A Survey on Deep Learning for Human Mobility

MASSIMILIANO LUCA, Fondazione Bruno Kessler (FBK), Italy and Free University of Bolzano, Italy

GIANNI BARLACCHI*, Amazon Alexa, Germany

BRUNO LEPRI, Fondazione Bruno Kessler (FBK), Italy

LUCA PAPPALARDO, Institute of Information Science and Technologies, National Research Council (ISTI-CNR), Italy

The study of human mobility is crucial due to its impact on several aspects of our society, such as disease spreading, urban planning, well-being, pollution, and more. The proliferation of digital mobility data, such as phone records, GPS traces, and social media posts, combined with the predictive power of artificial intelligence, triggered the application of deep learning to human mobility. Existing surveys focus on single tasks, data sources, mechanistic or traditional machine learning approaches, while a comprehensive description of deep learning solutions is missing. This survey provides a taxonomy of mobility tasks, a discussion on the challenges related to each task and how deep learning may overcome the limitations of traditional models, a description of the most relevant solutions to the mobility tasks described above and the relevant challenges for the future. Our survey is a guide to the leading deep learning solutions to next-location prediction, crowd flow prediction, trajectory generation, and flow generation. At the same time, it helps deep learning scientists and practitioners understand the fundamental concepts and the open challenges of the study of human mobility.

---

*Work done before joining Amazon.

Authors' addresses: Massimiliano Luca, Fondazione Bruno Kessler (FBK), Via Sommarive, 19, Povo - Trento, Italy, Free University of Bolzano, Piazza Domenicani, 3, Bolzano, Italy, mluca@fbk.eu; Gianni Barlacchi, Amazon Alexa, , Berlin, Germany, gianni.barlacchi@gmail.com; Bruno Lepri, Fondazione Bruno Kessler (FBK), Via Sommarive, 19, Povo - Trento, Italy, ; Luca Pappalardo, Institute of Information Science and Technologies, National Research Council (ISTI-CNR), Via G. Moruzzi 1, 56124, Pisa, Italy, luca.pappalardo@isti.cnr.it.

---

1 INTRODUCTION

Urban population is increasing strikingly and human mobility is becoming more complex and bulky, affecting crucial aspects of people lives such as the spreading of viral diseases (e.g., the COVID-19 pandemic) [101, 105, 127, 130, 141, 156], the behavior of people in case of natural disasters [83, 175, 199], the public and private transportation and the resulting traffic volumes [31, 58, 94, 153], the well-being of citizens [137, 176, 192], the severity of air pollution, energy and water consumption [19, 126, 178]. Furthermore, crowds' movement between cities is influenced by migrations from rural to urban areas, such as those induced by natural disasters, climate change, and conflicts [2, 70, 117, 144, 149, 166, 169].

Fortunately, policymakers are not unarmed in facing these challenges. The rise of ubiquitous computing (e.g., mobile phone, the Internet of Things, social media platforms) provides an always up-to-date and precise way to sense human movements at various temporal and spatial scales. Examples of mobility data include tracks from GPS devices embedded in smartphones [21, 106, 123, 143, 232], vehicles [13, 60, 133, 136] or boats [29, 57, 151, 210]; records produced by the communication between phones and the cellular network [18, 65]; and geotagged posts from social media platforms [17, 41, 88, 111, 146]. This deluge of digital data fostered a vast scientific production on various aspects of human mobility, such as the mining of trajectory data [84, 119, 193, 226, 230], the uncovering of the statistical patterns [9, 18, 65, 173], and the estimation of the privacy risk [45, 59, 139, 140, 142, 155]. The development of powerful Artificial Intelligence (AI) techniques and the availability of big mobility data offered unprecedented opportunities for researchers to use Deep Learning (DL) approaches to solve mobility-related challenges. In this survey, we focus on DL solutions to predict or generate human movements and exclude other approaches solving other problems, such as semantic enrichment of mobility data (e.g., predicting the purpose of movement) [150], home location detection [132], and population inference [46]. In particular, we focus on two categories of tasks: predictive and generative (see Figure 1). We discuss two predictive tasks, namely next-location prediction and crowd flow prediction, and two generative tasks, namely trajectory generation and flow generation.

Next-location prediction is about forecasting which location an individual will visit given historical data about their mobility. It is crucial in many applications such as travel recommendation, location-aware advertisements and geomarketing, early warning of potential public emergencies, and recommendation of friends in social network platforms [23, 203, 227, 228, 235]. Crowd flow prediction, instead, is the task of forecasting the incoming and outgoing flows of people on a geographic region, which has an impact on public safety, the definition of on-demand services, the management of land use, and traffic optimization [50, 81, 164, 204, 219]. Concerning generative tasks, trajectory generation deals with generating synthetic trajectories that can reproduce, realistically, the individual statistical patterns of human mobility [9, 74, 171, 194]. Flow generation deals with generating realistic flows among locations, given their characteristics and the distance among them, and without any knowledge about the real flows. Although approaches based on Machine Learning (ML) achieve good results in solving these four tasks [23, 50, 164, 203, 204, 219, 228], multiple reasons pushed researchers to adopt DL techniques, such as the ability to automatically extract relevant patterns from (un)structured and heterogeneous data, and the outstanding results obtained in other fields (e.g., computer vision, natural language processing).

Several survey papers that provide interesting perspectives on single mobility tasks [9, 50, 74, 100, 164, 165, 171, 204], mobility data sources [207, 228], or traditional ML approaches [74, 185, 194]. A few surveys discuss DL approaches to spatio-temporal data mining [197], tasks related to the smart city ecosystem [32] or traffic-related issues [86, 200], covering some aspects related to human mobility but without a specific focus on the challenges and solutions to mobility-related tasks.

In this survey, we discuss DL solutions to next-location prediction, crowd flow prediction, trajectory generation and flow generation, organizing them into a proper taxonomy and discussing why those solutions may overcome limitations of existing traditional models. In order to find relevant papers for these tasks, we searched on Scopus
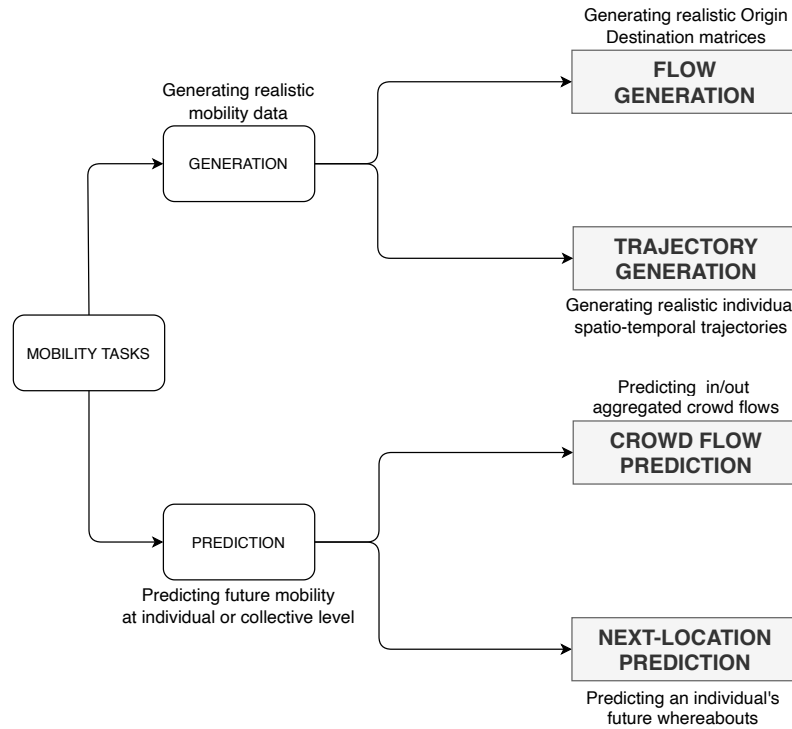
Fig. 1. A taxonomy of the mobility tasks we discuss in this survey. We classify mobility tasks in predictive, aiming at forecasting future mobility at an individual or collective level (Section 3), and generative, aiming at generating realistic trajectories of mobility flows (Section 4). Among the predictive tasks, we cover *i)* next-location prediction, the problem of forecasting future whereabouts given the mobility history of individuals (Section 3.1), and *ii)* crowd flow prediction, whose goal is to forecasting future aggregated flows given historical observations (Section 3.2). On the other hand, we have two generative tasks: *i)* trajectory generation aims at generating realistic individual trajectories (Section 4.1), and *ii)* flow generation whose goal is to generate realistic flows among locations on a geographic region (Section 4.2). We use this taxonomy to map relevant works to the task they solve and shape this survey's structure.

