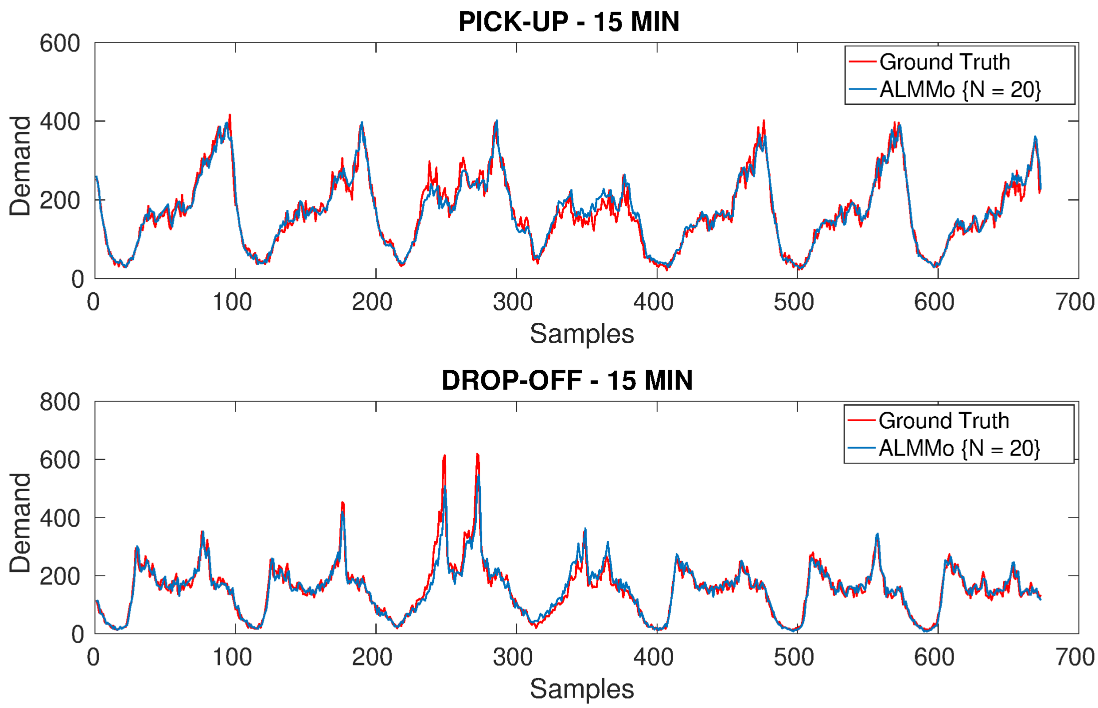


Figure 6: ALMMo forecasts (with 20 input variables) for Times Square (Zone 230), in 15-minute intervals, for the last 7 days (10/24/2018 to 10/30/2018).

Table 3 describes the best results for the 30-minute interval. ALMMo (with 20 variables) got the smallest error for Pick-Up. For Drop-Off, the best results were obtained by pmlLSTM. The ALMMo (with 20 variables) proved to be competitive, presenting a difference of only 3% concerning the Drop-Off results obtained by pmlLSTM. Then follows the eFCE (with 10 variables), eMG (with 10 variables), and eNFN (with 5 variables). Figure 7 illustrates the forecast by ALMMo (with 20 input variables), in the interval of 30 minutes, for the last 7 days in Times Square (Zone 230).

6.2 Demand Classification

Table 4 illustrates, for Pick-Up and Drop-Off, the range of values of each class for the 15 and 30-minute intervals in New York, where D represents the demand and the number of samples in each class is defined in

[8] Em Zhang et al. [67] there were no experiments that performed the demand forecast with an interval of 15 minutes.

Table 3: Performance in forecasting Pick-Up and Drop-Off demand for the 30-minute interval.

Model	Pick-Up	Drop-Off
	RMSE	RMSE
ALMMo {N = 20}	**25.072**	22.957
pmlLSTM [67]	26.311	**22.411**
eFCE {N = 10}	27.150	25.461
eMG {N = 10}	28.512	26.473
eNFN {N = 5}	29.017	25.358

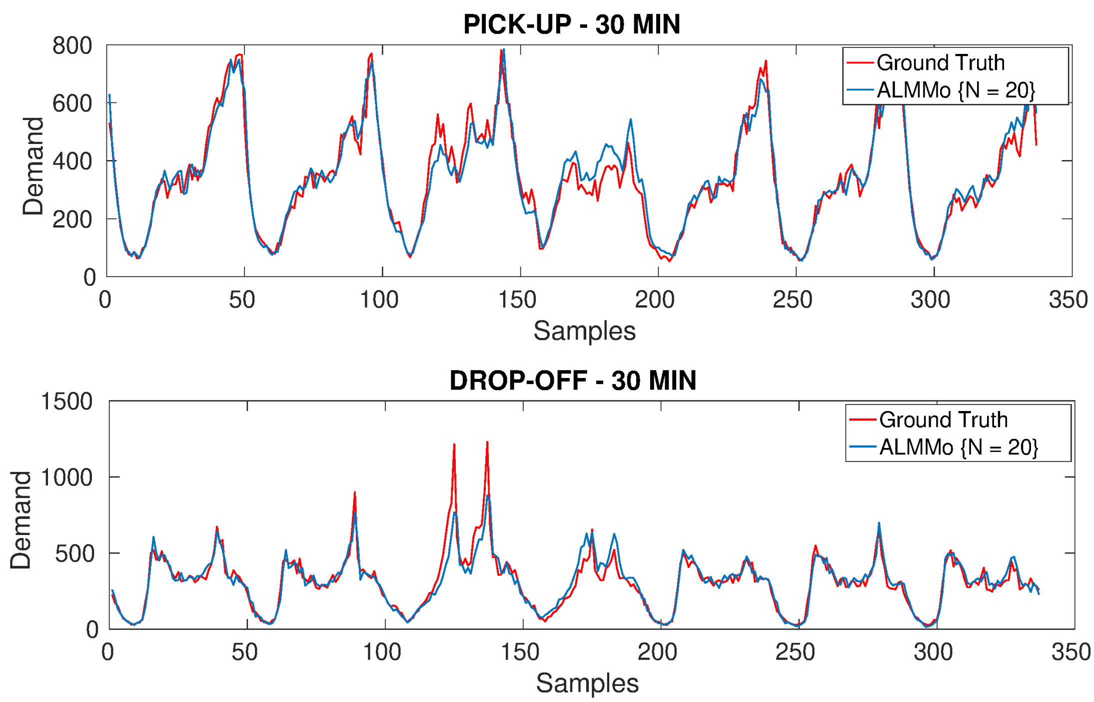


Figure 7: ALMMo forecasts (with 20 input variables) in Times Square (Zone 230), in 30-minute intervals, for the last 7 days (10/24/2018 to 10/30/2018).

parentheses. The histograms in Figure 8 graphically represent the data described in Table 4.

