Information Fusion 91 (2023) 1–12

Contents lists available at ScienceDirect

Information Fusion

journal homepage: www.elsevier.com/locate/inffus

Full Length Article

Combining heterogeneous data sources for spatio-temporal mobility demand forecasting

Ignacio-Iker Prado-Rujas [a], Emilio Serrano [a], Antonio García-Dopico [b,c], M. Luisa Córdoba [b], María S. Pérez [a,*]

[a] Ontology Engineering Group, ETSI Informáticos, Universidad Politécnica de Madrid, 28660 Madrid, Spain
[b] DATSI Computer Science, ETSI Informáticos, Universidad Politécnica de Madrid, 28660 Madrid, Spain
[c] Centro de Investigación en Simulación Computacional, ETSI Informáticos, Universidad Politécnica de Madrid, 28660 Madrid, Spain

ARTICLE INFO

Keywords:
Mobility demand
Spatio-temporal forecasting
Multi-sensor information
Deep learning

ABSTRACT

There is a growing need to optimize mobility in medium to large-size cities. The use of a car for one-person trips is widely established as a common trend, which combined with the age of the vehicle fleet from many countries leads to high levels of pollution. Besides, the time wasted on commuting is more than significant for many people. Under these premises, it is paramount to understand the dynamics of mobility in every city. In this work, the problem of modeling and predicting transport demand in large cities with high spatio-temporal resolution is tackled. The city studied and its metropolitan area are subdivided into a new mobility mesh-grid, and transport demand is binned into short time intervals. The proposed Spatio-Temporal Mobility Demand Forecaster (ST-MDF) model is trained with real mobility demand data (such as taxi and bicycle rental), historical weather data (e.g., temperature, precipitation, and wind speed), and temporal information (e.g., weekday, time, and holiday) to predict mobility demand in every region of the mesh, for several forecast horizons.

The experiments show that the ST-MDF model exhibits flexibility and robustness, while at the same time it outperforms the baseline models, such as a Long Short-Term Memory (LSTM) network, or the persistence and naive models.

1. Introduction

Commuting is one of the most time-consuming tasks of modern life, loathed by most people. Traffic jams have become part of many people's routines, and the environment has long begging for a change. Analysis and modeling of global public transport demand are key to improve mobility and therefore for the sustainable development of any city.

To this end, the first step is to learn about the commuting habits of citizens. Fiorello et al. [1] report that the car is the most widely used transport mode for frequent trips in the EU. This fact, combined with the low occupancy rate reported in their work (1.7 persons per car), brings into the second key element of mobility: traffic congestion. Vlahogianni et al. [2] touch upon a second challenge when forecasting short-term traffic, which is the choice of data resolution and the identification of spatial and temporal flow patterns. In their comprehensive review, they also emphasize the role of Artificial Intelligence (AI) as a flexible tool for the development of many transportation applications (e.g., [3–5]). However, there is a third question that can help capture the full picture: how to meet global mobility demand? Collecting and fusing urban data from individuals to analyze and model it can be beneficial. However, Liu et al. [6] discuss that this is not an easy task. One way to bridge this gap is by harnessing datasets that collect taxi, for-hire vehicle (FHV), or public bike trip records.

The problem under study in this work can be stated as a multiple time series forecasting one, i.e., given a time-dependent sequence of historical mobility demand data, predict the future demand for certain horizons. To this end, as a first step, an analysis of real-world taxi trips and bike rides datasets from the city of Chicago was performed. Afterward, a mobility demand mesh-grid is formally defined based on the most relevant aspects (trip counts, time, spatial distribution, etc.) obtained from the analysis. The mobility mesh-grid captures the spatial correlations that occur between different city zones. It can be seen as a snapshot of mobility demand as time evolves, similar to the frames that constitute a video. Thereafter, the time-dependent mobility mesh-grid dataset is used to train the Spatio-Temporal Mobility Demand Forecaster (ST-MDF) model in order to predict mobility demand in

* Corresponding author.
E-mail address: maria.s.perez@upm.es (M.S. Pérez).

https://doi.org/10.1016/j.inffus.2022.09.028
Received 6 August 2021; Received in revised form 5 September 2022; Accepted 30 September 2022
Available online 7 October 2022

several future horizons. It is based on the Convolutional Long Short-Term Memory (LSTM) layers, which are known to seize both spatial and temporal aspects. Besides them, the model incorporates two additional modules that extract features from weather data and temporal information (such as the time of the days or whether that day was a public holiday). Thereupon, a comparison is presented between the proposed network and different baselines, including the persistence and naive average models, or an LSTM network. Furthermore, an analysis is conducted to show how the different components of the ST-MDF model contribute towards the final prediction. Finally, experiments show that the proposed framework is flexible (i.e., it can cope with changes in the number or distribution of the city taxi zones and bike racks), and robust (i.e., missing data can be filled thanks to the mobility mesh-grid).

The main contributions of this work are summarized as follows:

1. Definition of a global mobility mesh-grid of a city or region, which integrates heterogeneous data (taxi trips, bike rides, etc.) thanks to the use of spatial information. Furthermore, it is possible to expand the mobility mesh-grid with other data sources (scooters, metro, bus, etc.), as will be discussed.
2. Development and training of the ST-MDF model, which harness spatio-temporal mobility demand data as the input, alongside additional relevant features like weather or temporal data. Moreover, the forecasts are obtained for several horizons with that single model at once.
3. Flexibility and robustness in the number and location of data sources (bike racks, taxi zones, etc.). This aspect is fundamental when designing models that deal with many components that may vary over time, which is usually the case in smart city applications.
4. Forecasting error that beats the baselines, such as the persistence and naive models, or an LSTM network.

The structure of this article is as follows: Section 1 introduces the problem and presents the main contributions. Section 2 summarizes the main related works currently being studied in the field. Section 3 presents the developed methodology and model, including background and notation 3.1, the temporal aspect of the problem 3.2, the spatial dimension 3.3, and the presented ST-MDF model 3.4. Section 4 describes the case of study 4.1, and the experiments and evaluation, including error 4.2.2, flexibility 4.2.3 and robustness 4.2.4. Section 5 discusses the main results of the experiments and their implications. Finally, Section 6 outlines the main conclusions and future research lines.

2. Related work

There is a clear growing interest in understanding mobility dynamics in cities, proven by the numerous and diverse approaches being developed in the last decade (see Table 1). Usually, these works focus on flow prediction from one of three different categories [7]: crowd flow (human mobility) [4,8,9], traffic flow (vehicle congestion) [10–12], and public transport flow (taxi, bicycle, bus, metro, etc.) [13–15]. As for the methods used to tackle such tasks, they can be grouped as statistics-based, machine learning-based, deep learning-based, and reinforcement learning-based methods. Specifically, the current work lies within the category of forecasting public transit flow using deep learning-based methods.

Statistics-based methods. Early works on the topic treat mobility demand as a time series forecasting task [17–19]. Li et al. [18] forecast taxi pick-ups using ARIMA in individual hotspots from GPS trajectories. Moreira-Matias et al. [19] also make use of ARIMA, although they use it in combination with time-varying Poisson averaged models, as an ensemble. Seasonal ARIMA (SARIMA) is an extension to traditional ARIMA, which is employed by Zhang et al. [17] to predict short-term traffic flow. Other approaches simply base the future mobility demand on a probability distribution [20]. For instance, Ma et al. [20] propose a taxi scheduling system to respond to real-time requests sent by pedestrians simulated with a Poisson distribution. Miao et al. [35] propose a taxi dispatching system as well, although they also study the problem of imbalance (taxis tend to concentrate in certain zones, neglecting other areas). They proposed a framework that combines historical and real-time GPS and occupancy data, along with a receding horizon control approach. Taxi dispatching and imbalance are difficult problems related to crowd flow. Ma et al. [16] designed a set of crowd flow visualization techniques, providing also a temporal correlation analysis in their work. Despite the fact that these methods have proven successful for many time series forecasting problems, they cannot capture spatial dependencies inherent to urban mobility flow.

Table 1
Summary of related works, classified by target and method.

	Crowd flow	Traffic flow	Public transit flow
Statistics-based	[16]	[10,17]	[18–20]
Machine learning-based	[8]	[11,21]	[22–25]
Deep learning-based	[4,26]	[12]	Taxi: [3,27–29] Bike: [14,30–32]
Reinforcement learning-based	[9]	[33,34]	[15]

Machine learning-based methods. Similar to statistical methods, many Machine Learning (ML)-based algorithms have been employed in the field [8,11,21,22]. The early work of Li et al. [18] already uses a simple Bayesian Network (BN) for the prediction of taxi demand. More recently, Roos et al. [22] also employs Bayesian Networks, albeit for metro passenger flow forecasting. They manage to beat the historical average while being able to work with missing data thanks to a structural expectation–maximization (EM) algorithm. Markov random fields can capture some dependencies that BNs may not, see for instance the work by Hoang et al. [8]. Differently, Habtemichael and Cetin [21] train a k-nearest neighbors algorithm (k-NN) for the problem of traffic flow forecasting. They conducted thorough experiments on 36 datasets from UK and USA, obtaining better results when compared to parametric models. Lippi et al. [11] compare a SARIMA model that includes a Kalman filter with a Support Vector Machine (SVM) with a Radial Basis Function (RBF) kernel for traffic flow prediction, obtaining similar results. Despite the fact that the former scores better results on average, it is discussed that the SVR-based approach may be a better compromise between accuracy and computational cost. Nevertheless, the availability of huge amounts of heterogeneous data coming from different sources and sensors restricts the use of standard ML-based methods, which are lately being pushed aside primarily as baselines for more advanced methods.

Deep learning-based methods for taxi demand prediction. More recent work on the topic has tended to shift towards Deep Learning (DL)-based methods, for instance for taxi demand prediction [3,27–29]. Xu et al. [27] develop LSTM networks alongside Mixture Density Networks (MDNs) for predicting taxi demand in New York City (USA) zones using previous demand data and other information (weather, time, taxi drop-offs, etc.). The city is divided into many small regions, all of them serving as input to a single model that predicts the probability distribution of taxi demand, which binds the model to a certain number of areas. Their results are compared with the proposed framework in Section 4.2.2. One of the two architectures proposed by Rodrigues et al. [28] also makes use of LSTMs for taxi demand prediction, with the particularity that they introduce textual information that is fused with the LSTM output. Yao et al. [3] go one step beyond, capturing spatial dependencies among nearby regions with convolutional layers in several timesteps, which are later fed to LSTM network and concatenated with the output of a so-called semantic view. The recent work by Liu et al. [29] employs a context-aware attention mechanism to integrate three different predictions: an instant spatio-temporal module (using 1-dimensional convolutions and GRU cells), a short-term (4 days) module, and a long-term (2

weeks) periodic module. Their work includes experiments showing how precision improves when progressively combining those modules. Their results are further studied in Section 4.2.2.