the following keywords: "crowd flow", "next-location", "flow prediction", "flow generation", "trajectory generation", "mobility generation", and "mobility prediction". Among the obtained results, we selected those using DL, which are commonly used in subsequent papers as baselines, which are seminal works for the task they address, and which add some novelty in terms of DL pipelines, how DL modules are combined and data handled. For each task, we also describe the open mobility datasets used by the papers and the evaluation metrics typically adopted. In the Appendix, we provide a more detailed description of deep learning modules (Appendix B), datasets (Appendix C), evaluation metrics (Appendix D), and mobility patterns' (Appendix E). In summary, this survey provides the reader with:

- An introduction to the fundamental concepts and nomenclature of human mobility (Section 2.1) and the key ideas behind the DL modules (Sections 2.2).
- A taxonomy of tasks related to predicting and generating human movements, with a comprehensive discussion on next-location prediction (Section 3.1), crowd flow prediction (Section 3.2), trajectory generation (Section 4.1), and flow generation (Section 4.2). For each task, we define the problem, discuss the DL modules

used in the literature to address it, we highlight the advantages of using DL over traditional models and list the public datasets and evaluation metrics commonly used for each task.

- A discussion of the most interesting open challenges about the four tasks (Section 5). We also provide a GitHub repository (bit.ly/DL4HM) where researchers can co-operate to update the list of relevant mobility datasets and papers.
- An Appendix in which we discuss the characteristics of DL modules, the data sources, the public datasets, the evaluation metrics used in the selected papers, and the mobility patterns used to assess the realism of generative models.

The survey is structured as follows. In Section 2, we provide the definition of spatio-temporal trajectories and spatial aggregations (Section 2.1), and we briefly discuss the key DL modules used by the models tackling human mobility challenges (Section 2.2). A detailed discussion of such modules and a more detailed overview of human-mobility well-known laws and patterns are discussed in Appendix E.

In Sections 3 and 4, we explore, respectively, the predictive and generative tasks summarized in Figure 1. For each task, after discussing why it is a relevant problem, we formally define it. Then, after briefly presenting the traditional mechanistic and/or ML-based approaches, we describe in detail the aspects that are not captured or only partially captured by such models. Then, we highlight how DL-models overtake these limitations and which DL modules are commonly used to succeed. Finally, In Section 5, we discuss some of the open challenges and we derive some conclusions.

2 BACKGROUND

Here we introduce the notation used in the remainder of the paper. In Section 2.1, we define key mobility concepts, and in Section 2.2 we briefly introduce the DL concepts and notation used in this survey.

2.1 Spatio-temporal trajectories and spatial aggregations

Mobility data describe the movements of a set of individuals during a period of observation. They are typically collected through electronic devices and stored in the form of spatio-temporal trajectories or mobility flows.

The trajectory of an individual is a sequence of records that allows for reconstructing their movements during the period of observation [230, 231]. Typically, each record contains the individual's identifier, a geographic location expressed as a spatial point, and a timestamp indicating when the individual went through that location.

Definition 2.1. Let u be an individual, a trajectory $T_u = \langle p_1, p_2, ..., p_{n_u} \rangle$ is a time-ordered sequence composed by n_u spatio-temporal points visited by u. A spatio-temporal point is a pair $p = (t, l)$, where t indicates the time when point $l = (x, y)$ is visited by u, and x and y are spatial coordinates in a given reference system, e.g., latitude and longitude. A semantic spatio-temporal point p is a tuple $p = (o, t, l)$, where t indicates the time when point $l = (x, y)$ is visited by u, l is a pair of coordinates (x, y), and o is a parameter that brings some meaning to the point (e.g., home, workplace, or some other categories), if any.

Typically, stay points (or stops) are detected on spatio-temporal trajectories to find locations in which users spend a minimum amount of time [135, 230]. In some tasks, the geographic space is discretized by mapping the coordinates to a spatial tessellation, i.e., a covering of the bi-dimensional space using a countable number of geometric shapes called tiles, with no overlaps and no gaps. For instance, for crowd flow prediction, a spatial tessellation is used to aggregate the flows of people moving among the tiles.

Definition 2.2. Given an area A, a set of geographical polygons called *tessellation*, $\mathcal{G}$, is defined with the following properties: (1) $\mathcal{G}$ contains a finite number of polygons, g_i, called *tiles*, $\mathcal{G} = \{g_i : i = 1, ..., n\}$; (2) the tiles are non-overlapping, $g_i \cap g_j = \emptyset$, $\forall i \neq j$; (3) the union of all tiles completely covers A, $\bigcup_{i=1}^{n} g_i = A$.

The tiling of the geographic space aims at creating the covering of the entire area of interest using regular tiles, such as equilateral triangular, squared, quadrilateral, hexagonal tiles, or irregular tiles that define the shape of buildings, census cells, or administrative units. A spatial join can then be used to associate each trajectory's point or stay point with the tile that contains it. Since the tessellation has no overlapping tiles and no gaps, each point is assigned only to one tile. Further details may be found in Appendix A.

2.2 Deep Learning Modules

Here we introduce the DL notation used in the remainder of the paper. For a detailed description of the DL modules introduced here, see Appendix B.

FCs: Fully Connected networks (FCs) consist of a series of fully-connected layers, in which all the neurons are connected to those in the next layer. FCs are universal approximators (i.e., can learn any representation function) [66]. In human mobility tasks, FC networks are commonly used to capture the impact on individual or collective mobility of external features and/or preferences (e.g., weather conditions, presence of public events).

RNNs, LSTMs and GRUs: Recurrent Neural Networks (RNNs) [157] can efficiently deal with sequential data, and they are used to capture spatial and temporal patterns in mobility tasks. For example, RNNs are used in next-location prediction to find periodic trajectories patterns, represented as stay point sequences. In crowd flow prediction, RNNs are used to capture the flows' temporal patterns. However, RNNs suffer from the vanishing gradient problem [98] and cannot propagate information found at early steps, losing relevant information at the beginning of a sequence when it is time to analyze its end [98]. Long-short-term Memory networks (LSTMs) [75] and Gated Recurrent Units (GRUs) [35] are two gate implementations that mitigate this problem. Appendix B.1 provides further details and references on RNNs, LSTMs and GRUs.

Attention mechanisms: These mechanisms are based on the idea that, when dealing with a large amount of information, our brains focus on the most significant parts and consider all the others as background information. In attention mechanisms, the information in input is scored according to the context (e.g., attention maps), and the model focuses more on the information with high scores. In human mobility, attention is widely used for next-location prediction and crowd flow prediction to capture user preferences and highlight relevant historical patterns, respectively. Further details can be found in Appendix B.3.

CNNs: Similarly to the visual cortex [78, 79], Convolutional Neural Networks (CNNs) are made of neurons that react only to certain stimuli in a restricted region of the visual field [103]. They are effective in computer vision applications such as object recognition [147, 168], image classification and segmentation [53, 103], movement or event recognition [188], and more [95]. In human mobility tasks, CNNs are widely used to capture spatial patterns in the data, especially in crowd flow prediction where the distribution of people on a geographic region is represented as an image. Additional information on CNNs can be found in Appendix B.2.

Generative Models: in human mobility, Variational AutoEncoders (VAEs) and Generative Adversarial Networks (GANs) are used to generate realistic trajectories (i.e., synthetic trajectories that realistically reproduce mobility patterns). VAEs transform input data (e.g., trajectories) from a high-dimensional space to a low-dimensional space, encoding samples as a distribution [96]. GANs [67] set up a game between a generator (e.g., a neural network) and a discriminator (e.g., a classifier). The generator's goal is to generate realistic data to fool the discriminator, whose purpose is to classify real and fake data and provide feedback to the generator to improve the realism of the generated data. Details about VAEs and GANs can be found in Appendix B.4.