Table 5 shows the best accuracy and standard deviation of the evolving models for New York in the interval of 15 minutes. The results suggest that the best accuracy was obtained by ALMMo (with 20 input variables), followed by eFCE (with 10 input variables), eMG (with 10 input variables), and eNFN (with 5 input variables).

To represent the classification performance, Figure 9 presents the heatmap for Pick-Up and Drop-Off, with 15-minute intervals in New York with 86 zones, from 10/31/2018, covering the interval from 12:00 to 12:15. In 9A and 9C is shown the map generated by the classification performed by ALMMo (with 20 input variables). Figures 9B and 9D show the desired heatmap for the same period.

The best performance of accuracy and standard deviation of the evolving models for New York, in the interval of 30 minutes, is illustrated in Table 6. ALMMo (with 20 input variables) was the model that obtained the best results, followed by eFCE (with 10 input variables), eMG (with 10 input variables), and

Table 4: Classifications for New York.

Type	Interval	Ranges and Quantity			
		Very Low	Low	Medium	High
PU	15	$D \leq 25$ (287283)	$25 > D \leq 50$ (318128)	$50 > D \leq 100$ (389045)	$D > 100$ (433832)
PU	30	$D \leq 60$ (173844)	$60 > D \leq 120$ (182183)	$120 > D \leq 240$ (185552)	$D > 240$ (172565)
DO	15	$D \leq 25$ (313466)	$25 > D \leq 50$ (300727)	$50 > D \leq 100$ (391842)	$D > 100$ (422253)
DO	30	$D \leq 60$ (185395)	$60 > D \leq 120$ (169884)	$120 > D \leq 240$ (196772)	$D > 240$ (162093)

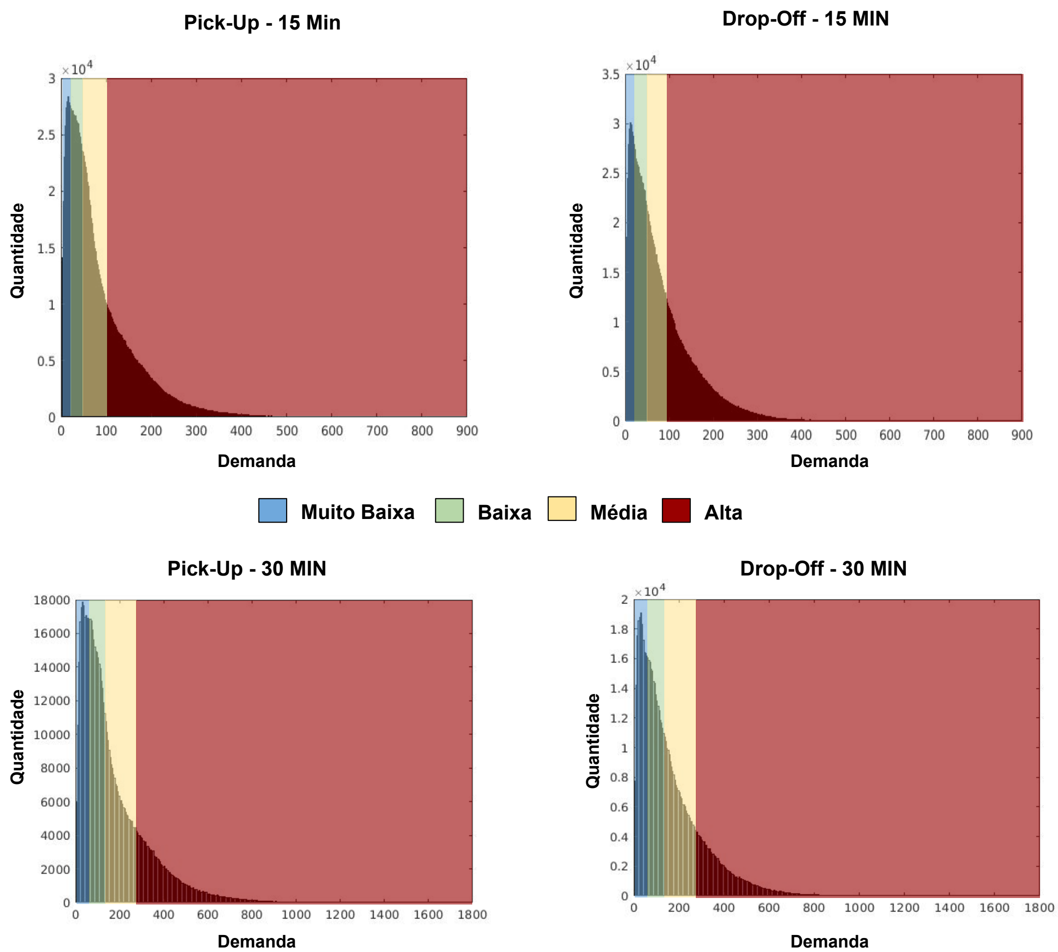


Figure 8: Histograms with the ranges of values in New York.

eNFN (with 5 input variables).

Figure 10 represents the heatmap for Pick-Up and Drop-Off, with 30-minute intervals in New York with 86 zones, on 10/31/2018, covering the 12:00 interval until 12:30. In 10A and 10C, the value predicted by ALMMo is shown (with 20 input variables). Figures 10B and 10D show the desired heatmap for the same

Table 5: Accuracy for the 15-minute interval.