Deep learning-based methods for bicycle demand prediction. Bicycle demand modeling is also shifting towards DL [14,30–32]. In his survey, Jiang [30] gathers many works on the topic that employ DL. He distinguishes different prediction problems on the basis of whether the data are treated as time series, graph, or grid formats. Furthermore, the evaluated works are classified according to the use of exogenous data (e.g., weather, points of interest, and calendar), the software framework employed, and the availability of the data. Chai et al. [31] transform the problem into a bike-sharing system graph, and implement it for Divvy bikes in Chicago and New York City. They employ multi-graph convolutions with an input window comprising the last 6 h to predict the bike flow in the next subsequent hour. Their results are discussed in further detail in Section 4.2.2. Lin et al. [14] also predict station-level hourly bike demand by means of Graph Neural Networks (GNNs), in this case in New York City. In addition to incorporating recurrent information, they explored how to capture pairwise correlations between stations to improve predictions. Differently, Li et al. [32] tackle the problem with a grid-based data structure and study four use cases (including Chicago). The temporal aspects are captured by three independent modules (closeness, period, and trend), which are later fused. Similarly to this work, they employ a Conv-LSTM-based model. However, weather or calendar information is not incorporated into the model, and temporal and spatial granularity are coarser (2 km × 2 km and 1 h, respectively). Furthermore, none of these works integrates different mobility services into a single model.

Deep learning-based methods. Additionally, the DL field has obtained promising results using the technique known as transfer learning (see for instance the works by Wang et al. [26] and Tian et al. [36]). Furthermore, recently developed Deep Neural Networks (DNNs) architectures are already being used for related tasks. One of them is the Convolutional LSTM, which combines the LSTM ability to remember long-term dependencies with the capability of convolutional networks to find spatial correlations. Some examples of their application are precipitation nowcasting [37,38], and solar irradiance forecasting [39]. The current work leverages this kind of network by translating mobility demand forecasting from an individual time series prediction task into an *image-to-image transformation problem*.

Reinforcement learning-based methods. Jiang et al. [15] make use of such techniques to optimize the number of passengers taking the metro at rush hours. They simulated its effect on the Shanghai metro line, reducing the number of passengers stranded on the platform. The work of Khaidem et al. [9] is another example, where reinforcement learning is used to predict human mobility in three real cities and a synthesized one. Similar methods are employed by Walraven et al. [33] in order to reduce traffic congestion while taking into account future traffic predictions. Q-learning is used to achieve this, which establishes the maximum driving speed allowed on highways. This work [33] is also a good example of how reinforcement learning-based methods can take advantage of Artificial Neural Networks (ANNs), and more generally of DL. Another such example is the work of Wei et al. [34], where the authors build a deep reinforcement learning-based model for the operation of traffic lights, with the aim of reducing waiting times. Furthermore, they raise an interesting question for these kinds of methods: how can a reward function be designed to be fair for all the actors that are involved in mobility (pedestrians, bikes, scooters, cars, etc.)?

The presented families of methods provide an extensive overview of works that aim at improving mobility in cities. Nevertheless, few of them propose a framework for integrating heterogeneous data sources into a single structure for mobility forecasting. Additionally, none of those works investigate flexibility or robustness features from their models, but rather focus just on error metrics. The current work aims at contributing to these two lines, beyond exceeding baseline metrics.

3. ST-MDF approach

In this section, the developed ST-MDF approach is detailed. Initially, Section 3.1 sets the concepts and terminology that will be used thereupon. Afterward, the temporal and spatial aspects when modeling transport demand are discussed in Sections 3.2 and 3.3 respectively, to later present the proposed model and its components in Section 3.4.

3.1. Background and definitions

The **temporal granularity/resolution** is the length of the intervals in which time is binned. For instance, the temporal granularity used in this work is 15 min (as suggested in [40]), e.g. t = [18:00, 18:15). The left endpoint will be used as a tag when referring to a certain time interval for the sake of clarity (for instance, 18:00 in the previous example). The **temporal coverage** is the period of time in which certain data were recorded. In this work, the temporal coverage of the mobility data goes from January 2013 to December 2020. The size of the input window when dealing with time series modeling is known as **timestep**, and it is denoted by n_x. Similarly, the **number of horizons** (size of the output window) is denoted by n_y throughout this work. The **shift** s is the leap (number of time intervals) between the input and output windows. When needed, the parameters n_x, n_y, and s will be displayed as a subscript of the model name to simplify the reading. For instance, ST-MDF$^*_{12,8,4}$ refers to the ST-MDF* model with $n_x = 12, n_y = 8$, and $s = 4$.

The **spatial granularity/resolution** is the length of the intervals into which the considered area is divided. For instance, the spatial resolution used in this work is ∼500 × 500 m. The **spatial coverage** is the area covered by the mobility mesh. For the current use case, the spatial coverage comprises the city of Chicago and its metropolitan area, which is approximately 47 by 31 km (see Fig. 1). A **mobility mesh-grid** $\mathcal{M}_t$ is a 2-dimensional array that collects mobility data for a certain time interval t. The shape of the mesh-grid is its number of rows r by its number of columns c (90 × 60 in this work). It is based on a regular partition of the studied area, and the selected spatial resolution determines r and c. Each element of $\mathcal{M}_t$ collects a numerical value that expresses the demand for a certain mobility service in its corresponding region of the mesh-grid during the period t. For instance, it may express the number of bikes rented in each region of the mesh-grid between 18:00 and 18:15.

3.2. Temporal modeling of transport demand

The mobility forecasting problem can be posed as a time series forecasting one (for each mobility service), i.e. given a time-dependent sequence of historical measurements $\{x_t, t \in T\}$ recorded at one fixed location l, predict x_{t+h} for a certain forecast horizon h at l. Here t represents a time interval from the ones considered T, and x_t records historical demand data for a certain mobility service and interval t. This scenario restricts to forecasts in a single location l, which is insufficient to obtain a general perspective of mobility demand in a certain region. Several time series can be considered for multiple locations as $\{x^l_t, t \in T, l \in L\}$, where l is a tag from the considered ones L that represents a certain geolocation. Therefore, these time series $\{x^l_t\}$ can be arranged in a tabular form of shape $|T| \times |L|$, in which the rows index the time interval t and the columns the location l. However, this 2-dimensional representation of the mobility demand problem still disregards spatial correlations between locations.

Once the data are arranged this way, it can be used to train an AI model by presenting samples one at a time and minimizing a loss function. Usually, several consecutive samples are presented to the model at once to provide more information by means of a sliding window. For instance, if $h > 0$: given $\{x^l_{t_0}, x^l_{t_1}, x^l_{t_2}, l \in L\}$ predict $\{x^l_{t_2+h}, l \in L\}$, then given $\{x^l_{t_1}, x^l_{t_2}, x^l_{t_3}, l \in L\}$ predict $\{x^l_{t_3+h}, l \in L\}$, etc. In the previous example n_x was 3, so if the temporal granularity

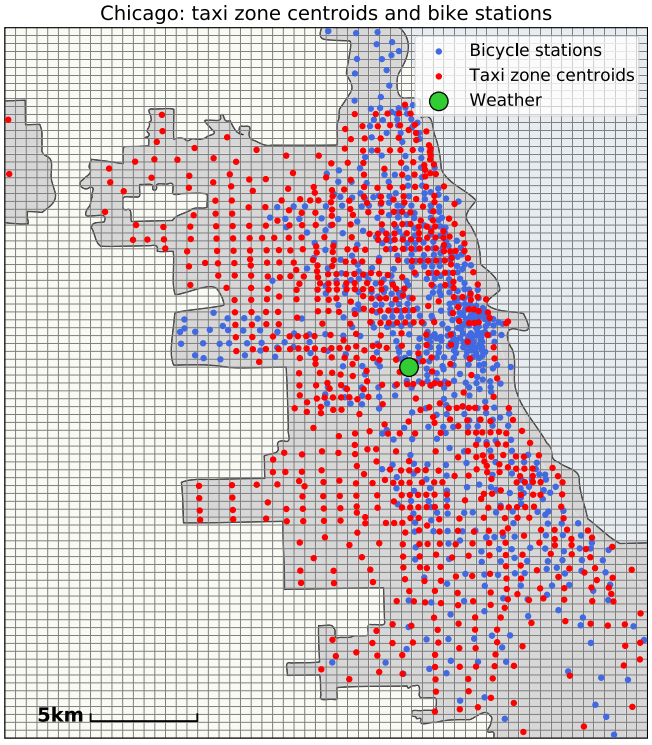


Fig. 1. Representation of Chicago (gray area) with an overlayed mesh-grid of 90 × 60 elements. On top of it, the location of the 684 bicycle racks, 801 taxi zone centroids and weather point are scattered. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

is 15 min then the last 45 min of data are fed to the model at every step. As an extension to this setup, several forecast horizons can be forecasted each time using a certain shift (e.g. $n_y = 2$ and $s = 4$). Following the previous example: given $\{x^l_{t_0}, x^l_{t_1}, x^l_{t_2}, l \in L\}$ predict $\{x^l_{t_6}, x^l_{t_7}, l \in L\}$, then given $\{x^l_{t_1}, x^l_{t_2}, x^l_{t_3}, l \in L\}$ predict $\{x^l_{t_7}, x^l_{t_8}, l \in L\}$, etc.

3.3. Spatio-temporal modeling of transport demand

Clearly, transport demand is strongly influenced by temporal factors, like the time of the day or the day of the week [41]. Section 3.2 presents a general methodology for arranging the data when developing a model for multiple time series forecasting, using a sliding window. While the described setup can suffice when the correlation between time series is merely temporal, some problems also present spatial correlations. Such an example is mobility demand forecasting, e.g. transport demand differs greatly from the airport to a residential area, but demands in neighboring regions of the city center relate to each other.