3 PREDICTIVE MODELS

The goal of predictive models for human mobility is to forecast future whereabouts, either at the individual or collective level. At the individual level, next-location predictors forecast an individual's future whereabouts, based on their historical observations (Section 3.1). At the collective level, crowd flow predictors forecast the

amount of people moving from or to geographic locations given historical information about aggregated crowd flows (Section 3.2).

In this Section, we describe both tasks, discussing how DL brings significant improvements with respect to traditional approaches, and describing the relevant state-of-the-art solutions to each task, with a reference to the public datasets and the metrics used for training and testing the models.

3.1 Next-Location Prediction

Predicting individuals' future locations is relevant in multiple applications such as monitoring public health [11, 26], well-being [137, 192], and traffic congestions [164], and to improve travel recommendation, geomarketing, and link prediction in social network platforms [23, 203, 227, 228, 235]. Next-location predictors may help policymakers organize the public transportation network, urban planners decide a city's future developments, and transportation companies provide citizens with a better service in terms of traffic reduction and ease of mobility. Predicting an individual's next location is challenging because it requires capturing the spatial and temporal patterns that characterize human habits [9], and combining heterogeneous data sources to model multiple factors influencing human displacements (e.g., weather, transportation mode, presence of POIs).

Problem Definition. Next-location prediction consists of forecasting the next location (stay point) an individual will visit in the future, given their historical mobility data. Formally, let u be a user, T_u their trajectory, and $p_t \in T_u$ u's current location, next-location prediction aims at predicting u's next destination p_{t+1}. This problem is treated in two ways: *(i)* as a multi-class classification task, in which we have as many classes as locations and we aim at predicting the next visited location p_{t+1}; or *(ii)* as a regression task, predicting $p_{t+1} = (x_{t+1}, y_{t+1})$, where x_{t+1} and y_{t+1} are the next location's geographic coordinates. A variant of next-location prediction aims at forecasting the next Point Of Interest (POI) p_{t+1} an individual u will visit given their trajectory T_u. Regardless of the specific definition, next-location predictors output a ranking of the probability of each location to be u's next destination.

DL vs Traditional approaches. Next-location prediction has been widely explored prior to the DL explosion using probabilistic or pattern-based approaches, which can work with a reasonably small amount of data [23, 228]. In a seminal work, Calabrese et al. [25] propose a probabilistic model combining people's trajectories and geographical features such as land use, POIs, and distance of trips. Ashbrook et al. [6] cluster GPS data into meaningful locations and incorporate them into a Markov model to predict individuals' future movements. Gambs et al. [62] introduce a Mobility Markov Chain (MMC) in which states represent POIs and transitions between states correspond to a movement between two POIs [62, 63]. Among the pattern-based approaches, Monreale et al. [121] develop a trajectory pattern mining algorithm to represent movement patterns as sequences of regions frequently visited with a typical travel time. Although traditional approaches achieve good performance with a small amount of data, they have substantial limitations. Notably, they require a considerable effort in feature engineering and cannot capture long-range temporal and spatial dependencies [159].

Next-location predictors should capture the spatial, temporal and social-geographic dimensions of human mobility (Figure 2). Regarding the spatial and temporal dimensions, predictors must capture at the same time spatial and temporal regularities hidden in human movements, as well as the tendency of people to get out of the routine. At the same time, predictors should capture the impact of external factors and individual preferences on the decision to move (e.g., weather conditions, preference for certain POIs, influence of friendships). Traditional approaches can only partially capture these aspects, and they particularly struggle in capturing complex sequential patterns in the data. DL approaches overtake these issues by using mechanisms such as RNNs, LSTMs, GRUs, FCs, attention mechanisms, and CNNs to capture temporal, spatial, and social-geographic patterns in the data.

Datasets and Evaluation Metrics. Next-location predictors are mainly trained and tested on public datasets of check-ins, generally coming from geosocial network platforms (e.g., Gowalla, Foursquare), and GPS traces,

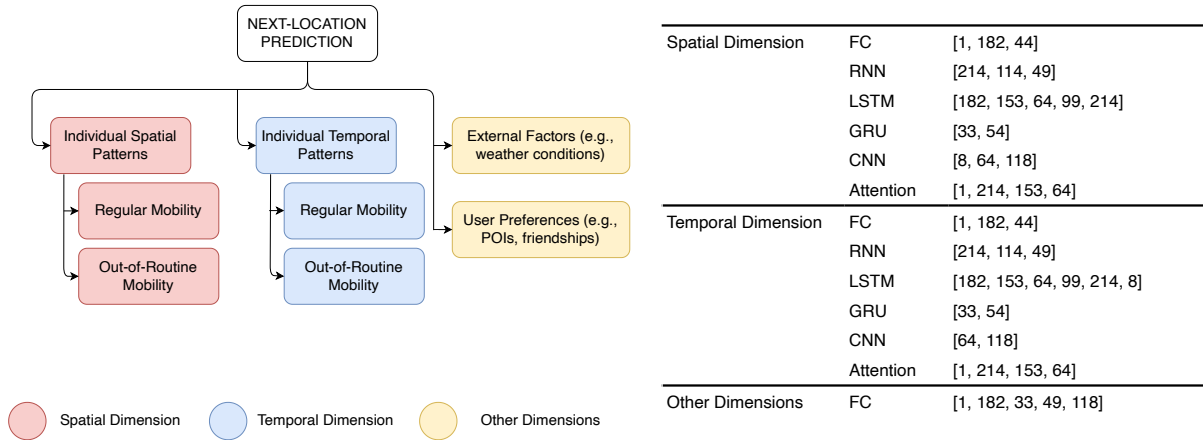| Spatial Dimension	FC	[1, 182, 44]
	RNN	[214, 114, 49]
	LSTM	[182, 153, 64, 99, 214]
	GRU	[33, 54]
	CNN	[8, 64, 118]
	Attention	[1, 214, 153, 64]
Temporal Dimension	FC	[1, 182, 44]
	RNN	[214, 114, 49]
	LSTM	[182, 153, 64, 99, 214, 8]
	GRU	[33, 54]
	CNN	[64, 118]
	Attention	[1, 214, 153, 64]
Other Dimensions	FC	[1, 182, 33, 49, 118]

Fig. 2. **a)** The aspects next-location predictors should capture regarding the spatial dimension (red), the temporal dimension (blue), and the social and geographic dimensions (yellow) of human mobility data. **b)** DL modules that allow to capture each dimension, with the reference to the selected papers in the literature that implement these modules.

collected with smartphones or vehicles on-board GPS devices. Examples of check-ins' datasets widely used for next-location prediction are Cho et al. [34], Feng et al. [54], and Yang et al. [213].

Examples of commonly used trajectory datasets are taxi traces collected in Porto, Portugal [123] and San Francisco, USA [143]. We refer to Appendix C.2 and C.3 for details on the check-in and trajectory datasets mentioned above.

Whenever next-location prediction is intended as a regression task, predictors are evaluated using the Haversine distance or the equirectangular distance between the actual location and the predicted one (see Appendix D.1 for details). When next-location prediction is intended as a multi-class classification task, predictors are evaluated with accuracy (ACC/ACC@k), recall (Rec@k), F1-score (F1@k), Mean Average Percentage Error (MAPE) and/or Area Under the Curve (see Appendix D.2 for details).

DL approaches. Table 1 contains the selected DL approaches to next-location prediction, with the corresponding DL modules, datasets and evaluation metrics used, and link to an implementation (if available).

De Brébisson et al. [44] use an FC to predict taxi's passenger drop-off locations. The input data consist of trajectories represented as a variable-length sequence of GPS points and other meta-information, such as departure time, driver identity, and client information. The model performance is evaluated on the dataset of taxis in Porto [123] in terms of equirectangular distance to the actual visited location. However, it cannot take into consideration the temporal dimension of the mobility data.