Modelo	**Pick-Up**		**Drop-Off**	
	Accuracy (%)	**± STD**	**Accuracy (%)**	**± STD**
ALMMo {N = 20}	**84.796**	**4.334**	**84.865**	**4.593**
eFCE {N = 10}	83.431	4.493	83.197	4.882
eMG {N = 10}	83.368	4.426	82.790	5.137
eNFN {N = 5}	81.950	5.101	81.824	5.373

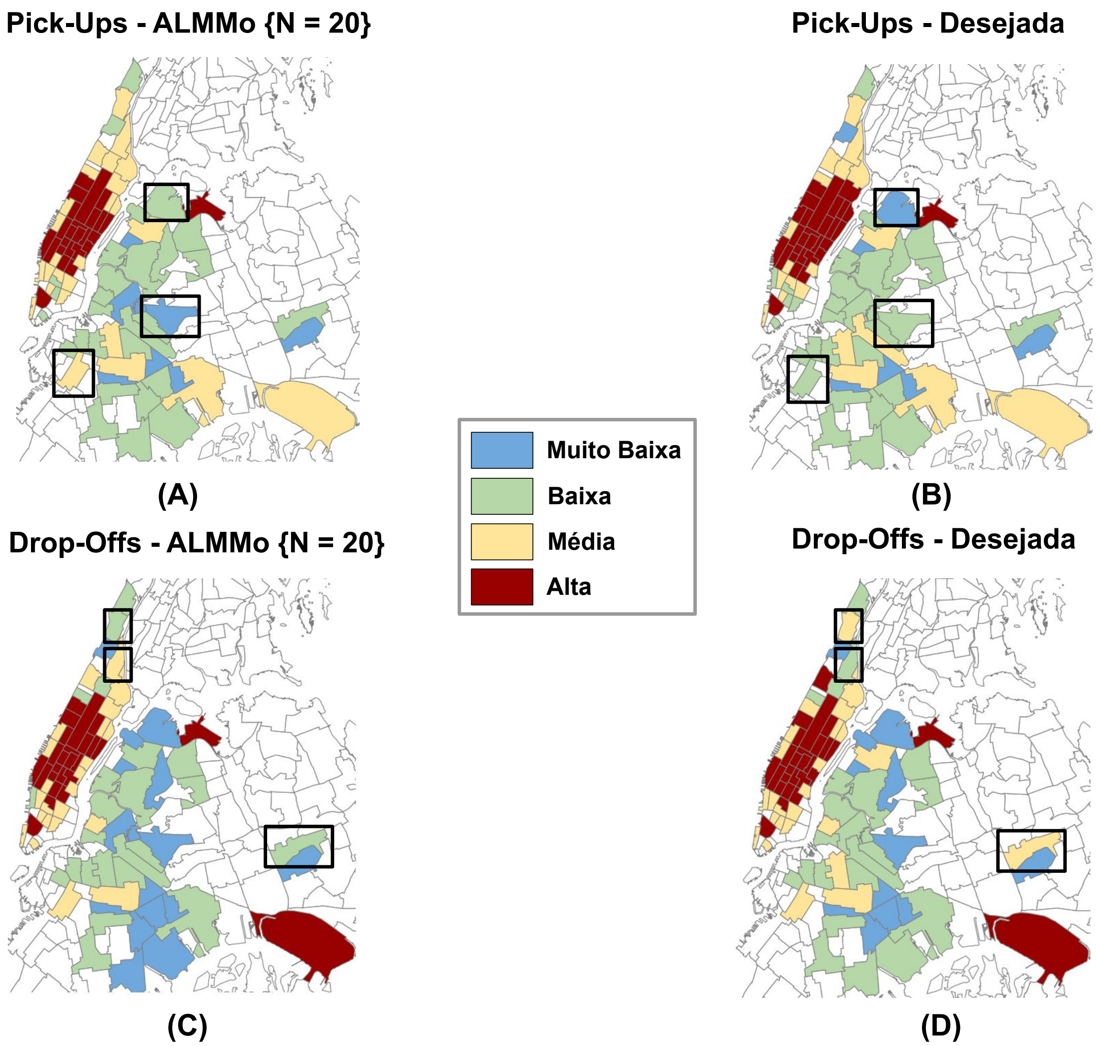


Figure 9: Heatmap for Pick-Up and Drop-Off in New York on 10/31/2018 from 12:00 to 12:15 (15-minute interval). In **A** and **C** the forecast by ALMMo is shown (with 20 input variables) for Pick-Up and Drop-Off. The actual output is shown in **C** and **D**.

period.

Table 6: Accuracy for the 30-minute interval.

Model	Pick-Up		Drop-Off	
	Accuracy (%)	± STD	Accuracy (%)	± STD
ALMMo {N = 20}	**86.529**	**3.538**	**87.256**	**3.329**
eFCE {N = 10}	85.556	3.696	86.161	3.405
eMG {N = 10}	85.203	3.732	85.843	3.530
eNFN {N = 5}	83.579	4.666	85.084	3.865