As in Section 3.2, consider multiple time series that record transport demand arranged in a tabular form of shape $|T| \times |L|$: $\{x^l_t, t \in T, l \in L\}$. Then, thanks to the geolocations of L the table $\{x^l_t\}$ can be unfolded into an additional dimension, provided that the elements of L are arranged as a regular mesh-grid of shape $r \times c$. Therefore, a set of mobility mesh-grids $\{\mathcal{M}_t, t \in T\}$ is obtained, which can be seen as an array of shape $|T| \times r \times c$. When several mobility services S are considered (e.g. taxi and bike), the mobility mesh-grids can be stacked as $\{\mathcal{M}^s_t, t \in T, s \in S\}$, which can be seen as an array of shape $|T| \times r \times c \times |S|$. For the sake of simplicity, S will be omitted from the set of mobility mesh-grids $\{\mathcal{M}_t\}$, and it is assumed that $|S| > 0$ throughout the work.

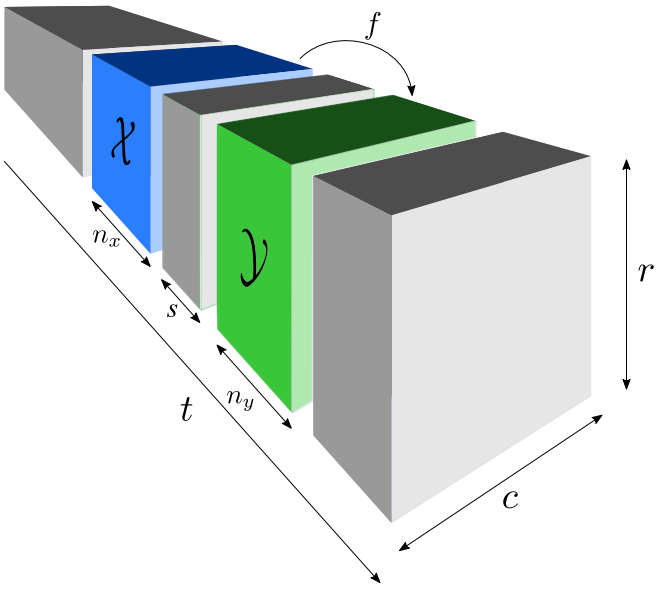


Fig. 2. Spatio-temporal modeling problem. In this framework, t represents time, c the number of columns, r the number of rows, n_x the number of instants taken as input, n_y the number of forecast horizons, s the shift, and f represents the function inferred by the network. In short, $f(\mathcal{X})$ approximates $\mathcal{Y}$, i.e., $f(\mathcal{X}) \approx \mathcal{Y}$.

Now, the problem of mobility demand forecasting can be stated as follows: Given n_x consecutive mobility mesh-grids for time intervals $t_1, \ldots, t_{n_x}$, forecast the mobility mesh-grids for $t_{n_x+s}, \ldots, t_{n_x+s+n_y-1}$, where $s, n_y > 0$, i.e.:

$$\{\mathcal{M}_{t_1}, \ldots, \mathcal{M}_{t_{n_x}}\} \longmapsto \{\mathcal{M}_{t_{n_x+s}}, \ldots, \mathcal{M}_{t_{n_x+s+n_y-1}}\}.$$

If we denote the left part of this expression as $\mathcal{X}$ and the right part as $\mathcal{Y}$, the studied forecasting task can be visually understood as seen in Fig. 2. Section 3.4 will exploit this definition of the problem and present the Conv-LSTM based model that works with mobility mesh-grids. Afterward, Section 4 will evaluate how the use of mesh-grids allows heterogeneous data to be combined and how this improves the forecasts while it equips the model with flexibility and robustness.

3.4. Proposed ST-MDF model

Once the problem is stated in Section 3.3, the proposed Conv-LSTM based model can be described. The ST-MDF* is composed of three modules (see Fig. 3):

(1) **Mobility module**: This module aims at extracting spatio-temporal mobility patterns. It takes $\{\mathcal{M}_{t_1}, \ldots, \mathcal{M}_{t_{n_x}}\}$ as an input and passes them through a series of Conv-LSTM and max-pooling layers. The resulting 3D tensor is then flattened into a mobility feature vector. Note that different transport services are incorporated into a single data structure thanks to the mobility mesh-grids.

(2) **Temporal module**: It provides the temporal aspects that affect transport demand at the last time interval of the input window. The input tensor is obtained from the time interval t_{n_x}, and it contains the time of the day, the time of the week, the time of the year, whether t_{n_x} is on a week or a weekend, and whether it is a holiday or not. The module passes the temporal tensor by several fully-connected layers, and yields a temporal feature vector.

(3) **Weather module**: It extracts weather information, since mobility choices are strongly influenced by current and future meteorological conditions. To this end, it takes eight weather variables (such as temperature, precipitation, and cloud cover)

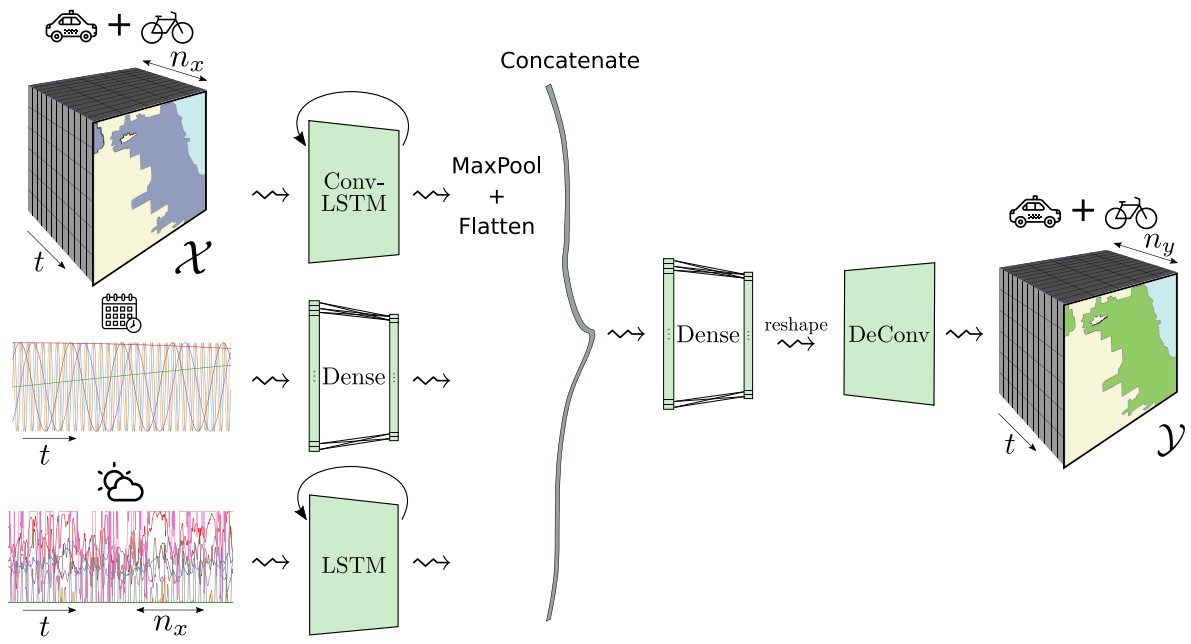


Fig. 3. ST-MDF* model. The top-left depicts the mobility module, below the temporal module is shown, and the bottom-left of the figure captures the weather module. After the concatenation of their feature vectors, they are passed by a dense layer and deconvoluted into n_y mobility demand mesh-grids for taxi and bike rides.

for the same periods as those of the mobility module $t_1, \ldots, t_{n_x}$ and passes that tensor through several LSTM layers that extract a weather feature vector.

Once the three modules have produced their feature vectors, they are concatenated and passed by fully-connected layers, yielding a 1-dimensional vector. This vector is then reshaped into a 4D tensor so that it matches the four output dimensions: forecast horizons, latitude, longitude, and mobility service. Furthermore, the reshape operation allows to perform the last step, which is a deconvolution [42,43] (also known as a transposed convolution). This transformation goes in the opposite direction of a conventional convolution while keeping the same connectivity pattern. Finally, the predicted mobility mesh-grids are: $\{\mathcal{M}_{t_{n_x}+s}, \ldots, \mathcal{M}_{t_{n_x}+s+n_y-1}\}$.

4. Evaluation

Initially, several real-world datasets from the city of Chicago are presented in Section 4.1 as a case of study. Thereafter, the experiments with which the model was trained and its flexibility and robustness evaluated are described in Sections 4.2.2, 4.2.3, and 4.2.4, respectively.

4.1. Case of study: Chicago

Several heterogeneous Chicago datasets were collected and combined to train and evaluate the model described in 3.4. In this section, these datasets and the transformations applied to them are described. Each of the subsequent subsections corresponds to the input data for the three modules of the model. It should be noted that this method could be replicated for a different city (for instance, for New York City, since similar data are available online).

4.1.1. Taxi and bike trips as a grid

Many datasets are collected in the Chicago council data portal.[1] One of them gathers all taxi pick-up and drop-off zones and times from 2013 onward, which has been used in the present work. The data can be downloaded as individual CSV files, one per year. In them, every row represents a taxi trip, including:

- Pick-up zone. These are based on the census tracts of the city of Chicago, which have changed little during the past century. Based on the data, 801 taxi zones are considered (see the red dots in Fig. 1, which represent their centroids).
- Pick-up time, with a temporal resolution of 15 min. The temporal coverage is 8 years (from 2013 until 2020). Therefore, $|T| = 4 * 24 * (365 * 8 + 2) = 280\,512$ for the taxi dataset, i.e. the number of 15 min intervals in 1 h, times the number of hours in a day, times the number of days in a year, times the number of years (plus 2 days accounting for leap years).
- Other fields that were not needed are trip duration (in seconds), trip length (in miles), drop-off zone and time, etc.