ST-RNN (Spatial Temporal Recurrent Neural Networks) [114] overtakes this issue by extending RNN with time- and spatial-specific transition matrices. Each RNN's layer learns an upper and lower bound for the temporal and spatial matrices through linear interpolation. The model is evaluated on the datasets of Gowalla [34] and the Global Terrorism Dataset (GTD) [177] using F1-score, Rec@k, MAPE and AUC.

DeepMove [54] is an attentional recurrent network for mobility prediction from lengthy and sparse trajectories. First, historical and current trajectories are passed to a multi-modal embedding module to create a dense representation of the spatio-temporal and individual-specific features. Historical trajectories are handled by an attention mechanism to extract mobility patterns, while a GRU handles current trajectories. The output of the multi-modal embedding, the GRU, and the attention mechanism are concatenated and passed to an FC to predict

	Reference	Name	Year	DL Modules	Evaluation	Dataset	Code (https://bit.ly)
Next-Location Prediction	*Abideen et al.*[1]	DWSTTN	2021	Encoder, Decoder, Attention, FC	Distance	[123]	-
	Tang et al.[182]	CLNN	2021	LSTM, Embedding, FC	Distance	[123]	-
	Bao et al.[8]	BiLSTM-CNN	2020	Embedding, BiLSTM, CNN	ACC@k	-	
	Chen et al.[33]	DeepJMT	2020	GRU, FC, Encoder	ACC@k	[213]	-
	Yang et al.[212]	Flashback	2020	Attention, RNN	ACC@k	[34]	Flashback-1
	Ebel et al.[49]	-	2020	RNN, FC, Embedding	Distance	[123, 143]	-
	Rossi et al.[153]	-	2019	Attention, LSTM	Distance	[123, 143, 184]	-
	Gao et al.[64]	VANext	2019	CNN, GRU, Attention	ACC@k	[34]	-
	Kong et al.[99]	HST-LSTM	2018	LSTM	ACC	-	HST-LSTM
	Lv et al.[118]	T-CONV	2018	CNN, FC	Distance	[123]	T-CONV
	Feng et al.[54]	DeepMove	2018	Attention, GRU, FC	ACC	[54]	DeepMove
	Yao et al.[214]	SERM	2017	LSTM	ACC@k	-	SERM-Repo
	Liu et al.[114]	ST-RNN	2016	RNN	Rec@k, F1@k, MAPE, AUC	[34, 177]	STRNN
	De Brébisson et al.[44]	-	2015	FC	Distance	[123]	next-loc-1

Table 1. List of the selected papers tackling next-location prediction. For each paper, we describe the name of the corresponding proposed model (if any), the year of publication, the DL modules used in the proposed solution, the metrics used for performance evaluation, the link to the public datasets used for training and testing the model, and the link to a repository with the code (if any). The papers are sorted by year of publication in decreasing order.

an individual's next-location. DeepMove is evaluated, using ACC@k, on Foursquare data [54], private mobile phone data, and a private dataset from a popular Chinese social network platform.

HST-LSTM (Hierarchical Spatial-Temporal LSTM) [99] aims at predicting an individual's short-term next-location. First, the authors design an ST-LSTM (Spatial Temporal LSTM), which combines the spatial and temporal characteristics of a trajectory using an LSTM. Then, ST-LSTM is extended into HST-LSTM, which models periodic patterns using an encoder-decoder module. The encoder encodes the locations visited by a user in a given time span and area of interest, while the decoder predicts the possible areas of interest the user will visit next. HST-LSTM is evaluated using ACC on private data from Baidu.

T-Conv [118] treats trajectories as images and handle them using CNNs to capture the spatial patterns at different scales. The output of the CNN, the trajectory's starting date-time, and other personal information about the user are passed to an FC that handles the prediction. T-Conv is evaluated using the datasets of taxis in Porto [123] and the Haversine distance.

Rossi et al. [153] propose an LSTM network equipped with a self-attention module to predict the coordinates of a taxi's next drop-off location. Locations are enriched with geographical data to describe the surrounding area of the location semantically. The model is tested using the Haversine distance and on the datasets of taxis in Porto [123], New York City [184], and San Francisco [143].

Another strand of research focuses on predicting the next POI an individual will visit using semantic trajectories. For example, SERM (Semantics-Enriched Recurrent Model) [214] relies on an embedding layer to represent the timestamp, the location, and the keywords of a social media post concisely. Both the user's trajectory and the embedding are fed into an LSTM responsible for predicting the next POI. SERM is evaluated using ACC@k on Foursquare check-ins in New York City [222] and tweets in Los Angeles [223].

In VANext (Variational Attention based Next Location) [64], the historical trajectories and the current one are embedded using two separate causal encoders to represent the semantic relationships among POIs. The encoded historical trajectories are passed to a CNN; the encoded current trajectory is passed to a GRU. The outputs of the CNN and the GRU are passed to an attention mechanism, which detects the most similar historical trajectory to the current one and passes it to an FC that predicts the individual's next POI. VANext is evaluated using ACC@k on the datasets of Gowalla [34] and Foursquare for Singapore and New York City.

Flashback [212] is based on an RNN and the concept of flashback, a technique that uses a sparse semantic trajectory to predict the next POI by looking for similar trajectories in terms of temporal characteristics. Flashback

also uses an embedding to model the preferences of individuals to visit specific POIs. The outputs of the RNN and the embedding are passed to an FC that predicts the next visited POI. Flashback is evaluated using ACC@k on check-ins from Gowalla [34] and Foursquare.

DeepJMT (Deep Model for Joint Mobility and Time) [33] can predict an individual's next POI as well as when they will visit it. The model is based on four pipelines: a sequential dependency encoder, a spatial context encoder, a periodicity context extractor, and a social-temporal context extractor. The sequential dependency encoder is a hierarchical GRU that takes as input an embedding of a user's trajectories. The high-level GRU captures the transitions between trajectories; while the low-level GRU models the transition within a trajectory. The spatial context extractor determines the dynamic influence of spatial neighbors, modeled as a graph in which nearest points influence the final prediction. The periodicity context extractor is an attentional GRU aiming at extracting periodicity patterns from an individual's historical trajectories. The social-temporal context extractor leverages social relationships using an FC and pooling functions to facilitate both next-POI and time prediction. Finally, the outputs of the four modules are concatenated to generate the prediction. DeepJMT is evaluated using ACC@k on Foursquare check-ins in New York City [213], Tokyo [213] and Istanbul.

Ebel et al. [49] propose a model to predict a vehicle's destinations and routes, given a partial trajectory and contextual data (e.g., day, time, weather). First, the area is tessellated and GPS points are assigned to these tiles using a k-d tree-based space partitioning method. The model is based on two main modules. The first module is an RNN that takes as input the mapped trajectory; while the second module is an FC that takes as input the embedded contextual data. The two modules' outputs are merged and passed to an additional FC that produces the individual's probabilities to end a trip into a specific tile. The model is evaluated with the mean Haversine distance and the distance to the actual arrival point on taxis' traces in Porto [123] and San Francisco [143].

BiLSTM-CNN [8] relies on a spatial clustering algorithm to derive areas of interest from POIs and uses bi-LSTMs and CNNs to predict the next area an individual will visit. The historical mobility data of people are passed into a bi-directional LSTM and then to a CNN to capture the overall spatial and temporal patterns. The CNN's output is then passed to an FC handling the prediction of the next location. The model is evaluated with ACC@1 and ACC@5 on a private dataset of Weibo check-ins collected in Wuhan, China.

CLNN (Classification Learning Neural Network) [182] relies on an information extraction module that extracts coordinates, date, time and driver characteristics, POIs and historical information from mobility data (e.g., similar trajectories). An LSTM is used to process the coordinates. Date, time, driver characteristics and POIs are embedded into a dense representation. Two FCs process the embedded and historical information, respectively. The outputs of the two FCs and the LSTM are fused together with a weighted sum, whose output is fed into an FC to predict the coordinates of the next taxi's destination. The model is evaluated using the mean Haversine distance on the dataset of taxis in Porto [123].