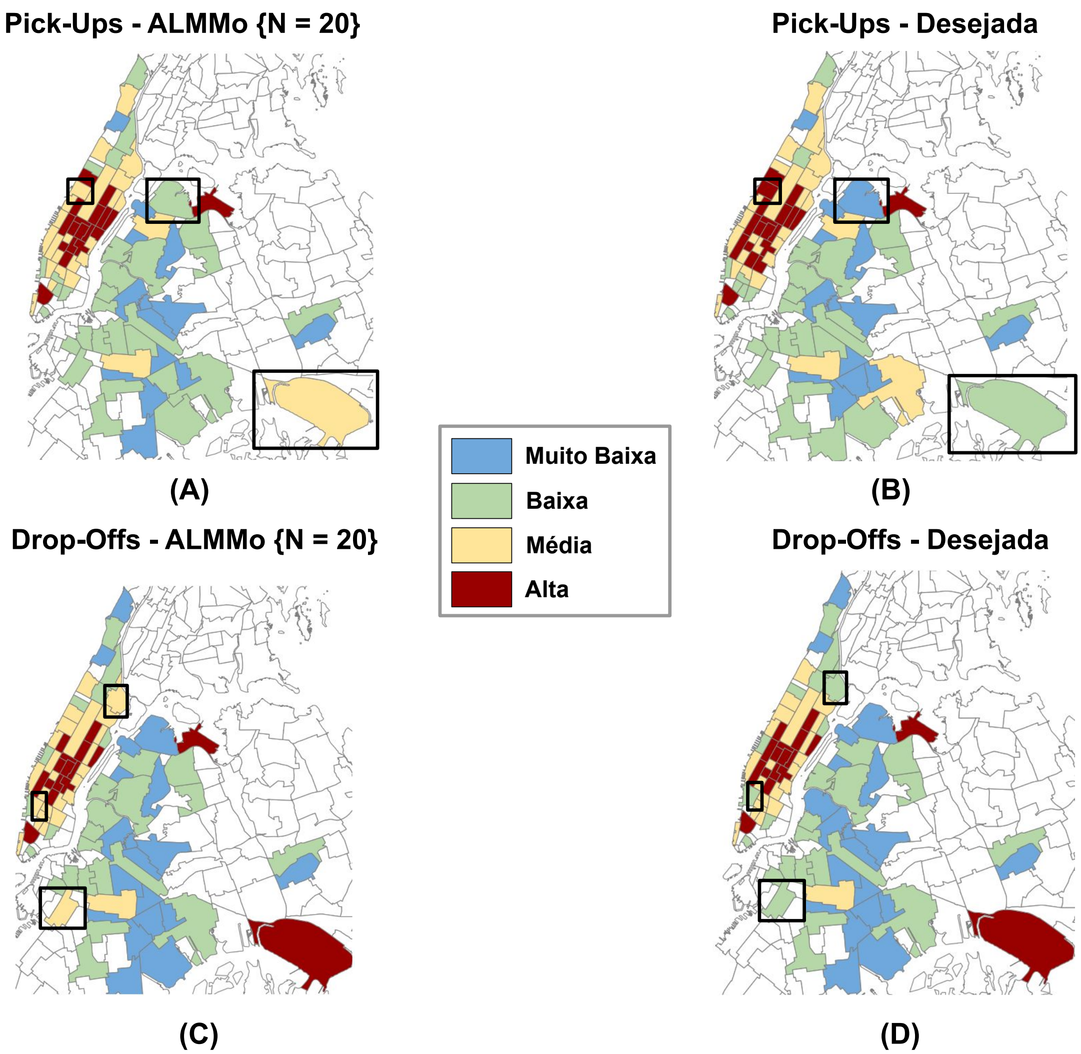


Figure 10: Heatmap for Pick-Up and Drop-Off in the 86 zones in New York on 10/31/2018 from 12:00 to 12:30 (30-minute interval). In **A** and **C** the forecast by ALMMo is shown (with 20 input variables). The actual output is shown in **B** and **D**.

6.3 Feature Selection Analysis

This section presents the analysis of the input variables identified by the Kruskal-Wallis method. Heatmaps are used to show the percentage of selection of each variable. Warmer colors indicate higher percentages and cooler colors lower percentages. To make it easy to understand how the percentage calculation is performed, consider the following example: take a study made in 25 zones for a 15-minute Pick-Up, and the selection considering the most relevant $N = 5$ variables. Therefore, there are 25 data sets. Assuming that the Kruskal-Wallis identified dhd_{tr-2} among the most relevant $N = 5$ variables in 24 of the 25 zones, its percentage will be 96% (24/25 * 100), and a warmer color will illustrate it.

Figures 11 and 12 show the heatmaps in 15-minute intervals for Pick-Up and Drop-Off, respectively. When $N = 5$, all demands from previous days showed some percentages, indicating great relevance as possible candidates for the models. From $N = 5$ to $N = 10$, the variable dhi increases in times $t-1$ to $t-5$. When $N = 20$, all historical demand variables proved relevant. Regarding exogenous and auxiliary variables, the weekend (fs), day of the week (ds), and temperature (te) showed relevant percentages.

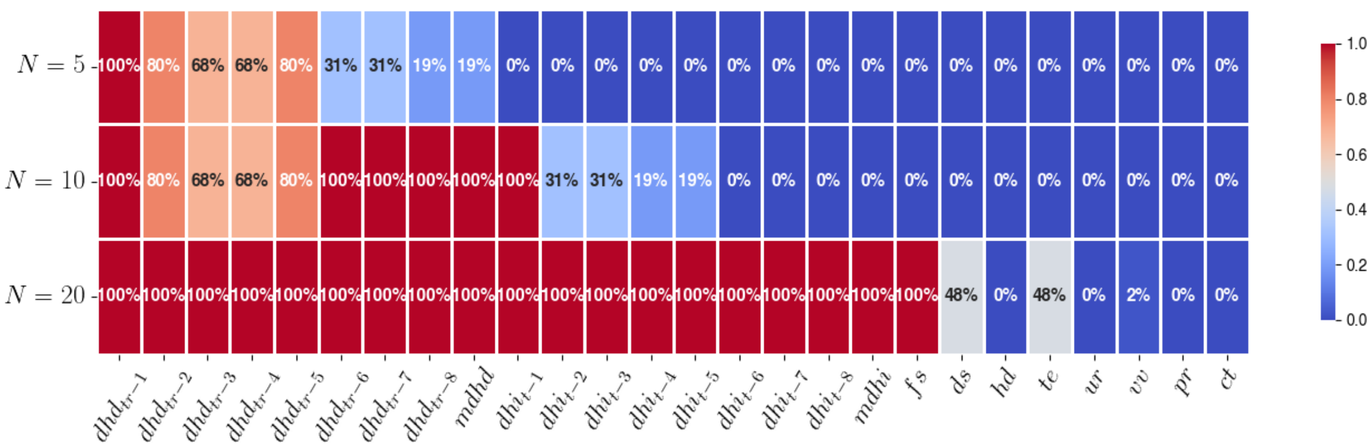


Figure 11: Heatmap of input variables for Pick-Up in New York at 15-minute intervals.

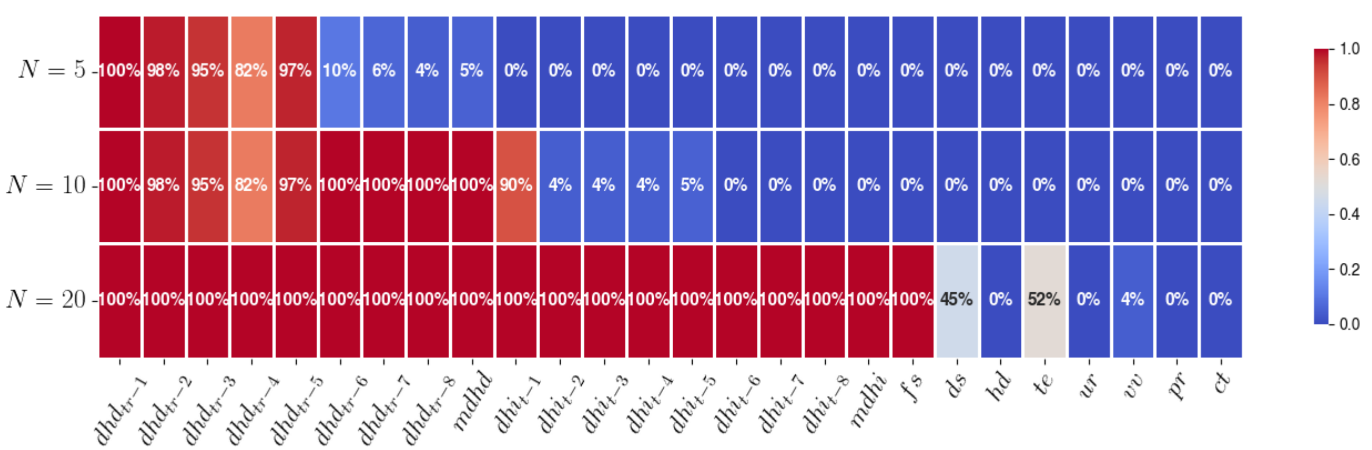


Figure 12: Heatmap of input variables for Drop-Off in New York (86 zones) at 15-minute intervals.