From the original 195M trips, approximately 11.9% had to be removed since the pick-up zone was missing for them. This leaves almost 172M taxi trips, which were processed and incorporated into the final dataset. Firstly, taxi trip records were binned for every zone and time interval, obtaining a dataset $X_{\text{taxi}} = \{x_t^l, t \in T, l \in L\}$ that records the number of trips that began during each time interval $t \in T$ for every zone $l \in L$. Afterward, all taxi trip counts x_t^l were normalized following Eq. (1):

$$x' = \frac{x - \min(X)}{\max(X) - \min(X)}, \tag{1}$$

where $x \in X$, and x' represents the normalized value of x. The next step is to build the mobility mesh-grid $\mathcal{M}_t$ from X_{taxi}. A regular mesh-grid that covers the entire city of Chicago was defined to achieve a spatial granularity of ~500 × 500 m, obtaining $r = 90$ and $c = 60$. In addition, the centroid of each taxi zone was computed to obtain L. From there, the mobility mesh-grids $\{\mathcal{M}_t^{\text{taxi}}, t \in T\}$ were linearly interpolated on the 90 × 60 mesh-grid, using X_{taxi} as input to the interpolator along with the geolocations specified by L. In this way, the taxi mesh-grid was built, which has shape $|T| \times 90 \times 60$.

Divvy[2] is a Lyft-owned bicycle rental company that operates many bike sharing systems in the United States. They offer anonymized trip data for public use [44], eliminating the first trips that are less than 1 min, and the trips taken by their staff. Data can be downloaded from

[1] https://data.cityofchicago.org/.

[2] https://www.divvybikes.com/.

their website as individual CSV files, which correspond to periods of variable length. On them, every row represents a bike ride, including (among other fields):

- Start bike station. Here, it should be noted that some periods included bike station identifiers alongside lookup files for the stations' location, others were missing the lookup files (so the geolocation was assigned based on the information from the other periods), and others included the exact geolocation where the trip started. Depending on the period, around 684 bike stations were operational (see the blue dots in Fig. 1, which represent their location).
- Start timestamp, using a temporal granularity of seconds. Time was binned into 15 min intervals to match the temporal granularity of the taxi data. The same as for the taxi data, the temporal coverage is 8 years (from 2013 until 2020). Therefore, $|T| = 4 * 24 * (365 * 8 + 2) = 280\,512$ as before.
- Other fields that were not needed, such as end bike station, end time, rider type, etc.

Approximately 0.35% of the trips had to be removed because their longitude or latitude was missing, leaving over 24M of bike rides that were processed. Bike rental mesh-grids were produced for each $t \in T$, similarly to the case of taxis. However, this dataset has a different nature than the taxi one, since the start location is recorded exactly instead of providing a disjoint set of zones. Initially, the bike rides were temporally and spatially binned, obtaining the dataset $X_{\text{bike}} = \{x_t^l, t \in T, l \in L\}$, where L corresponds to the same 90×60 mesh-grid as before. Thus, X_{bike} collects the number of bicycle rides that began during each time interval $t \in T$ for every zone of the grid $l \in L$. Now, X_{bike} can be unfolded into an additional dimension, which yields a set of bike mesh-grids $\{\mathcal{M}_t^{\text{bike}}, t \in T\}$. These mesh-grids were normalized using Eq. (1) as before. Finally, the taxi and bike mesh-grids were stacked together (as explained in Section 3.3), obtaining the final mobility mesh-grid dataset $\{\mathcal{M}_t, t \in T\}$ of shape $|T| \times 90 \times 60 \times 2$. As mentioned in Section 3.3, the mesh-grid dataset can be expanded with more mobility data sources, obtaining a shape of $|T| \times 90 \times 60 \times |S|$. This will be further discussed in Section 5.

During the data analysis process, it was noticed that some bike stations changed their location over time, likely due to construction works or relocations. Under this scenario, working with mesh-grids is beneficial due to their flexibility, as will be discussed in Section 4.2.3. In the same vein, note that other bike-sharing systems are not dock-based, but instead the user can leave the bicycle anywhere in the city (within some defined boundaries). Again, due to the flexibility of the mesh-grid, such a system can also be integrated into the ST-MDF model.

4.1.2. Temporal data

As discussed in Section 3.4, temporal information regarding t_{n_x} is fed to the temporal module of the ST-MDF model, encoded in a vector:

- time of the day converted with the sine and the cosine to $[-1, 1]$,
- time of the week converted with the sine and the cosine to $[-1, 1]$,
- time of the year converted with the sine and the cosine to $[-1, 1]$,
- weekday or weekend in $\{0, 1\}$,
- holiday[3] in $\{0, 1\}$.

4.1.3. Meteorological data

Similarly to Section 4.1.2, weather information is fed to the weather module of the ST-MDF model. Meteorological data were collected for the entire time period with a temporal resolution of 1 h (data from http://www.meteoblue.com). The weather location was chosen to be on the centroid of the set of points made up of taxi zones and bicycle stations (see the green dot of Fig. 1). The variables included are as follows:

- temperature [° C],
- precipitation amount [mm],
- snowfall amount [cm],
- relative humidity [%],
- wind speed [km/h],
- wind direction [deg],
- cloud cover [%], and
- solar irradiance [W/m^2].

In this case, no empty records were found. In order to account for the different temporal resolution when compared to the mobility grids (Section 4.1.1), nearest neighbors interpolation was used to fill the missing time intervals. Lastly, these data were normalized using Eq. (1) on each weather variable independently.

4.2. Experiments

Several aspects are key when building spatio-temporal models that work with many sensors. In this section, the experiments that analyze some of them are explained, i.e. the error, flexibility, and robustness of the ST-MDF* model. Initially, Section 4.2.1 depicts the parameter choice and model training, and it explains the experimental setup. Afterward, Section 4.2.2 presents the error scored by the model under different scenarios and compares them with those of the baselines. Finally, flexibility and robustness are studied in Sections 4.2.3 and 4.2.4, respectively.

4.2.1. Experimental setup and model training

The models presented were developed using the Keras library [45] with the Tensorflow backend [46]. Additionally, NumPy, Pandas, and PyTables Python libraries were used for data manipulation and storage. The experiments were carried out using a Ubuntu machine with 12 cores and 24 logical processors. Additionally, the machine was equipped with an NVIDIA GeForce GTX TITAN X graphic card. As a reference, training times ranged from 9 to 34 min per epoch, depending on the parameters n_x and n_y.

As explained in Section 4.1.1, the data used for training cover the period between 2013 and 2020. Here, it should be noted that mobility patterns from 2020 deviate from the usual due to the COVID-19 pandemics. Therefore, data were considered only until mid-March 2020. Specifically, the years 2013 to 2017 were used for training (~70%), 2018 for validation (~14%), and 2019 and from January to mid-March 2020 for testing (~16%). An analysis between eight optimizers (Adadelta, Nadam, SGD, RMSprop, Adagrad, Adamax, Adam, Ftrl) and 3 loss functions (Mean Squared Error (MSE), Mean Absolute Error (MAE) and Mean Squared Logarithmic Error (MSLE)) was carried out for the ST-MDF$^*_{4,4,4}$ during 10 epochs to choose the best possible combination. The training times were all very similar (around 9 min per epoch). Regarding the error, the best results were obtained for MSE loss with Adam, Adamax, RMSprop, and Nadam optimizers, with a slight preference for the latter. Therefore, the Nadam algorithm was used to optimize the MSE loss function in the training phase for 50 epochs in all trained models.

The parameters of the ST-MDF* model (described in Section 3.4) are explained hereafter. The mobility module takes a $n_x \times 90 \times 60 \times 2$ input, and it is composed of one convolutional LSTM layer with 3 filters and kernel size 8×8, and a max-pool layer of size 4×4. The temporal module consists of a fully connected layer with 16 neurons, and the weather module also uses an LSTM layer with 16 neurons. The input tensors to the temporal and weather modules have shapes 8 and $n_x \times 8$, respectively. The outputs of the three modules are then concatenated into a single tensor, which is passed by a dense layer with 1204 neurons and reshaped into a 4D tensor. The final step involves a 3D deconvolution with 2 filters, kernel size 3×6, and strides $2 \times 2 \times 2$, which produces the output tensor of shape $n_y \times 90 \times 60 \times 2$.

[3] Obtained from https://www.officeholidays.com/countries/usa/illinois/.

Table 2
Error comparison between models that work with separated or combined mobility services. Forecast horizons: h_0 = 1 h, h_1 = 1 h 15 min h_2 = 1 h 30 min, h_3 = 1 h 45 min (for 50 epochs in all cases).

#	Model	Evaluated service	Combined services	n_x, n_y, s	RMSE [–]			
					h_0	h_1	h_2	h_3
1	ST-MDF*	Taxi rides	Yes	4, 4, 4	0.146	0.147	0.150	0.153
2			No		0.150	0.153	0.156	0.160
3	ST-MDF*	Bike trips	Yes	4, 4, 4	0.085	0.086	0.086	0.086
4			No		0.091	0.091	0.091	0.091

Table 3
Error assessment of the developed models and baselines for the taxi trips task. Given s, the forecast horizon is obtained as: $h_i = 15 \cdot (i + s)$min, with $, i \in \{0, \dots, 7\}$. Training time is displayed in `days hh:mm` format.

#	Model	n_x, n_y, s	Training time	RMSE [–]							
				h_0	h_1	h_2	h_3	h_4	h_5	h_6	h_7
1	ST-MDF*	4, 4, 4	07:35	0.146	0.147	0.150	0.153	–	–	–	–
2	LSTM		04:24	0.225	0.219	0.222	0.219				
3	BiLSTM		05:29	0.194	0.191	0.190	0.190				
4	Persistence		–	0.143	0.151	0.158	0.165				
5	Naive		–	0.151	0.158	0.165	0.172				
6	ST-MDF*	4, 8, 4	09:40	0.154	0.153	0.153	0.154	0.156	0.158	0.162	0.166
7	LSTM		05:44	0.196	0.195	0.195	0.195	0.196	0.193	0.194	0.193
8	BiLSTM		07:50	0.175	0.175	0.174	0.173	0.173	0.173	0.174	0.174
9	Persistence		–	0.143	0.151	0.158	0.165	0.171	0.178	0.184	0.190
10	Naive		–	0.151	0.158	0.165	0.172	0.178	0.184	0.189	0.195
11	ST-MDF*	8, 4, 4	13:05	0.145	0.145	0.147	0.150	–	–	–	–
12	LSTM		06:42	0.186	0.188	0.191	0.196				
13	BiLSTM		08:18	0.192	0.193	0.192	0.193				
14	Persistence		–	0.143	0.151	0.158	0.165				
15	Naive		–	0.163	0.169	0.175	0.181				
16	ST-MDF*	8, 8, 4	14:52	0.153	0.151	0.150	0.151	0.152	0.155	0.158	0.162
17	LSTM		07:44	0.199	0.202	0.205	0.201	0.210	0.212	0.210	0.220
18	BiLSTM		11:02	0.683	0.701	0.706	0.710	0.709	0.694	0.675	0.658
19	Persistence		–	0.143	0.151	0.158	0.165	0.171	0.178	0.184	0.190
20	Naive		–	0.163	0.169	0.175	0.181	0.187	0.193	0.198	0.203
21	ST-MDF*	8, 8, 8	14:44	0.162	0.160	0.158	0.158	0.158	0.160	0.162	0.165
22	LSTM		07:48	0.170	0.166	0.162	0.161	0.160	0.160	0.159	0.160
23	BiLSTM		11:18	0.139	0.141	0.142	0.144	0.145	0.146	0.148	0.150
24	Persistence		–	0.171	0.178	0.184	0.190	0.194	0.200	0.205	0.210
25	Naive		–	0.187	0.193	0.198	0.203	0.208	0.213	0.218	0.222

4.2.2. Error assessment

In this section, the errors obtained by several variations of the model during the testing phase are presented. Firstly, the error metrics are formulated and their implementation is described. Afterward, the proposed model for multiple mobility services is compared to the same models for separated services. Then, the model is also compared to several baselines and models from the literature. Finally, the impact of two aspects is studied: the size of the input window n_x and the modules that constitute the proposed model.