DWSTTN (Deep Wide Spatio Temporal Transformer Network) [1] is a transformer-like architecture that predicts a taxi's next destination using historical pick-up and drop-off observations and taxis' features and preferences. The model consists of two identical parts, the encoder and the decoder. The encoder is responsible for the learning phase, the decoder for predicting the coordinates of the taxi's next location. In both, information about a taxi (e.g., identifier and stand identifier) and temporal information (e.g., weekday, hour, day type) are passed to an embedding layer (temporal transformer), while information about locations is embedded by a spatial transformer. The outputs of the temporal and spatial transformers are passed to two attention mechanisms to extract further relevant spatial and temporal information. The outputs of the attention mechanisms are fused together using a weighted sum, whose output is passed to an FC. In the encoder, the FC outputs the learned features' representation; in the decoder, the FC outputs the next location's coordinates. The model is evaluated using the Haversine distance on the datasets of taxis in Porto [123] and New York [184].

3.2 Crowd Flow Prediction

Crowd flow prediction is the problem of forecasting the incoming and outgoing flows of locations in a geographic region, usually split into tiles on a spatial tessellation [204, 231]. It is a crucial problem given its impact on several aspects of society, from public safety [231] to the definition of on-demand services [234], the management of land use [81], and traffic optimization [231]. For example, crowd flow predictors may help city managers and policymakers discover the traffic congestions in the city; people in business find potential areas of business investment; citizens improve travel plans and stagger the peaks of travel. These predictors may also help prevent or mitigate dangerous situations, such as creating massive crowds of people streamed into a strip region, by sending out warnings or evacuating people in advance.

Crowd flow prediction is challenging because it requires dealing with both spatial and temporal dependencies. Indeed, a region's out-flow may affect the in-flows of both near and far regions. At the same time, crowd flows are characterized by temporal closeness, trends, and periodicity. Temporal closeness marks the dependencies between events that are close in time; trends highlight patterns that repeat over time (e.g., weekends and working days); periodicity captures the repetitive nature of relevant events (e.g., rush hours in the morning). Furthermore, exogenous factors such as weather conditions, holidays, and the presence of public city events may affect crowd flow patterns.

Problem Definition. Given an individual's trajectory T_u and a spatial tessellation $\mathcal{G}$ of the geographic space (generally a $i \times j$ grid), the set of locations (tiles) the trajectory intersects in a time interval Δt is:

$$q_{T_u}^t = \{(p_k \rightarrow t) \in \Delta t \land (p_k \rightarrow (x, y)) \in (i, j) | (i, j)\}, \tag{1}$$

where (i, j) indicates a location on $\mathcal{G}$ and p_k is the user u's current location, identified by the coordinates (x, y). Let Q be the set of locations covered by all the individual trajectories and let $t - 1$, t, and $t + 1$ be three consecutive time spans, the incoming flow $\text{in}_t^{(i,j)}$ to a location (i, j) is the number of the individuals that are in (i, j) at time t but were not in (i, j) at time $t - 1$. Similarly, the outgoing flow $\text{out}_t^{(i,j)}$ to location (i, j) is the number of individuals that are in (i, j) at time t and move to another location at time $t + 1$:

$$\text{in}_t^{(i,j)} = \sum_{T \in Q} |\{t > 1 | (i, j) \notin q_T^{t-1} \land (i, j) \in q_T^t\}|; \quad \text{out}_t^{(i,j)} = \sum_{T \in Q} |\{t > 1 | (i, j) \in q_T^t \land (i, j) \notin q_T^{t+1}\}|. \tag{2}$$

We can represent the flows of a region as a tensor $X_t \in R^{2 \times I \times J}$, where one dimension is associated with the in-flow $(X_t)_{1,i,j} = \text{in}_t^{(i,j)}$ and the other with the out-flow $(X_t)_{2,i,j} = \text{out}_t^{(i,j)}$. Therefore, crowd flow prediction is the task of predicting $X_{t+\Delta}$ given the historical flows $\{X_t | 1, \ldots, X_t | t\}$. In most of the selected papers, $\Delta = 1$. When $\Delta > 1$ (e.g., in [196, 233]), the problem is named multi-step crowd flow prediction.

A variant of crowd flow prediction aims at forecasting the entire origin-destination matrix (i.e., flows among pairs of locations) given the historical observations of crowd flows [152].

DL vs Traditional approaches. Crowd flow prediction may be tackled using classic time-series prediction models based on autoregression (AR), such as the AutoRegressive Moving Average (ARMA) [20], the Autoregressive Integrated Moving Average (ARIMA) [107, 122], and variants like the Stationary and Seasonal ARIMA (SARIMA) [201], vector ARMA [89] and space-time ARIMA [90]. Since ARMA and ARIMA can make predictions only out of stationary time-series that do not statistically change over time, they cannot accurately predict new events. SARIMA, vector ARMA and space-time ARMA are designed to overtake this assumption, but they present other issues. For instance, SARIMA's predictive performance decreases when dealing with patterns with long seasonal periods. In general, the AR predictors have difficulties handling spatial dependencies and short-term samples, capturing patterns way before in the time series, and including additional features (e.g., weather conditions, presence of public city events), making them ineffective for crowd flow prediction.

In contrast, DL approaches can effectively model spatial and temporal dependencies and capture the influence of external factors (see Figure 3). Since spatial tessellations describe a bidimensional space, DL predictors represent

a)   b)

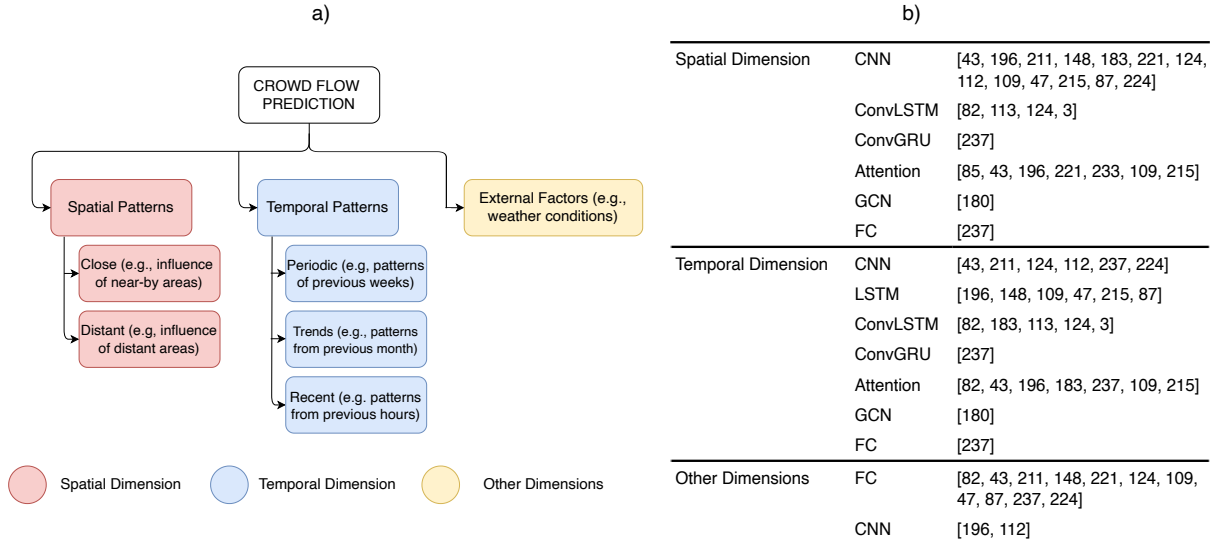| Spatial Dimension	CNN	[43, 196, 211, 148, 183, 221, 124, 112, 109, 47, 215, 87, 224]
	ConvLSTM	[82, 113, 124, 3]
	ConvGRU	[237]
	Attention	[85, 43, 196, 221, 233, 109, 215]
	GCN	[180]
	FC	[237]
Temporal Dimension	CNN	[43, 211, 124, 112, 237, 224]
	LSTM	[196, 148, 109, 47, 215, 87]
	ConvLSTM	[82, 183, 113, 124, 3]
	ConvGRU	[237]
	Attention	[82, 43, 196, 183, 237, 109, 215]
	GCN	[180]
	FC	[237]
Other Dimensions	FC	[82, 43, 211, 148, 221, 124, 109, 47, 87, 237, 224]
	CNN	[196, 112]

Fig. 3. **a)** The aspects crowd flow predictors should capture regarding the spatial dimension (red), the temporal dimension (blue), and the social and geographic dimensions (yellow) of human mobility data. **b)** DL modules that allow capturing each dimension, with reference to the selected papers in the literature that implement these modules.

crowd flows as matrices and exploit the effectiveness of CNNs in detecting near and far away spatial and temporal dependencies in matrices. At the same time, since gated RNNs such as LSTMs and GRUs may be used to model complex sequential patterns in the data, DL approaches may efficiently capture patterns in the temporal evolution of crowd flows. CNNs and RNNs may be combined in ConvLSTM models, which may capture spatial and temporal patterns at the same time. Finally, we should also consider other aspects that may affect crowd flows, such as weather conditions and the presence of public events in the city. Such external factors may be handled with FCs and combined with the output of the CNN/RNN modules.