Figures 13 and 14 represent the heatmaps at 30-minute intervals for Pick-Up and Drop-Off, respectively. Compared to the 15-minute maps, it can be seen that in $N = \{5, 10\}$, in addition to the demands of the previous days, there is a greater selection of the historical demands of the intervals (dhi). At $N = 20$, we only have the weekend (fs), day of the week (ds), and temperature (te) with some selection percentage.

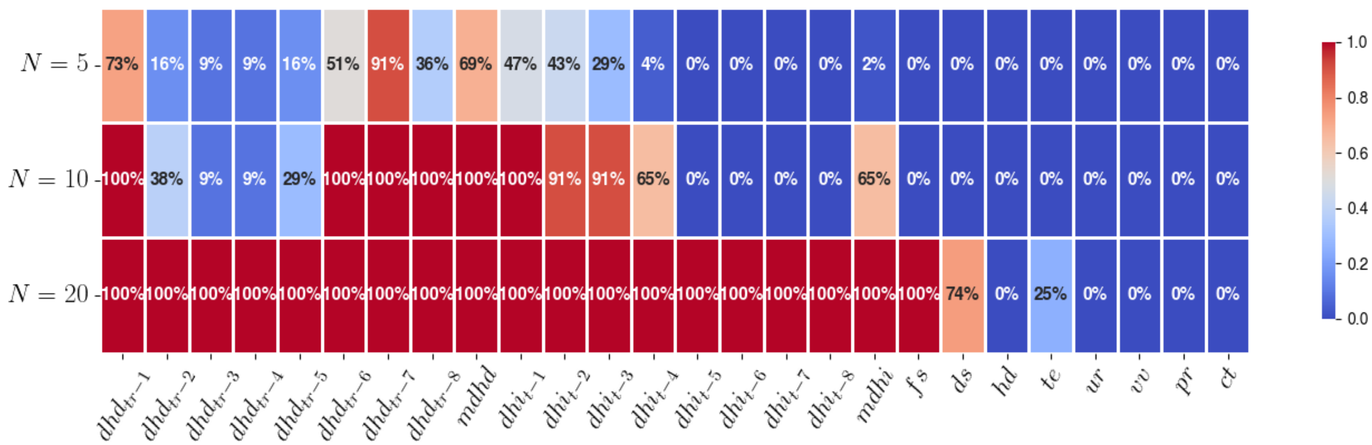


Figure 13: Heatmap of input variables for Pick-Up in New York (86 zones) at 30-minute intervals.

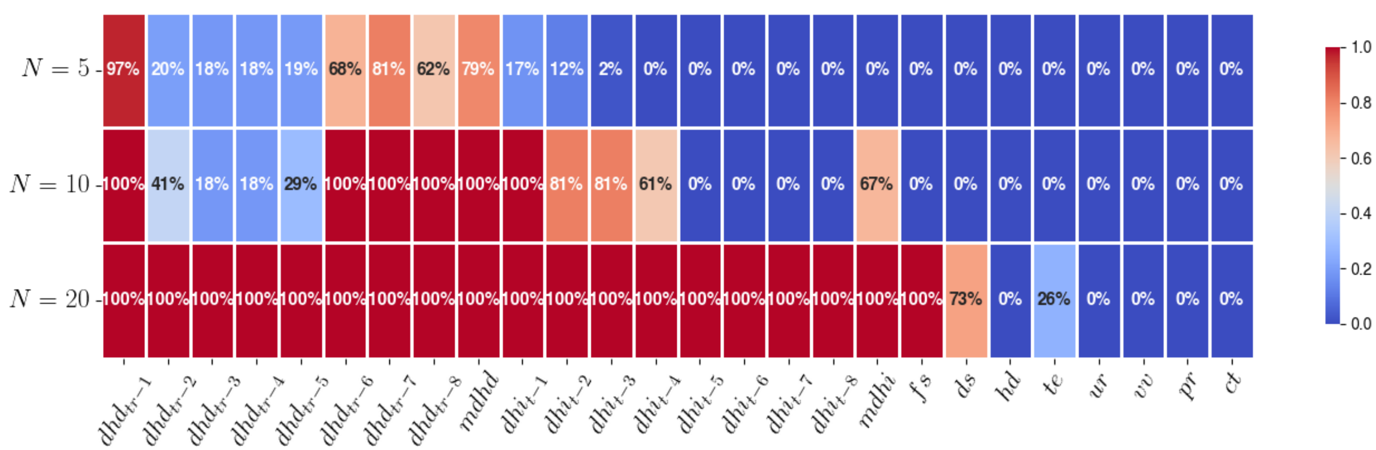


Figure 14: Heatmap of input variables for Drop-Off in New York (86 zones) at 30-minute intervals.

7 Conclusions

This work presented an approach for forecasting and classifying taxi passenger demand using evolving fuzzy systems. This approach uses historical data and meteorological information as input to the models. First, these variables are ordered according to relevance using the Kruskal-Wallis method. Then, they are used as inputs in an evolving fuzzy system to forecast demand for the next time interval. Demand classification converts the forecast outputs into one of the four classes defined in this work. This classification is represented by a color, which will compose a heatmap.

The approach was evaluated by computational experiments considering four evolving systems: ALMMo, eMG, eFCE, and eNFN. Demand forecasting and classification were performed in 86 zones in New York City for Pick-Up and Drop-Off at 15 and 30-minute intervals. The evolving models for prediction were compared to results obtained in another work in the literature. ALMMo presented the best results compared with other evolving models and in contrast with state of the art. In terms of accuracy, ALMMo was also the model that showed the best performance in all experiments. All algorithms demonstrated accuracy above

80%, which indicates good results from the classification approach. The results obtained and the analyses in this work suggest that the proposed approach is promising for forecasting and classifying passenger demand. Furthermore, it can be noted that the incremental processing of samples by evolving systems and the use of heatmaps as a form of visualization are approaches that differ from most works with the same theme in the literature.