Error metrics. Throughout the experiments, the error is evaluated using the RMSE for the whole testing period. MAE was also calculated and made available in a repository (see Appendix), but the insights extracted from it are similar. The RMSE (2) and MAE (3) metrics are expressed as follows:

$$\text{RMSE}\left(y, y^*\right) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y_i - y_i^*\right)^2}, \tag{2}$$

$$\text{MAE}\left(y, y^*\right) = \frac{1}{n}\sum_{i=1}^{n}|y_i - y_i^*|, \tag{3}$$

where y are the predicted values and y^* the truth ones. The error is always evaluated on the locations of the mesh-grid that contain any bike station or taxi zone centroid. This provides a more representative evaluation than if the error was considered in the whole mesh-grid, since some of the areas fall outside the city of Chicago or on Lake Michigan (see Fig. 1). Therefore, for each mobility service, the error is calculated as a table with n_y rows and as many columns as the taxi or grid zones that the mobility service has. In the interest of reflecting the performance on the most relevant zones from the point of view of mobility demand, the error is multiplied by normalized weights that are obtained as the average number of trips per 15 min on each zone, during the whole temporal coverage.

Comparison with separated mobility services. Section 4.1.1 detailed how different mobility services (taxi and bike rides) were combined into a single data structure. To investigate whether this strategy overall benefits the results, two ST-MDF$^*_{4,4,4}$ models were trained for separated mobility services: one to predict taxi rides and one for bicycle trips. In the first case, the model that works with both mobility services at once obtained 2.7, 3.5, 4, and 4.2% better results on each horizon, respectively. Similar results were observed for bike rides, where the improvements of the proposed model were of 5.6, 5.4, 5.1, and 4.8% per horizon. The RMSE scored by these models can be observed in Table 2.

Comparison with baselines. The proposed baselines are:

- LSTM network: It has a very similar structure to the ST-MDF* model that includes 3 modules (see Fig. 3), with the difference that instead of a convolutional LSTM layer, it has an LSTM layer with 16 neurons and that instead of a deconvolution it includes an additional dense layer.
- BiLSTM network: It has the same structure as the baseline LSTM, but instead of an LSTM with 16 neurons, it has 8 bidirectional LSTM neurons that return 16 values (8 in the forward pass and 8 more in the backward pass).

Table 4
Error assessment of the developed models and baselines for the bicycle rides task. Given s, the forecast horizon is obtained as: $h_i = 15 \cdot (i + s)$ min, with $, i \in \{0, \dots, 7\}$. Training time is displayed in `days hh:mm` format.

#	Model	n_x, n_y, s	Training time	RMSE [–]							
				h_0	h_1	h_2	h_3	h_4	h_5	h_6	h_7
1	ST-MDF*		07:35	0.085	0.086	0.086	0.086				
2	LSTM		04:24	0.136	0.135	0.144	0.142				
3	BiLSTM	4, 4, 4	05:29	0.080	0.082	0.082	0.082	–	–	–	–
4	Persistence		–	0.079	0.084	0.088	0.092				
5	Naive		–	0.075	0.079	0.083	0.086				
6	ST-MDF*		09:40	0.085	0.085	0.086	0.086	0.086	0.086	0.086	0.087
7	LSTM		05:44	0.080	0.083	0.081	0.080	0.080	0.086	0.084	0.082
8	BiLSTM	4, 8, 4	07:50	0.082	0.082	0.083	0.084	0.085	0.086	0.087	0.087
9	Persistence		–	0.079	0.084	0.088	0.092	0.095	0.097	0.099	0.101
10	Naive		–	0.075	0.079	0.083	0.086	0.088	0.09	0.092	0.093
11	ST-MDF*		13:05	0.085	0.086	0.086	0.086				
12	LSTM		06:42	0.091	0.092	0.092	0.092				
13	BiLSTM	8, 4, 4	08:18	0.099	0.096	0.096	0.096	–	–	–	–
14	Persistence		–	0.080	0.084	0.088	0.092				
15	Naive		–	0.078	0.081	0.083	0.086				
16	ST-MDF*		14:52	0.085	0.085	0.086	0.086	0.086	0.086	0.086	0.087
17	LSTM		07:44	0.097	0.094	0.099	0.093	0.092	0.092	0.090	0.090
18	BiLSTM	8, 8, 4	11:02	0.301	0.309	0.306	0.294	0.280	0.261	0.248	0.243
19	Persistence		–	0.080	0.084	0.088	0.092	0.095	0.097	0.099	0.101
20	Naive		–	0.078	0.081	0.083	0.086	0.088	0.089	0.090	0.092
21	ST-MDF*		14:44	0.087	0.087	0.087	0.087	0.087	0.087	0.088	0.088
22	LSTM		07:48	0.087	0.084	0.081	0.080	0.078	0.077	0.078	0.079
23	BiLSTM	8, 8, 8	11:18	0.062	0.061	0.061	0.062	0.062	0.063	0.064	0.065
24	Persistence		–	0.095	0.097	0.099	0.101	0.102	0.103	0.104	0.105
25	Naive		–	0.088	0.089	0.090	0.092	0.093	0.094	0.094	0.095

- Persistence: It is based on the assumption that the forecasted variable will remain unchanged in the future. Hence, it can be expressed as $f(\{\mathcal{M}_{t_1}, \dots, \mathcal{M}_{t_{n_x}}\}) = \mathcal{M}_{t_{n_x}}$.
- Naive: Its prediction is the point-wise average of the input window, which can be written as: $f(\{\mathcal{M}_{t_1}, \dots, \mathcal{M}_{t_{n_x}}\}) = \frac{1}{n_x} \sum_{i=1}^{n_x} \mathcal{M}_{t_i}$.

Note that the persistence and naive models do not need to be trained, since they calculate the output directly from the input mobility mesh-grids with a formula. Table 3 presents the RMSE scored by the ST-MDF* and the baseline models for different combinations of n_x, n_y and s for the taxi trips forecasting task. Table 4 also presents the RMSE for the same models, but is evaluated for the bike rides forecasting problem. Hence, the ith row of Table 3 presents the results for the same model of the ith row of Table 4, evaluated for different mobility services. The error is evaluated on each forecast horizon (based on n_y and s), and the mean values across zones or bike stations are shown in the tables.

Comparison with the literature. Finding works that are comparable to the proposed framework is not trivial for several reasons:

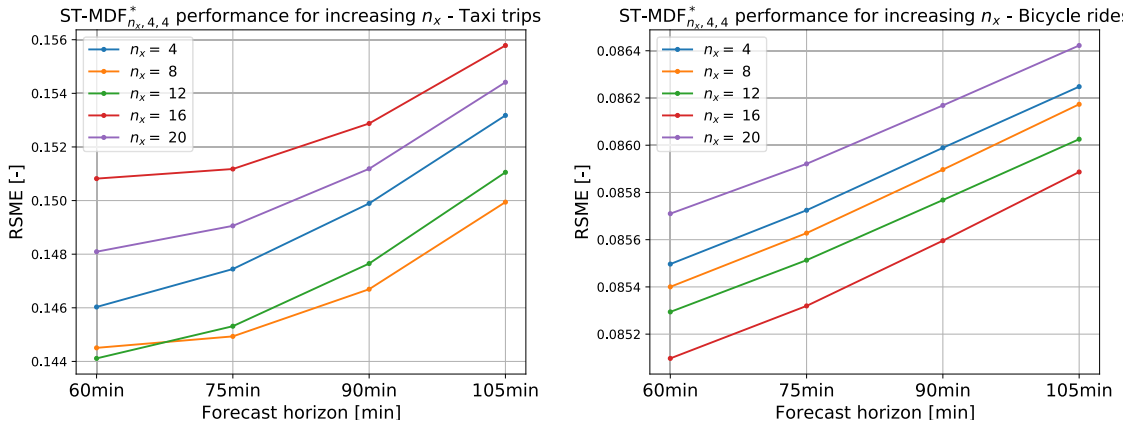
- Most works do not disclose the code of the experiments.
- To the best of the authors' knowledge, no other work had previously combined taxi and bike trips in a single modeling framework.
- The temporal granularity can differ from 15 min.
- The forecast horizons can be different from the studied ones (between 1 h and 3 h 45 min).
- The target city, temporal coverage, and train/test separation are also prone to be different.
- Other relevant aspects that can differ are the error metric, the usage of weather or temporal variables, mesh-grid shape, data preprocessing steps, etc.

Table 5 presents the (unweighted) RMSE of the ST-MDF$^*_{4,4,4}$ model for taxi rides along with the model and the baselines presented by Liu et al. [29]. Furthermore, Xu et al. [27] present the RMSE of their LSTM-based model per hour for the prediction of taxi demand in New York City (figure 11). It can be derived that their average RMSE is above 2.5, which is consistent with the results of Table 5. Additionally, Yao et al. [3] present the results of their proposal and several baselines for the city of Guangzhou, obtaining RMSEs around 10 (Table 1). As for bicycle rides, Table 6 presents the (unweighted) RMSE of the ST-MDF$^*_{4,4,4}$ model along with the model and the baselines presented by Chai et al. [31]. In their work, bike rides from Chicago are studied and the forecast horizon is 1 h. Their model is based on multi-graph Convolutional Neural Networks (CNNs), as discussed in Section 2. The RMSE is presented as the average between the 5 and 10 stations with the highest demand, as well as the average between all stations.