Datasets and Evaluation metrics. The most used datasets to train and evaluate crowd flow predictors are the Citi Bike dataset [14], which describes the trips between bike-sharing stations in New York City, and Zhang et al.'s dataset [224], which provides the aggregated incoming and outgoing flows for each tile of a squared tessellation of New York City and Beijing, extracted from raw taxis' GPS traces. Some papers use a dataset about the bike-sharing system of Washington D.C. [15], and a dataset describing the pick-up and drop-off locations of taxis in New York City [184]. We refer to Appendix C.2 and C.3 for details about these datasets.

The performance of crowd flow predictors is evaluated as the error between the empirical crowd flows and the predicted ones. Commonly used error metrics are Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Mean Absolute Percent Error (MAPE) (see Appendix D.3 for definitions and details).

DL approaches.

Table 2 shows the selected papers on crowd flow prediction, highlighting the DL modules, evaluation metrics, and datasets they use, with a link to an implementation (if available).

In their seminal work, Zhang et al. propose ST-ResNet [224], which consists of three modules that rely on CNNs to capture trends, periodic patterns, and temporal closeness. The modules' output is combined with the output of an FC that deals with external factors such as weather conditions and presence of public events. The model is evaluated on the datasets of taxis in Beijing and bikes in New York City [224] using RMSE.

	Reference	Name	Year	DL Modules	Evaluation	Dataset	Code (https://bit.ly)
	Jiang et al.[82]	DeepCrowd	2021	ConvLSTM, Attention, FC	MSE, MAE, MAPE, RMSE	[82]	DeepCrowd
	Dai et al.[43]	-	2021	Attention, CNN, FC	RMSE	[224]	-
	Wang et al.[196]	SeqST	2020	LSTM, CNN, Attention	MAE, RMSE	[14, 184]	-
	Yang et al.[211]	ST-ESNet	2020	CNN, FC	RMSE	[224]	-
	Ren et al.[148]	HIDLST	2020	LSTM, CNN, FC	RMSE	[224]	-
	Tian et al.[183]	LDRSN	2020	CNN, ConvLSTM, Attention	RMSE, MAPE, MAE	[14, 184]	-
	Yuan et al.[221]	MV-RANet	2020	CNN, Attention, FC	RMSE, MAPE	[224]	-
	Liu et al.[113]	ATFM	2020	ConvLSTM	RMSE	[224]	ATFM-2
	Sun et al.[180]	MVGCN	2020	GCN, FC	RMSE, MAE	[14, 15, 184, 224]	-
	Mourad et al.[124]	ASTIR	2019	ConvLSTM, CNN, FC	RMSE	[224]	ASTIR_Model
	Zhou et al.[233]	ST-Attn	2019	Attention, FC	RMSE	[14, 224]	ST-Attn
	Rong et al. [152]		2019	CNN, ResNet	RMSE, MAE	-	
	Li et al.[109]	ST-DCCNAL	2019	CNN, Attention, LSTM, FC	RMSE	[224]	ST-DCCNAL
	Lin et al.[112]	DeepSTN+	2019	CNN	RMSE, MAE	[14]	DeepSTN
	Du et al.[47]	DST-ICRL	2019	CNN, LSTM, FC	RMSE, MAE	[224]	DST-ICRL
	Ai et al.[3]	-	2018	ConvLSTM	RMSE, MAE	-	-
	Yao et al.[215]	STDN	2018	CNN, LSTM, Attention, FC	RMSE, MAPE	[14]	STDN-2
	Jin et al. [87]	STRCN	2018	CNN, LSTM, FC	RMSE	[224]	-
	Zonoozi et al.[237]	PCRN	2018	ConvGRU, CNN, FC	RMSE	[224]	-
	Zhang et al.[224]	ST-ResNet	2017	CNN, FC	RMSE	[224]	ST-ResNet

(Crowd Flow Prediction)

Table 2. List of selected papers tackling crowd flow prediction. For each paper, we describe the name of the corresponding proposed model (if any), the year of publication, the DL modules used in the proposed solution, the metrics used for performance evaluation, the link to the public datasets used for training and testing the model, and the link to a repository with the code (if any). The papers are sorted by year of publication in decreasing order.

ST-ESNet [211] extends STRes-Net [224] by adding convolutional layers to upsample and downsample the matrices representing crowd flows. The model is evaluated with the datasets of taxis in Beijing [224] using RMSE.

Many works in the literature combine CNNs with RNNs to exploit the latter's capability to deal with temporal patterns. STRCN (Spatio-Temporal Recurrent Convolutional Network) [87] uses three CNNs to capture close, short-term (daily influence), and mid-term (the difference between workdays and weekends) spatial patterns. The output of the CNNs is fed into three LSTMs, which handle temporal dynamics. STRCN also uses external features, such as weather conditions and features to distinguish between workdays and holidays, through an FC. The output of the FC and the LSTMs are combined. Model's performances are evaluated on the datasets of taxis in Beijing and bikes in New York City [224] using RMSE.

STDN (Spatial-Temporal Dynamic Network) [215] consists of two CNNs: the first one captures the local spatial dependencies based on the similarities of historical traffic volumes to uncover the flows; the second CNN captures the traffic flows. The outputs of the two CNNs are pairwise multiplied and summed with an FC's output that handles external features (e.g., weather events). Moreover, STDN uses three LSTMs to capture the temporal dependencies of historical data describing the crowd flows. CNNs and LSTMs work on two separate pipelines, the outputs of which are summed together and forwarded to another LSTM with an attention mechanism that analyzes the temporal dynamics of the current day. Finally, the output of this latter LSTM is passed to an additional FC to perform crowd flow and traffic prediction. The model performance is evaluated on the New York City's bikes dataset [14] and on a dataset of taxis' flows in New York City [184] using MAPE.

DST-ICRL (Deep Spatio-Temporal with Irregular Convolutional Residual Network) [47] combines convolutional residual units with LSTMs to capture the irregular properties of traffic flows in different transportation lines. Similarly to STDN [215] and ST-RESNET [224], DST-ICRL uses three pipelines to uncover daily, weekly, and recent spatio-temporal dynamics and handles external features using an FC. The authors evaluate DST-ICRL on two private datasets of checkins from e-cards on buses and subways in Beijing, using MAE and RMSE.

Another category of crowd flow predictors exploits attention mechanisms. ST-DCCNAL (Spatio-Temporal Densely Connected Convolutional Networks and Attention LSTM) [109] combines a CNN with an attentional

LSTM to simplify the selection of the inputs. The authors use DenseNet [77] to cope with the spatial patterns and FCs to deal with the external features. The outputs of DenseNet and the FCs are fed into an attentional LSTM to extract temporal patterns and make the prediction. The performances are evaluated on the datasets of taxis in Beijing and bikes in New York City [224] using RMSE.