A detailed analysis of the input variables made it possible to identify higher selection rates by historical demands (*dhd* and *dhi*), along with their respective means. The exogenous and auxiliary variables were the least selected by the Kruskal-Wallis method, indicating little relevance of these as candidates for the models. For future experiments, a possible suggestion would be to discard the exogenous and auxiliary variables as potential inputs to the models, submitting only the historical demands to the Kruskal-Wallis method.

As future work proposals, expanding the experiments to more rides databases in other cities is suggested. Another suggestion for future work would be to consider fuzzy sets to classify demands as an alternative to histograms. Finally, evaluating the proposed approach in more evolving systems consolidates its applicability.

Acknowledgements: The authors acknowledge CAPES, Brazilian Ministry of Education, code 001.

Conflict of Interest: The authors declare no conflict of interest.

References

[1] P. Angelov, Evolving Takagi-Sugeno Fuzzy Systems from Streaming Data (eTS+), Evolving Intelligent Systems: Methodology and Applications, (2010), 21-50. doi:10.1002/9780470569962.ch2.

[2] P. P. Angelov and X. Gu, Applications of autonomous learning multi-model systems, *in: Empirical Approach to Machine Learning*, (2019), 277-293. doi:10.1007/978-3-030-02384-3_12.

[3] P. Angelov and R. Yager, A new type of simplified fuzzy rule-based system, *International Journal of General Systems*, 41(2) (2012), 163-185. doi:10.1080/03081079.2011.634807.

[4] P. Angelov and X. Zhou, Evolving fuzzy systems from data streams in real-time, *Proceedings of the 2006 International Symposium on Evolving Fuzzy Systems, EFS06 (March)*, (2006), 29-35. doi:10.1109/ISEFS.2006.251157.

[5] P. P. Angelov, X. Gu and J. C. Principe, Autonomous learning multimodel systems from data streams, *IEEE Transactions on Fuzzy Systems*, 26(4) (2017), 2213-2224. doi:10.1109/TFUZZ.2017.2769039.

[6] P. Angelov, X. Gu, D. Kangin and J. Principe, Empirical data analysis: A new tool for data analytics, *in: 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC), IEEE, Budapest, Hungary*, (2016), 000052-000059. doi:10.1109/SMC.2016.7844219.

[7] S. Babicki, D. Arndt, A. Marcu, Y. Liang, J. R. Grant, A. Maciejewski and D. S. Wishart, Heatmapper: web-enabled heat mapping for all, *Nucleic acids research*, 44(W1) (2016), W147-W153.

[8] L. Bai, L. Yao, S. S. Kanhere, Z. Yang, J. Chu and X. Wang, Passenger demand forecasting with multi-task convolutional recurrent neural networks,*in: Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer*, (2019), 29-42. doi:10.1007978-3-030-16145-3_3.

[9] A. A. Bojko, Informative or misleading? heatmaps deconstructed, *in: J. A. Jacko (Ed.), Human-Computer Interaction. New Trends, Springer Berlin Heidelberg, Berlin, Heidelberg*, (2009), 30-39.

[10] G. Box and G. Jenkins, Time Series Analysis: Forecasting and Control, *Wiley, Hoboken, Nova Jersey, EUA*, (1976).

[11] C. Chatfield, Time-series forecasting, *1st Edition, CRC press, CRC*, (2000). doi:10.1201/9781420036206.

[12] J. Crampton, Gis and geographic governance: reconstructing the choropleth map, *Cartographica: The International Journal for Geographic Information and Geovisualization*, 39(1) (2004), 41-53.

[13] F. D. da Cunha, G. Maia, C. Celes, D. Guidoni, F. de Souza, H. Ramos and L. Villas, Sistemas de Transporte Inteligentes: Conceitos, Aplicações e Desafios, in: XXXV simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC) - 2017, no. May, (2017), 1-47.

[14] N. Davis, G. Raina and K. Jagannathan, Taxi Demand Forecasting: A HEDGE-Based Tessellation Strategy for Improved Accuracy, *IEEE Transactions on Intelligent Transportation Systems*, 19(11) (2018), 3686-3697. arXiv:1805.06619.

[15] S. S. Faghih, A. Safikhani, B. Moghimi and C. Kamga, Predicting Short-Term Uber Demand in New York City Using Spatiotemporal Modeling, *Journal of Computing in Civil Engineering*, (2019).

[16] M. M. Ferdaus, M. Pratama, S. G. Anavatti and M. A. Garratt, Palm: An incremental construction of hyperplanes for data stream regression,*IEEE Transactions on Fuzzy Systems*, 27(11) (2019), 2115-2129. doi:10.1109/TFUZZ.2019.2893565.

[17] S. George and A. K. Santra, An improved long short-term memory networks with Takagi-Sugeno fuzzy for traffic speed prediction considering abnormal traffic situation, *Computational Intelligence (January)*, (2020), 1-30.

[18] S. Ishiguro, Y. Fukazawa and S. Kawasaki, Taxi demand forecast using real-time population generated from cellular networks, *UbiComp/ISWC 2018 - Adjunct Proceedings of the 2018 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2018 ACM International Symposium on Wearable Computers*, (2018), 1024-1032.

[19] S. Jiang, W. Chen and Z. Li, H. Yu, Short-term demand prediction method for online car-hailing services based on a least squares support vector machine, *IEEE Access*, 7 (2019), 11882-11891. doi:10.1109/ACCESS.2019.2891825.

[20] N. Kasabov and D. Filev, Evolving intelligent systems: Methods, learning, applications, *in: 2006 International Symposium on Evolving Fuzzy Systems, IEEE, Ambelside, UK*, (2006), 8-18. doi: 10.1109/ISEFS.2006.251185.

[21] J. Ke, H. Zheng, H. Yang and X. M. Chen, Short-term forecasting of passenger demand under on-demand ride services: A spatio-temporal deep learning approach, *Transportation Research Part C: Emerging Technologies*, 85 (2017), 591-608. doi:10.1016/j.trc.2017.10.016.

[22] X. Kong, F. Xia, Z. Fu, X. Yan, A. Tolba and Z. Almakhadmeh, TBI2Flow: Travel behavioral inertia based long-term taxi passenger flow prediction, *World Wide Web*, 23(2) (2020), 1381-1405.

[23] W. H. Kruskal and W. A. Wallis, Use of ranks in one-criterion variance analysis, *Journal of the American statistical Association*, 47(260) (1952), 583-621.

[24] D. Leite, I. krjanc and F. Gomide, An overview on evolving systems and learning from stream data, *Evolving Systems*, (2020). doi:10.1007/s12530-020-09334-5.