Table 5
Error assessment compared to the results and baselines of Liu et al. [29]. In this case, the error is **not** weighed (as in Tables 2, 3 and 4) to be comparable with the other works. Context-aware Attention-based Convolutional Recurrent Neural Network (CACRNN) is the model developed by Liu et al. [29], and the baselines they present are Spatial–Temporal Graph Convolutional Network (STGCN), Diffusion Convolution Recurrent Neural Network (DCRNN), LSTM, Autoregressive Integrated Moving Average (ARIMA), and Historical Average (HA). In the comparison, the different forecast horizons and target cities should be considered.

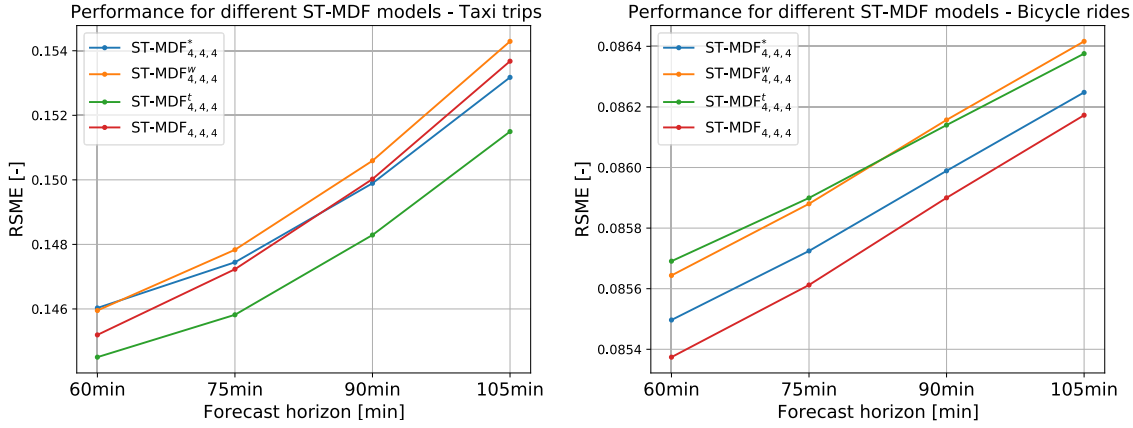
#	Model	Evaluated service	Forecast horizon	City	RMSE [–]
1	ST-MDF*	Taxi rides	1 h	Chicago	2.43
2	CACRNN	Taxi rides	15 min	NYC	3.17
3	STGCN				3.48
4	DCRNN				3.50
5	LSTM				3.65
6	ARIMA				9.53
7	HA				5.84
8	CACRNN	Taxi rides	15 min	Chengdu	2.73
9	STGCN				2.77
10	DCRNN				3.14
11	LSTM				4.30
12	ARIMA				5.75
13	HA				7.10

Impact of n_x and the ST-MDF* modules. To understand the impact of the parameter n_x, several ST-MDF* models were trained with the same n_y and s parameters, but with increasing n_x. The results are shown in Fig. 4 for the ST-MDF$^*_{n_x,4,4}$. Furthermore, the impact of the three different modules that constitute the ST-MDF* model was evaluated



(a) Results for taxi trips. (b) Results for bicycle rides.

Fig. 4. Error comparison for ST-MDF$^{*}_{n_x,4,4}$, where $n_x \in \{4, 8, 12, 16, 20\}$, i.e. input windows of 1 h, 2 h, 3 h, 4 h and 5 h, respectively.



(a) Results for taxi trips. (b) Results for bicycle rides.

Fig. 5. Error comparison between the ST-MDF*, ST-MDF^{w}, ST-MDF^{t}, and ST-MDF models. The experiment was carried out for $n_x = 4, n_y = 4$ and $s = 4$.

Table 6
Error assessment compared to the results and baselines of Chai et al. [31]. "Top n stations" means that the RMSE is evaluated on the n stations with higher demand. In this case, the error is **not** weighed (as in Tables 2, 3 and 4) to be comparable with the other works. Multi-graph CNN is the model developed by Chai et al. [31], and the baselines they present are LSTM, GBRT (Gradient Boosting Regression Tree), SARIMA (seasonal ARIMA), ARIMA (Auto-Regressive Integrated Moving Average), and HM (Historical Mean).

#	Model	Evaluated service	RMSE [–]		
			Top 5 stations	Top 10 stations	Average
1	ST-MDF*	Bike trips	5.331	4.646	1.702
2	Multi-graph CNN	Bike trips	5.177	4.930	3.658
3	LSTM		6.231	5.853	4.405
4	GBRT		5.945	5.738	4.410
5	SARIMA		6.797	6.175	4.608
6	ARIMA		9.853	8.535	6.163
7	HM		7.078	6.179	4.347

for $n_x = 4, n_y = 4$ and $s = 4$ (see Fig. 5). ST-MDF^{t} corresponds to the same model as the ST-MDF* but includes only the mobility and temporal modules. Similarly, ST-MDF^{w} includes the mobility and weather modules, and ST-MDF only includes the mobility module.

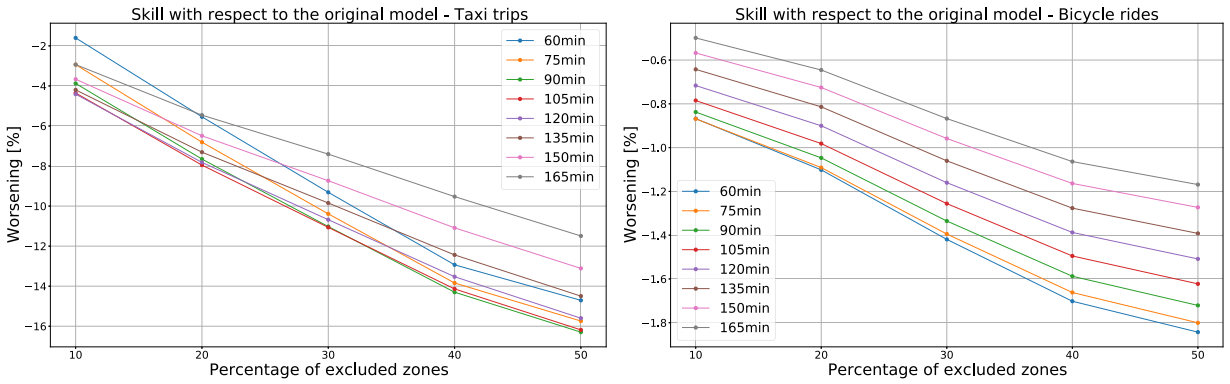
4.2.3. Flexibility assessment

Earlier in this work, it was discussed that features beyond the error should be analyzed when modeling spatio-temporary fused data. Flexibility is defined as the ability to accept a different number of inputs and outputs. For the current use case, this implies that any variation in the number or distribution of taxi zones or bike racks should not mean that the system needs to be redefined and trained from scratch. Reviewing how the mobility mesh-grid is built (see Section 4.1.1), it can be seen that the interpolation step converts the (variable) number of stations/racks into the fixed-size mesh-grid. In this way, the input to the model is decoupled from the number of zones, bike stations, and their location. Therefore, taxi zones and bicycle racks can be added or removed without modifying the ST-MDF* model.

4.2.4. Robustness assessment

Generally, robustness refers to the properties of a system that allow it to tolerate disruptions in the external environment without malfunctioning or changing its structure or dynamics [47]. In the context of this work, it is defined as the ability to recover from sensor failure and continue producing reliable predictions. If we consider each bike rack as a sensor, one may expect that some of them may fail for some periods. In this situation, a robust forecasting framework should be able to recover an approximation of that missing data. Thanks to the interpolation method described in Section 4.1.1, the gaps in the mesh-grid can be filled using neighboring information with a realistic approximation of their mobility demand, allowing the model to produce reliable predictions. This situation was simulated for the ST-MDF* to evaluate its robustness.

Table 7 shows how the error evolves when a percentage of bike racks or taxi zones stop working. The percentage of failing sensors



(a) Results for taxi trips (2nd row of Table 7). (b) Results for bicycle rides (7th row of Table 7).

Fig. 6. Robustness assessment for the ST-MDF$^{*}_{4,8,4}$, i.e., error worsening when missing an increasing percentage of data.

Table 7
Robustness assessment for the ST-MDF*, both for taxi and bicycle trips. Worsening on the test set when missing an increasing percentage of data. The test was executed independently for five models and 20 random repetitions for each percentage. The results are shown separately for taxis and bicycles.

#	Model	Target	n_x, n_y, s	Worsening [%] when missing				
				10%	20%	30%	40%	50%
1			4, 4, 4	−11.7	−15.3	−15.1	−25.3	−26.3
2			4, 8, 4	−3.5	−6.9	−9.8	−12.7	−14.7
3	ST-MDF*	Taxi trips	8, 4, 4	−5.6	−13.8	−20.3	−23.7	−29.2
4			8, 8, 4	−5.3	−8.3	−11.6	−13.3	−15.6
5			8, 8, 8	−5.8	−10.3	−8.7	−12.9	−16.7
1			4, 4, 4	−1.3	−1.6	−1.6	−2.2	−2.2
2			4, 8, 4	−0.7	−0.9	−1.2	−1.4	−1.5
3	ST-MDF*	Bicycle rides	8, 4, 4	−0.8	−1.3	−1.6	−1.9	−2.2
4			8, 8, 4	−0.5	−0.7	−0.9	−1.2	−1.4
5			8, 8, 8	−0.1	−0.4	−0.2	−0.5	−0.5

ranges from 10% to 50% (with steps of 10%), and for each case 20 independent repetitions were executed removing random racks for the testing period (from the beginning of 2019 to mid-March 2020). Additionally, Fig. 6 illustrates the worsening per percentage of excluded zones on every forecast horizon for the ST-MDF$^{*}_{4,8,4}$ model.