MV-RANet (Multi-View Residual Attention Network) [221] uses two pipelines to deal with spatio-temporal patterns and mobility patterns. The first pipeline models the closeness, the periods, and the trends of crowd flows using three attention residual networks, the outputs of which are fused and passed to a convolutional layer. The output of the convolutional layer is then passed to an FC. The second pipeline captures the mobility patterns in the data, generating three graphs: one for the transition probability, one for the transition distance, and one that mimics the flow patterns. The three graphs are then encoded using Node2Vec [71], an algorithmic framework that learns a continuous feature representation of the nodes of a graph, and passed to an FC to extract relevant mobility patterns. The outputs of the FC handling the graph representation and the FC that handles the spatio-temporal patterns are fed into an additional FC responsible for making the prediction. MV-RANet is evaluated using RMSE, MAPE, and MAE on the flows of taxis in Beijing [224] and a private mobile phone data (CDRs) for the city of Sanya.

LDRSN (Local-Dilated Region-Shifting Network) [183] consists of five modules. The first one is a CNN that handles local spatial dependencies; the second one uses dilated units to deal with distant spatial dependencies. A k-dilated unit is a convolution that captures the correlation of two regions with a distance of k. The outputs of these two modules are fused and forwarded to two other modules: one handles long-term temporal dependencies, the other deals with short-term temporal patterns. The long-term module consists of three pipelines with consecutive convolutional layers, each followed by attention LSTMs. A single ConvLSTM handles the short-term patterns. The outputs of the two temporal modules are fed into a final module: as a first step, it sums the outputs of the two temporal modules and forwards the result to a convolutional layer to make the prediction. The model is evaluated on the dataset of bikes' [224] and taxis' in New York City [184] using MAE, MAPE, and RMSE.

To capture long-range spatial dependencies, DeepSTN+ (Deep Spatio Temporal Network Plus) [112] replaces the traditional residual units and the convolutional layers with ResPlus and ConvPlus. ResPlus units employ a ConvPlus layer and a standard convolutional layer to capture distant spatial dependencies. The idea beyond a ConvPlus layer is to separate the channels of the input matrix and use an FC to capture the long-range spatial dependencies among each pair of regions. Finally, DeepSTN+ uses an average pooling layer before the FC to reduce the number of parameters. Another peculiarity of DeepSTN+ is that it uses temporal factors and the distribution of POIs to gain prior knowledge of the crowd flows. The experiments are conducted on the dataset of bikes in New York City [14] and a private dataset from the most popular social network vendor in China using MAE and RMSE.

ST-Attn [233] is specifically designed for multi-step crowd flow prediction. It relies on an encoder and a decoder based on two attention mechanisms (one capturing spatial patterns and one capturing temporal patterns) and several FCs. The encoder is a stack of a spatial and temporal attention mechanism and FC layers. The decoder has two consecutive attention mechanisms to capture spatial and temporal patterns: the first one uses historical observations about crowd flows, the second one uses the output of the first attention mechanism and the output of the encoder. The decoder's output is passed to an FC layer to predict the Δ next crowd flows. The model is evaluated for $\Delta \in 1 \ldots 6$ on the datasets of taxis in Beijing [224], taxis in New York City [184], and bikes in New York City [14] using RMSE.

ASTIR [124] consists of three pipelines each using ConvLSTMs to capture periodic, long, and short-terms spatial and temporal dependencies, respectively. The outputs of the three ConvLSTMs are passed to three attention mechanisms, whose outputs are passed to three ConvLSTMs. The outputs of the three pipelines are combined with the results of an FC handling external factors and passed to a tanh activation function to perform the prediction. ASTIR is evaluated on the dataset of taxis in Beijing and bikes in New York City [224] using RMSE.

SeqST [196] performs multi-step crowd flow prediction using a sequence to sequence GAN. The GAN's generator and discriminator rely on CNNs and LSTMs to extract spatio-temporal patterns from the data. In the generator, they use further CNNs to consider the impact of external factors, while the discriminator drives the generator to predict realistic crowd flows. The model is evaluated on the dataset of taxis [184] and bikes in New York City [14] using MAE and RMSE.

DeepCrowd [82]uses three pipelines to capture patterns in previous hours, days, and weeks. It consists of three ConvLSTMs that downsample and upsample the input data. Each convLSTM is followed by an attention mechanism to extract relevant local spatio-temporal patterns. The outputs of the three attention mechanisms are concatenated and passed to an additional attention mechanism to further extract relevant patterns. The output of the attention mechanism is passed to a CNN to predict crowd flows. DeepCrowd is evaluated on a public dataset of crowd flows in Tokyo and Osaka, Japan [82] using MSE, MAE, MAPE, and RMSE.

In contrast with the majority of solutions to crowd flow prediction, Dai et al. [43] uses only a single pipeline to capture both near and far away temporal and spatial patterns. All the matrices representing crowd flows are passed to a temporal attention mechanism, whose output is passed to a spatial attention mechanism. The output of the spatial attention mechanism is fed into a CNN to extract spatio-temporal patterns. The output of this CNN is fused with the results of a FC which handles external features. The model is evaluated on the dataset of taxis in Beijing and bikes in New York City [224] using RMSE.

Some other models address slightly different definition of crowd flow prediction [3, 180, 237]. PCRN (Periodic Convolutional Recurrent Networks) [237] solves the problem of predicting the distribution of presences in a city. The authors use a pyramidal model made of three ConvGRUs (which has both the advantages of CNNs and GRUs) with an external module that captures the patterns' periodicity by memorizing the periodic representations learned by the stacked ConvGRUs. The authors propose three ways to retrieve and update the periodic patterns: using a sequential periodic representation, using an estimated average of periodic representation, adopting a temporally ordered representation. The outputs of the periodic module and the ConvGRUs are fused. The model is evaluated on the datasets of taxis in Beijing and bikes in New York City [224] using RMSE.

Ai et al. [3] aim to predict the short-term distribution of features that describe the movements of bikes of a dockless bike-sharing system. They rely on ConvLSTMs to address the spatial and temporal dependencies. ConvLSTMs take as input a spatio-temporal sequence composed of the number of bicycles in an area, the distribution uniformity, the usage distribution, and the time of the day to predict their values in the near future. The authors evaluate their model using private datasets from two bike-sharing companies in Chengdu, China. The metrics used for the evaluation are MAE and RMSE.

Some recent works solves the problem of flow prediction, a variant of crowd flow prediction in which they aim to use historical observations of crowd flows to forecast the entire origin-destination matrix, i.e., flows among pairs of locations.

MVGCN (Multi-View Graph Convolutional Network) [180] does so handling the external features (e.g., weather data) with two FCs, which deals with weather information and meta-information such as time and day of the week. The authors select key time snapshots to process graphs representing recent, daily, weekly, monthly and quarterly mobility flows. Each node in these graphs represents a region with time-varying flows. The graphs are forwarded to five graph convolutional networks and the outputs of the seven networks (five graph convolutional networks and two FCs for the external features) are fused using a multi-view fusion mechanism. The fusion module's output is fed into an additional FC that outputs the predicted graph, in which each node is a region and the links are the predicted flows. The experiments are conducted on the dataset of bikes [14] and taxis [184] in New York City, bikes in Washington D.C. [15], and taxis in Beijing [224]. In both cases, RMSE and MAE are used to evaluate the model's performance.

Rong et al. [152] use a distinct CNN for each location represented as an image-like matrix, to extract long and short-terms spatial and temporal features. Each matrix is made of several channels, each corresponding to

one time interval of the historical observations. The model is evaluated on a private dataset of a Chinese social network referring to flows in Beijing using MAE and RMSE.

4 GENERATIVE MODELS

Generative models of human mobility aim at generating realistic spatio-temporal trajectories (trajectory generation, Section 4.1) or mobility flows (flow generation, Section 4.2).

In this Section, we describe both tasks, discussing how DL brings significant improvements with respect to traditional approaches, and describing the relevant state-of-the-art solutions to each task, with a reference to the public datasets and the metrics used for training and testing the models.