[25] A. Lemos, W. Caminhas and F. Gomide, Multivariable gaussian evolving fuzzy modeling system, *IEEE Transactions on Fuzzy Systems*, 19(1) (2010), 91-104. doi:10.1109/TFUZZ.2010.2087381.

[26] B. Li, D. Zhang, L. Sun, C. Chen, S. Li, G. Qi and Q. Yang, Hunting or waiting? discovering passengerfinding strategies from a large-scale real-world taxi dataset, *in: 2011 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops), IEEE, Seattle, WA, USA*, (2011), 63-68. doi:10.1109/PERCOMW.2011.5766967.

[27] S. Liao, L. Zhou, X. Di, B. Yuan and J. Xiong, Large-scale short-term urban taxi demand forecasting using deep learning, *Proceedings of the Asia and South Pacific Design Automation Conference, ASP-DAC 2018-Janua*, (2018), 428-433. doi:10.1109/ASPDAC.2018.8297361.

[28] C.-Y. Lin, S.-L. Tung, P.-W. Lu and T.-C. Liu, Predictions of taxi demand based on neural network algorithms, *International Journal of Intelligent Transportation Systems Research*, 19(3) (2021), 477-495. doi:10.1007/s13177-020-00248-9.

[29] Z. Liu, H. Chen, Y. Li and Q. Zhang, Taxi demand prediction based on a combination forecasting model in hotspots, *Journal of Advanced Transportation 2020*, (2020). doi:10.1155/2020/1302586.

[30] T. Liu, W. Wu, Y. Zhu and W. Tong, Predicting taxi demands via an attention-based convolutional recurrent neural network, *Knowledge-Based Systems 206*, (2020), 106-294. doi:10.1016/j.knosys.2020.106294.

[31] L. Liu, Z. Qiu, G. Li, Q. Wang, W. Ouyang and L. Lin, Contextualized spatialtemporal network for taxi origin-destination demand prediction, *IEEE Transactions on Intelligent Transportation Systems*, 20(10) (2019), 3875-3887. doi:10.1109/TITS.2019.2915525.

[32] E. Lughofer, Evolving fuzzy systems: Fundamentals, *reliability, interpretability, useability and applications, in: 2015 7th International Joint Conference on Computational Intelligence (IJCCI)*, 1 (2015), 11-11.

[33] E. Lughofer and P. Angelov, Handling drifts and shifts in on-line data streams with evolving fuzzy systems, *Applied Soft Computing Journal*, 11(2) (2011), 20572068. doi:10.1016/j.asoc.2010.07.003.

[34] H. Luo, J. Cai, K. Zhang, R. Xie and L. Zheng, A multi-task deep learning model for short-term taxi demand forecasting considering spatiotemporal dependences, *Journal of Traffic and Transportation Engineering (English Edition)*, 8(1) (2021), 83-94. doi:10.1016/j.jtte.2019.07.002.

[35] I. Markou, F. Rodrigues and F. C. Pereira, Real-Time Taxi Demand Prediction using data from the web, *in: IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, (2018).

[36] L. Moreira-Matias, J. Gama, M. Ferreira and L. Damas, A predictive model for the passenger demand on a taxi network, *in: 2012 15th International IEEE Conference on Intelligent Transportation Systems, IEEE*, (2012), 1014-1019. doi:10.1109/ITSC.2012.6338680.

[37] L. Moreira-Matias, J. Gama, M. Ferreira, J. Mendes-Moreira and L. Damas, Predicting taxi-passenger demand using streaming data, *IEEE Transactions on Intelligent Transportation Systems*, 14(3) (2013), 1393-1402. doi:10.1109/TITS.2013.2262376.

[38] L. Moreira-Matias, J. Gama, M. Ferreira, J. Mendes-Moreira and L. Damas, Time-evolving O-D matrix estimation using high-speed GPS data streams, *Expert Systems with Applications*, 44 (2016), 275-288. doi:doi.org/10.1016/j.eswa.2015.08.048.

[39] J. D. Novaković, A. Veljović, S. S. Ilić, Ž. Papić and T. Milica, Evaluation of classification models in machine learning, *Theory and Applications of Mathematics & Computer Science*, 7(1) (2017), 39-46.

[40] F. Rodrigues, I. Markou and F. C. Pereira, Combining time-series and textual data for taxi demand prediction in event areas: A deep learning approach, *Information Fusion*, 49 (2019), 120-129. arXiv: 1808.05535, doi:10.1016/j.inffus.2018.07.007.

[41] F. P. S. Rodrigues, A. M. Silva and A. P. Lemos, Evolving fuzzy system with multivariable gaussian participatory learning and recursive maximum correntropy - efce, *in: 2021 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, (2021), 1-6. doi:10.1109/FUZZ45933.2021.9494451.

[42] F. P. S. Rodrigues, A. M. Silva and A. P. Lemos, An evolving fuzzy predictor with multivariable gaussian participatory learning and multi-innovations recursive weighted least squares - efmi, *Evolving Systems*, (2022). doi:10.1007/s12530-022-09421-9.

[43] A. Rossi, G. Barlacchi, M. Bianchini and B. Lepri, Modelling Taxi Drivers Behaviour for the Next Destination Prediction, *IEEE Transactions on Intelligent Transportation Systems PP*, (2019), 1-10. arXiv:1807.08173.

[44] R. Salman and V. Kecman, Regression as classification, *in: 2012 Proceedings of IEEE Southeastcon, Institute of Electrical and Electronics Engineers (IEEE), Orlando, Florida, USA*, (2012). doi:10.1109/SECon. 2012.6196887.

[45] M. Sayed-Mouchawe and E. Lughofer, Learning in Non-Stationary Environments, *1st Edition, Springer-Verlag New York*, (2012). doi:10.1007/978-1-4419-8020-5.

[46] A. M. Silva, W. Caminhas, A. Lemos and F. Gomide, A fast learning algorithm for evolving neo-fuzzy neuron, *Applied Soft Computing*, 14 (2014), 194-209. doi:10.1016/j.asoc.2013.03.022.

[47] I. krjanc, Cluster-volume based merging concept for incrementally evolving fuzzy Gaussian clustering - eGAUSS+, *IEEE Transactions on Fuzzy Systems*, 28(9) (2020), 2222-2231. doi:10.1109/TFUZZ.2019. 2931874.