5. Discussion

The proposed ST-MDF* model combines heterogeneous spatio-temporal data and provides robustness and flexibility while obtaining smaller RMSE when compared to several baseline models. Section 4.2.2 presents the error results and explains how it was computed. Tables 3 and 4 show that the ST-MDF* model beats the proposed baseline models, except for a few cases with small forecast horizons. The **skill** or relative improvement can be expressed as:

$$S = \left(1 - \frac{\text{error}_{\text{proposed}}}{\text{error}_{\text{reference}}}\right) \cdot 100 \tag{4}$$

Taking the persistence model as reference (which is the best baseline in most cases, according to the combination of Tables 3 and 4), it can be seen that the proposed model gets skills of up to 21.1% for taxi trips and 16.5% for bike rides. Furthermore, Section 4.2.2 shows that the ST-MDF* model benefits from combining both mobility services into a single data structure. The RMSE of this model is 3.6% and 5.2% better on average than the same model for only taxis or bicycles, respectively. This suggests that the predictions of one mobility service can benefit from the data of a different one. Intuitively, if the number of bicycle rides increases for a given period and location, it can mean that the demand for taxis will decrease in the same area.

Interestingly, the analysis of Fig. 4 shows that depending on whether the targets are taxi or bike rides, different n_x are optimal. For the former, $n_x = 8$ provides the smallest error, while for the latter, $n_x = 16$ works better (fixing $n_y = 4$ and $s = 4$). This may be due to the difference between the number of taxis and bikes rides, and because of the more stable dynamics or the taxi ride problem. Additionally, the error per horizon is rather consistent with the choice of the parameter n_x.

To see the impact of the different modules described in Section 3.4, Fig. 5 compares ST-MDF* with the ST-MDF^{t}, ST-MDF^{w} and ST-MDF models, that is, when weather, temporal or both modules are removed, respectively. Fig. 5(a) shows that the ST-MDF^{t} model consistently outperforms its three peers for taxi trips. However, the opposite happens for bike rides (Fig. 5(b)). As for the ST-MDF model, even if it comes first for this second task, its performance for taxi trips is poorer (especially for the farthest horizons). A similar situation can be observed for the ST-MDF^{w} model for the second problem in Fig. 5(b). Therefore, ST-MDF* seems to provide the most consistent behavior among its peers when considering both mobility prediction tasks. Furthermore, its RMSE increase for farther horizons is less pronounced than for the other three models, in both mobility cases.

Thanks to the mobility mesh-grid, the ST-MDF* model is flexible enough to work with a different number of input sources (taxi zones and bicycle racks). Therefore, when such changes occur, the model is ready to work with the new ones. Furthermore, the ST-MDF* model is robust so that missing information can be dynamically filled with the aid of the mobility mesh-grid, as Section 4.2.4 experiments show. For instance, Fig. 6 (left) reveals that RMSE deteriorates only 11% or less when randomly removing 30% of taxi data during the testing phase for ST-MDF$^{*}_{4,8,4}$. Similarly, the RMSE worsens less than 1.4% for bike rides when up to 30% of the testing data is missing (see Fig. 6, right). The reason for the contrast in robustness between these two mobility services is likely to be the different volume of trips (the number of taxi trips is over 7 times higher than that of the bicycles for the whole studied period).

As mentioned in Section 4.2, the testing period was trimmed when the COVID-19 pandemics began, since the lockdown greatly changed the mobility habits of people, and this was evident in the error assessment. In any case, if such situations arise, any of the presented models could be tuned to adapt to the new mobility dynamics.

6. Conclusions and future work

The proposed mobility demand mesh-grid captures the relevant information to model transport demand in an intuitive way. The proposed Spatio-Temporal Mobility Demand Forecaster (ST-MDF) model combines the demand mesh-grid for two mobility services (namely taxi and bicycle rides), weather, and temporal information, to forecast the future mobility demand of the same mobility services on several horizons. In this way, a global mobility model is trained for the whole city with

fine spatio-temporal resolution. Furthermore, the proposed model can adjust to the number of data sources that feed it, i.e., the number of taxi zones and bicycle racks, which shows its flexibility. Besides, when some of those data sources fail and those data are missing, it can be filled thanks to the use of the mobility mesh-grid. Lastly, the proposed model beats the baselines, such as a Long Short-Term Memory (LSTM) network, naive, and persistence models.

Regarding future work, a deeper study than the one of Section 4.2.2 about the influence of the parameter n_x could lead to an increase in the performance of the ST-MDF* model (for example, incorporating n_x into the temporal module). Furthermore, the model could be expanded with additional data sources, such as for-hire vehicle (FHV) data, metro and bus transit data. The integration of additional weather stations could also benefit the model (potentially as a mesh-grid), since meteorological conditions can vary in a large area, such as the studied one. Also, the use of Graph Neural Networks could be useful, which is already used in some related tasks, such as traffic forecasting [48], or just ride-hailing [49,50]. Yet another extension to this work is the forecasting of drop-off zones in addition to the pick-up ones. This would provide information about the distribution of taxis once they finish their current trips, helping to organize the vehicle fleet. Similarly, the end bike station could be forecasted for the management of its distribution. Moreover, Points Of Interest (POIs) could be included as an additional layer of the mobility mesh-grid, or as a separate module of the ST-MDF*. Their spatio-temporal nature could reduce the error when particular events take place in the city (e.g. concerts or sports matches).

Lastly, besides flexibility and robustness, some additional important features that will be studied in future work include:

- **Extensibility**: Capability to include new data sources of a different nature. For instance, trips by scooter, bus, and metro.
- **Scalability**: Ability to cope with the computational demands of an increase in the number of sensors of the same nature. An example is the addition of FHV trips to the taxi dataset, without a notable degradation in training and prediction times.
- **Portability**: Possibility to use the same model architecture for multiple different locations (e.g. New York). In this case, the model would be trained from scratch with new data while maintaining the same structure.
- **Transferability**: Feasibility of using transfer learning to employ a model that has been trained for a certain place in a different one. This would imply tuning the outer layers of the model with a smaller dataset from the target location, reducing training times and development effort.

CRediT authorship contribution statement

Ignacio-Iker Prado-Rujas: Conceptualization, Methodology, Software, Formal analysis, Investigation, Data curation, Writing – original draft, Writing – review & editing, Visualization. **Emilio Serrano:** Conceptualization, Methodology, Investigation, Writing – original draft, Writing – review & editing, Visualization. **Antonio García-Dopico:** Conceptualization, Methodology, Investigation, Resources, Writing – original draft, Writing – review & editing, Visualization. **M. Luisa Córdoba:** Conceptualization, Methodology, Investigation, Resources, Writing – original draft, Writing – review & editing, Visualization. **María S. Pérez:** Conceptualization, Methodology, Investigation, Writing – original draft, Writing – review & editing, Visualization, Supervision, Funding acquisition, Project administration.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This work has been supported by the Autonomous Region of Madrid (Spain) through the program CABAHLA-CM (GA No. P2018/TCS-4423). This work is also funded in part by the Euratom research and training program 2019-2020 ENTENTE (GA No. 900018) and by Knowledge Spaces project (Grant PID2020-118274RB-I00 funded by MCIN/AEI/10.13039/501100011033).

Appendix. Complete results, datasets and code

The complete set of graphs, the code to produce the datasets, and the code to reproduce the experiments are available at https://github.com/iipr/mobility-demand. Additionally, the mobility mesh-grid datasets and other complementary ones are available at 10.5281/zenodo.5166839.

References

[1] D. Fiorello, L. Zani, et al., EU survey on issues related to transport and mobility, JRC Sci. Policy Rep. (2015) http://dx.doi.org/10.2791/48322.
[2] E.I. Vlahogianni, M.G. Karlaftis, J.C. Golias, Short-term traffic forecasting: Where we are and where we're going, Transp. Res. C 43 (2014) 3–19, http://dx.doi.org/10.1016/j.trc.2014.01.005, Special Issue on Short-term Traffic Flow Forecasting.
[3] H. Yao, F. Wu, J. Ke, X. Tang, Y. Jia, S. Lu, P. Gong, J. Ye, Z. Li, Deep multi-view spatial-temporal network for taxi demand prediction, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 32, 2018.
[4] J. Zhang, Y. Zheng, D. Qi, R. Li, X. Yi, DNN-based prediction model for spatio-temporal data, in: Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, 2016, pp. 1–4, http://dx.doi.org/10.1145/2996913.2997016.
[5] Z. Cui, R. Ke, Z. Pu, X. Ma, Y. Wang, Learning traffic as a graph: A gated graph wavelet recurrent neural network for network-scale traffic prediction, Transp. Res. C 115 (2020) 102620, http://dx.doi.org/10.1016/j.trc.2020.102620.
[6] J. Liu, T. Li, P. Xie, S. Du, F. Teng, X. Yang, Urban big data fusion based on deep learning: An overview, Inf. Fusion 53 (2020) 123–133, http://dx.doi.org/10.1016/j.inffus.2019.06.016.
[7] P. Xie, T. Li, J. Liu, S. Du, X. Yang, J. Zhang, Urban flow prediction from spatiotemporal data using machine learning: A survey, Inf. Fusion 59 (2020) 1–12, http://dx.doi.org/10.1016/j.inffus.2020.01.002.
[8] M.X. Hoang, Y. Zheng, A.K. Singh, FCCF: forecasting citywide crowd flows based on big data, in: Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, 2016, pp. 1–10, http://dx.doi.org/10.1145/2996913.2996934.
[9] L. Khaidem, M. Luca, F. Yang, A. Anand, B. Lepri, W. Dong, Optimizing transportation dynamics at a city-scale using a reinforcement learning framework, IEEE Access 8 (2020) 171528–171541, http://dx.doi.org/10.1109/ACCESS.2020.3024979.
[10] J.A. Deri, F. Franchetti, J.M. Moura, Big data computation of taxi movement in New York City, in: 2016 IEEE International Conference on Big Data (Big Data), IEEE, 2016, pp. 2616–2625, http://dx.doi.org/10.1109/BigData.2016.7840904.
[11] M. Lippi, M. Bertini, P. Frasconi, Short-term traffic flow forecasting: An experimental comparison of time-series analysis and supervised learning, IEEE Trans. Intell. Transp. Syst. 14 (2) (2013) 871–882, http://dx.doi.org/10.1109/TITS.2013.2247040.
[12] N.G. Polson, V.O. Sokolov, Deep learning for short-term traffic flow prediction, Transp. Res. C 79 (2017) 1–17, http://dx.doi.org/10.1016/j.trc.2017.02.024.
[13] X. Ma, J. Zhang, B. Du, C. Ding, L. Sun, Parallel architecture of convolutional bi-directional LSTM neural networks for network-wide metro ridership prediction, IEEE Trans. Intell. Transp. Syst. 20 (6) (2018) 2278–2288, http://dx.doi.org/10.1109/TITS.2018.2867042.
[14] L. Lin, Z. He, S. Peeta, Predicting station-level hourly demand in a large-scale bike-sharing network: A graph convolutional neural network approach, Transp. Res. C 97 (2018) 258–276, http://dx.doi.org/10.1016/j.trc.2018.10.011.
[15] Z. Jiang, W. Fan, W. Liu, B. Zhu, J. Gu, Reinforcement learning approach for coordinated passenger inflow control of urban rail transit in peak hours, Transp. Res. C 88 (2018) 1–16, http://dx.doi.org/10.1016/j.trc.2018.01.008.
[16] Y. Ma, T. Lin, Z. Cao, C. Li, F. Wang, W. Chen, Mobility viewer: An Eulerian approach for studying urban crowd flow, IEEE Trans. Intell. Transp. Syst. 17 (9) (2015) 2627–2636, http://dx.doi.org/10.1109/TITS.2015.2498187.

[17] N. Zhang, Y. Zhang, H. Lu, Seasonal autoregressive integrated moving average and support vector machine models: prediction of short-term traffic flow on freeways, Transp. Res. Rec. 2215 (1) (2011) 85–92, http://dx.doi.org/10.3141/2215-09.

[18] X. Li, G. Pan, Z. Wu, G. Qi, S. Li, D. Zhang, W. Zhang, Z. Wang, Prediction of urban human mobility using large-scale taxi traces and its applications, Front. Comput. Sci. 6 (1) (2012) 111–121.

[19] L. Moreira-Matias, J. Gama, M. Ferreira, L. Damas, A predictive model for the passenger demand on a taxi network, in: 2012 15th International IEEE Conference on Intelligent Transportation Systems, 2012, pp. 1014–1019, http://dx.doi.org/10.1109/ITSC.2012.6338680.

[20] S. Ma, Y. Zheng, O. Wolfson, T-share: A large-scale dynamic taxi ridesharing service, in: 2013 IEEE 29th International Conference on Data Engineering, ICDE, 2013, pp. 410–421, http://dx.doi.org/10.1109/ICDE.2013.6544843.

[21] F.G. Habtemichael, M. Cetin, Short-term traffic flow rate forecasting based on identifying similar traffic patterns, Transp. Res. C 66 (2016) 61–78, http://dx.doi.org/10.1016/j.trc.2015.08.017, Advanced Network Traffic Management: From dynamic state estimation to traffic control.

[22] J. Roos, G. Gavin, S. Bonnevay, A dynamic Bayesian network approach to forecast short-term urban rail passenger flows with incomplete data, Transp. Res. Procedia 26 (2017) 53–61, http://dx.doi.org/10.1016/j.trpro.2017.07.008, Emerging technologies and models for transport and mobility.

[23] J. Zhang, X. Pan, M. Li, S.Y. Philip, Bicycle-sharing system analysis and trip prediction, in: 2016 17th IEEE International Conference on Mobile Data Management, MDM, IEEE, 2016, pp. 174–179, http://dx.doi.org/10.1109/MDM.2016.35.

[24] N. Boufidis, A. Nikiforiadis, K. Chrysostomou, G. Aifadopoulou, Development of a station-level demand prediction and visualization tool to support bike-sharing systems' operators, Transp. Res. Procedia 47 (2020) 51–58, http://dx.doi.org/10.1016/j.trpro.2020.03.072.

[25] X. Qian, S.V. Ukkusuri, C. Yang, F. Yan, Short-term demand forecasting for on-demand mobility service, IEEE Trans. Intell. Transp. Syst. (2020) http://dx.doi.org/10.1109/TITS.2020.3019509.

[26] L. Wang, X. Geng, X. Ma, F. Liu, Q. Yang, Cross-city transfer learning for deep spatio-temporal prediction, in: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19, International Joint Conferences on Artificial Intelligence Organization, 2019, pp. 1893–1899, http://dx.doi.org/10.24963/ijcai.2019/262.

[27] J. Xu, R. Rahmatizadeh, L. Bölöni, D. Turgut, Real-time prediction of taxi demand using recurrent neural networks, IEEE Trans. Intell. Transp. Syst. 19 (8) (2017) 2572–2581, http://dx.doi.org/10.1109/TITS.2017.2755684.

[28] F. Rodrigues, I. Markou, F.C. Pereira, Combining time-series and textual data for taxi demand prediction in event areas: A deep learning approach, Inf. Fusion 49 (2019) 120–129, http://dx.doi.org/10.1016/j.inffus.2018.07.007.

[29] T. Liu, W. Wu, Y. Zhu, W. Tong, Predicting taxi demands via an attention-based convolutional recurrent neural network, Knowl.-Based Syst. 206 (2020) 106294, http://dx.doi.org/10.1016/j.knosys.2020.106294.

[30] W. Jiang, Bike sharing usage prediction with deep learning: a survey, Neural Comput. Appl. (2022) http://dx.doi.org/10.1007/s00521-022-07380-5.

[31] D. Chai, L. Wang, Q. Yang, Bike flow prediction with multi-graph convolutional networks, in: Proceedings of the 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, SIGSPATIAL '18, Association for Computing Machinery, New York, NY, USA, 2018, pp. 397–400, http://dx.doi.org/10.1145/3274895.3274896.

[32] X. Li, Y. Xu, Q. Chen, L. Wang, X. Zhang, W. Shi, Short-term forecast of bicycle usage in bike sharing systems: A spatial-temporal memory network, IEEE Trans. Intell. Transp. Syst. (2021) 1–12, http://dx.doi.org/10.1109/TITS.2021.3097240, Conference Name: IEEE Transactions on Intelligent Transportation Systems.

[33] E. Walraven, M.T. Spaan, B. Bakker, Traffic flow optimization: A reinforcement learning approach, Eng. Appl. Artif. Intell. 52 (2016) 203–212, http://dx.doi.org/10.1016/j.engappai.2016.01.001.

[34] H. Wei, G. Zheng, H. Yao, Z. Li, IntelliLight: A reinforcement learning approach for intelligent traffic light control, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD '18, Association for Computing Machinery, 2018, pp. 2496–2505, http://dx.doi.org/10.1145/3219819.3220096.

[35] F. Miao, S. Han, S. Lin, J.A. Stankovic, D. Zhang, S. Munir, H. Huang, T. He, G.J. Pappas, Taxi dispatch with real-time sensing data in metropolitan areas: A receding horizon control approach, IEEE Trans. Autom. Sci. Eng. 13 (2) (2016) 463–478, http://dx.doi.org/10.1109/TASE.2016.2529580.

[36] C. Tian, X. Zhu, Z. Hu, J. Ma, A transfer approach with attention reptile method and long-term generation mechanism for few-shot traffic prediction, Neurocomputing 452 (2021) 15–27, http://dx.doi.org/10.1016/j.neucom.2021.03.068.

[37] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-k. Wong, W.-c. Woo, Convolutional LSTM network: A machine learning approach for precipitation nowcasting, in: Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1, NIPS '15, MIT Press, Cambridge, MA, USA, 2015, pp. 802–810, URL: https://dl.acm.org/doi/abs/10.5555/2969239.2969329.

[38] S. Agrawal, L. Barrington, C. Bromberg, J. Burge, C. Gazen, J. Hickey, Machine learning for precipitation nowcasting from radar images, 2019, arXiv preprint arXiv:1912.12132.

[39] I.-I. Prado-Rujas, A. García-Dopico, E. Serrano, M.S. Pérez, A flexible and robust deep learning-based system for solar irradiance forecasting, IEEE Access 9 (2021) 12348–12361, http://dx.doi.org/10.1109/ACCESS.2021.3051839.

[40] J. Guo, W. Huang, B.M. Williams, Adaptive Kalman filter approach for stochastic short-term traffic flow rate prediction and uncertainty quantification, Transp. Res. C 43 (2014) 50–64, http://dx.doi.org/10.1016/j.trc.2014.02.006, Special Issue on Short-term Traffic Flow Forecasting.

[41] T.D. Wemegah, S. Zhu, C. Atombo, Modeling the effect of days and road type on peak period travels using structural equation modeling and big data from radio frequency identification for private cars and taxis, Eur. Transp. Res. Rev. 10 (2) (2018) 1–14, http://dx.doi.org/10.1186/s12544-018-0313-9.

[42] M.D. Zeiler, D. Krishnan, G.W. Taylor, R. Fergus, Deconvolutional networks, in: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2010, pp. 2528–2535, http://dx.doi.org/10.1109/CVPR.2010.5539957.

[43] V. Dumoulin, F. Visin, A guide to convolution arithmetic for deep learning, 2018, http://dx.doi.org/10.48550/arXiv.1603.07285, URL: http://arxiv.org/abs/1603.07285, arXiv:1603.07285 [cs, stat].

[44] Divvy (powered by Lyft), Divvy dataset, 2020, https://www.divvybikes.com/system-data.

[45] F. Chollet, et al., Keras, 2015, https://keras.io.

[46] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G.S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, X. Zheng, TensorFlow: Large-scale machine learning on heterogeneous systems, 2015, URL: https://www.tensorflow.org/. Software available from https://tensorflow.org.

[47] C. Sergiou, M. Lestas, P. Antoniou, C. Liaskos, A. Pitsillides, Complex systems: A communication networks perspective towards 6G, IEEE Access 8 (2020) 89007–89030, http://dx.doi.org/10.1109/ACCESS.2020.2993527.

[48] Y. Li, R. Yu, C. Shahabi, Y. Liu, Diffusion convolutional recurrent neural network: Data-driven traffic forecasting, in: International Conference on Learning Representations (ICLR '18), 2018.

[49] X. Geng, Y. Li, L. Wang, L. Zhang, Q. Yang, J. Ye, Y. Liu, Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33, 2019, pp. 3656–3663, http://dx.doi.org/10.1609/aaai.v33i01.33013656.

[50] G. Jin, Y. Cui, L. Zeng, H. Tang, Y. Feng, J. Huang, Urban ride-hailing demand prediction with multiple spatio-temporal information fusion network, Transp. Res. C 117 (2020) 102665, http://dx.doi.org/10.1016/j.trc.2020.102665.