[48] I. krjanc, J. Iglesias, A. Sanchis, D. Leite, E. Lughofer and F. Gomide, Evolving fuzzy and neuro-fuzzy approaches in clustering, *regression, identification, and classification: A survey, Information Sciences*, 490(2019), 344-368. doi:10.1016/j.ins.2019.03.060.

[49] Y. Tong, Y. Chen, Z. Zhou, L. Chen, J. Wang, Q. Yang, J. Ye and W. Lv, The simpler the better: a unified approach to predicting original taxi demands based on large-scale online platforms, *in: Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, (2017), 1653-1662. doi:10.1145/3097983.3098018.

[50] U. Vanichrujee, T. Horanont, W. Pattara-atikom, T. Theeramunkong and T. Shinozaki, Taxi demand prediction using ensemble model based on rnns and xgboost, *in: 2018 International Conference on Embedded Systems and Intelligent Technology International Conference on Information and Communication Technology for Embedded Systems (ICESIT-ICICTES)*, (2018). doi:10.1109/ICESIT-ICICTES.2018. 8442063.

[51] Y. Wang, T. Wo, H. Yin, J. Xu, H. Chen and K. Zheng, Origin-destination matrix prediction via graph convolution: A new perspective of passenger demand modeling, *in: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, (2019), 1227-1235. doi:10.1145/ 3292500.3330877.

[52] H. Wei, Y. Wang, T. Wo, Y. Liu and J. Xu, Zest, A hybrid model on predicting passenger demand for chauffeured car service, *in: Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, (2016), 2203-2208. doi:10.1145/2983323.2983667.

[53] P. Xie, T. Li, J. Liu, S. Du, X. Yang and J. Zhang, Urban flow prediction from spatiotemporal data using machine learning: A survey, Information Fusion, 59 (2020), 1-12. arXiv:1908.10218.

[54] J. Xu, R. Rahmatizadeh, L. Bölöni and D. Turgut, A sequence learning model with recurrent neural networks for taxi demand prediction,*in: 2017 IEEE 42nd Conference on Local Computer Networks (LCN), IEEE*, (2017), 261-268. doi:10.1109/LCN.2017.31.

[55] J. Xu, R. Rahmatizadeh, L. Boloni and D. Turgut, Real-Time prediction of taxi demand using recurrent neural networks,*IEEE Transactions on Intelligent Transportation Systems*, 19(8) (2018), 2572-2581. doi:10.1109/TITS.2017.2755684.

[56] R. Yager, A model of participatory learning,*IEEE Transactions on Systems, Man, and Cybernetics*, 20(5) (1990), 1229-1234. doi:10.1109/21.59986.

[57] C. Yang and E. J. Gonzales, Modeling taxi demand and supply in New York city using large-scale taxi GPS data, *Springer Geography*, (2017), 405-425.

[58] H. Yao, X. Tang, H. Wei, G. Zheng and Z. Li, Revisiting spatial-temporal similarity: A deep learning framework for traffic prediction, *in: Proceedings of the AAAI conference on artificial intelligence*, 33 (2019), 5668-5675. doi:10.1609/aaai.v33i01.33015668.

[59] H. Yao, F. Wu, J. Ke, X. Tang, Y. Jia, S. Lu, P. Gong, J. Ye and Z. Li, Deep multi-view spatial-temporal network for taxi demand prediction, *in: Proceedings of the AAAI Conference on Artificial Intelligence,* 32 (2018).

[60] H. Yu, X. Chen, Z. Li, G. Zhang, P. Liu, J. Yang and Y. Yang, Taxi-Based Mobility Demand Formulation and Prediction Using Conditional Generative Adversarial Network-Driven Learning Approaches, *IEEE Transactions on Intelligent Transportation Systems*, 20(10) (2019), 3888-3899.

[61] G. Zander, Predicting taxi passenger demand using artificial neural networks, *Masters thesis, KTH, School of Computer Science and Communication (CSC)*, (2017).

[62] J. Zhao, C. Chen, H. Huang and C. Xiang, Unifying uber and taxi data via deep models for taxi passenger demand prediction, *Personal and Ubiquitous Computing*, (2020). doi:10.1007/s00779-020-01426-y.

[63] Y. Zhou, H. Chen, J. Li, Y. Wu, J. Wu and L. Chen, St-attn: Spatial-temporal attention mechanism for multi-step citywide crowd flow prediction,*in: 2019 International Conference on Data Mining Workshops (ICDMW), IEEE, Beijing, China*, (2019), 609-614. doi:10.1109/ICDMW.2019.00092.

[64] K. Zhang, Z. Liu and L. Zheng, Short-term prediction of passenger demand in multi-zone level: Temporal convolutional neural network with multi-task learning, *IEEE Transactions on Intelligent Transportation Systems*, 21(4) (2019), 1480-1490. doi:10.1109/TITS.2019.2909571.

[65] D. Zhang, L. Sun, B. Li, C. Chen, G. Pan, S. Li and Z. Wu, Understanding taxi service strategies from taxi gps traces, *IEEE Transactions on Intelligent Transportation Systems*, 16(1) (2015), 123-135. doi: 10.1109/TITS.2014.2328231.

[66] C. Zhang, F. Zhu, Y. Lv, P. Ye and F.-Y. Wang, Mlrnn: Taxi demand prediction based on multi-level deep learning and regional heterogeneity analysis, *IEEE Transactions on Intelligent Transportation Systems*, (2021). doi:10.1109/TITS.2021.3080511.

[67] C. Zhang, F. Zhu, X. Wang, L. Sun, H. Tang and Y. Lv, Taxi Demand Prediction Using Parallel Multi-Task Learning Model,*IEEE Transactions on Intelligent Transportation Systems (August)*, (2020), 1-10. doi:10.1109/TITS.2020.3015542.

[68] K. Zhu, S. Zhang, W. Zhang and Z. Zhang, A novel hybrid deep learning model for taxi demand forecasting based on decomposition of time series and fusion of text data, *Journal of Intelligent & Fuzzy Systems (Preprint)*, (2021), 1-17. doi:10.3233/JIFS-210657.

[69] Online Learning versus Offline Learning, *Machine Learning*, 29(1) (1997), 45-63. doi:10.1023/A: 1007465907571.

Lus Fernando Linhares
Graduate Program in Mathematical and Computational Modeling
Federal Center of Technological Education of Minas Gerais
Belo Horizonte, MG, Brazil
E-mail: luisfl89@gmail.com

Alisson Marques da Silva
Graduate Program in Mathematical and Computational Modeling
Federal Center of Technological Education of Minas Gerais
Belo Horizonte, MG, Brazil
E-mail: alissonmarques@gmail.com