4.1 Trajectory Generation

The goal of generative models of individual human mobility is to generate synthetic trajectories with realistic mobility patterns [9, 55, 74, 92, 165, 194]. The generated synthetic trajectories must reproduce a set of spatial and temporal mobility patterns, such as the distribution of characteristic distances traveled and the predictability of human whereabouts (see Appendix E). The use of generative mobility models is crucial in many applications. First, synthetic trajectories are useful to the performance analysis of networking protocols such as mobile ad hoc networks, where the displacements of network users are exploited to route and deliver the messages [74, 92, 186]. Second, synthetic trajectories are fundamental for urban planning, what-if analysis, and computational epidemiology, e.g., simulating changes in urban mobility in the presence of new infrastructures, epidemic diffusion, terrorist attacks, or international events [42, 190, 208, 209]. Furthermore, generative models are a viable solution to protect geo-privacy of trajectory data [59, 120, 140]: while disclosing real data requires a hard-to-control trade-off between uncertainty and utility, synthetic records that preserve statistical properties may achieve in multiple tasks comparable performance to real data.

Solving trajectory generation requires capturing, simultaneously, the temporal and spatial patterns of individual human mobility. A realistic generative model should reproduce the temporal statistics observed empirically, including the number and sequence of visited locations together with the time and duration of the visits. In particular, the biggest hurdle consists of the simultaneous description of an individual's routine and sporadic out-of-routine mobility patterns. Regarding spatial patterns, a generative model should reproduce the tendency of individuals to move preferably within short distances [65, 133], the heterogeneity of characteristic distances [65, 133] and their scales [4], the tendency of individuals to split into returners and explorers [136], the routinary and predictable nature of human displacements [173], and the fact that individuals visit a number of locations that are constant in time [5].

Problem Definition. A generative mobility model M is any algorithm able to generate a set of n synthetic trajectories $\mathcal{T}_M = \{T_{a_1}, \ldots, T_{a_n}\}$, which describe the movements, during a certain period of time, of n independent agents $a_1, \ldots, a_n$. The synthetic trajectory generated for a single agent a_i should be in the form of Definition 2.1, i.e., a time-ordered sequence $T_{a_i} = \langle p_1, p_2, ..., p_k \rangle$ composed by spatio-temporal points, describing the k locations visited by a_i. The realism of M is evaluated with respect to:

(1) A set of spatial patterns $(s_1, \ldots, s_{m_s})$ and temporal patterns $(t_1, \ldots, t_{m_t})$ $\mathcal{K} = \{s_1, \ldots, s_{m_s}, t_1, \ldots, t_{m_t}\}$ (see Appendix E). The patterns refer to the distributions of individual measures, which quantify aspects related to the mobility of a single individual (e.g., radius of gyration, mobility entropy), or collective measures, which quantify aspects related to the mobility of a region as a whole (e.g., OD matrices). A realistic $\mathcal{T}_M$ is expected to reproduce as many mobility patterns as possible.
(2) A set $\mathcal{X} = \{T_{u_1}, \ldots, T_{u_m}\}$ of real mobility trajectories corresponding to m real individuals $u_1 \ldots u_m$ that move on the same region as the one on which synthetic trajectories are generated. Generally, a portion

$\mathcal{X}_{\text{train}} \in \mathcal{X}$ is used to train M or to fit its parameters. The remaining part $\mathcal{X}_{\text{test}}$ is used to compute the set $\mathcal{K}$ of patterns, which are compared with the patterns computed on $\mathcal{T}_M$.

(3) A function D that computes the dissimilarity between two distributions, such as the KL divergence or the JS (see Appendix D.4 for definitions). Specifically, for each measure in $f \in \mathcal{K}$, $D(P_{(f,\mathcal{T}_M)}||P_{(f,\mathcal{X}_{\text{test}})})$ indicates the dissimilarity between $P_{(f,\mathcal{T}_M)}$, the distribution of the measures computed on the synthetic trajectories in $\mathcal{T}_M$, and $P_{(f,\mathcal{X}_{\text{test}})}$, the distribution of the measures computed on the real trajectories in $\mathcal{X}_{\text{test}}$. The lower $D(P_{(f,\mathcal{T}_M)}||P_{(f,\mathcal{X}_{\text{test}})})$, the more realistic model M is with respect to f and $\mathcal{X}_{\text{test}}$.

DL vs Traditional approaches. There is a vast literature on mechanistic generative models that reproduce simple temporal, spatial, and social patterns of human mobility [9, 74, 92, 134, 194]. For example, in the Exploration and Preferential Return (EPR) model [172], an agent can choose between two competing mechanisms: exploration, during which an agent chooses a new location never visited before, based on a random walk process with a power-law jump-size distribution; and preferential return, in which an agent returns to a previously visited location based on its frequency. Several studies subsequently improved the EPR model by adding increasingly sophisticated spatial or social mechanisms [5, 10, 39, 136, 187]. EPR and its extensions focus mainly on the spatial aspects of human mobility, implementing unrealistic temporal mechanisms. TimeGeo [85] and DITRAS [134] improve the temporal mechanism integrating into an EPR-like model a data-driven model that captures the circadian propensity to travel and out-of-routine trips. Although mechanistic models have the advantage of being interpretable by design, their realism is limited because of the simplicity of the implemented mechanisms.
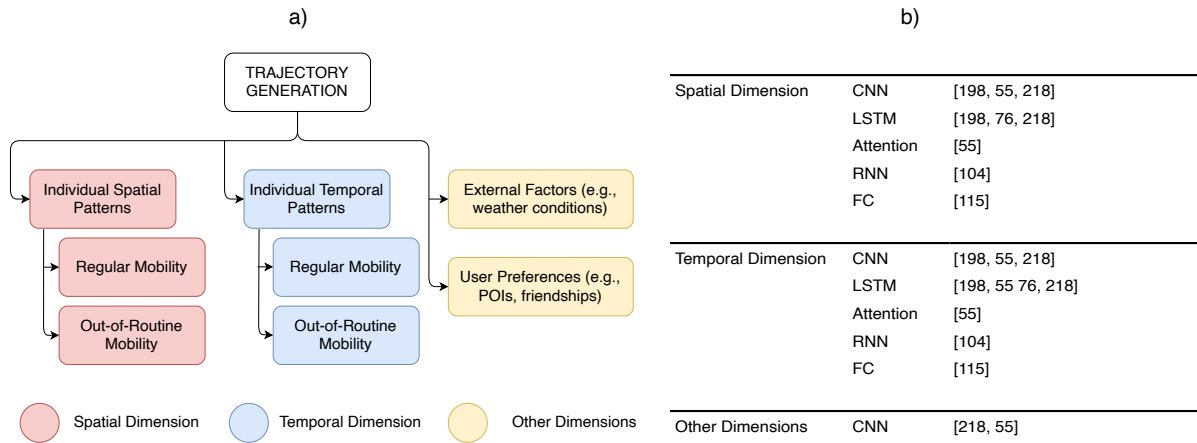

Fig. 4. **a)** The aspects trajectory generators should capture regarding the spatial dimension (red), the temporal dimension (blue), and the social and geographic dimensions (yellow) of human mobility data. **b)** DL modules that allow to capture each dimension, with the reference to the selected papers in the literature that implement these modules.

The limitations mentioned above can be tackled using DL generative paradigms such as GANs and VAEs. In both cases, models rely on DL modules to learn the distribution of data and generate mobility trajectories coming from the same distributions (Figure 4). Thanks to the versatility of DL modules, GANs and VAEs can capture different aspects simultaneously (e.g., spatial, temporal, social dimensions of mobility) while traditional approaches can capture peculiar aspects of mobility only (e.g., only the spatial dimension). Moreover, DL models can capture complex and non-linear relationships in the data that traditional approaches may fail to capture. Therefore, such models can generate more realistic data than traditional models.

Datasets and evaluation metrics. Trajectory generators are commonly trained and evaluated on GPS datasets such as Geolife [232], MDC [106], and taxi traces in San Francisco [143]. Additional information about these

	Reference	Name	Year	DL Modules	Evaluation	Dataset	Code (https://bit.ly)
	Wang et al. [198]	TSG	2021	GAN, CNN, LSTM	distances, r_g, $p(r,d)$	[123]	TSG_Model
	Feng et al. [55]	MoveSim	2020	GAN, self-attention CNN	distances, r_g, $p(r,d)$, DailyLoc, G-rank, I-rank	[232]	
	Huang et al. [76]	SVAE	2019	VAE, LSTM	MDE	-	-
	Ouyang et al. [128]	Ouyang GAN	2018	WGAN, CNN	$p(r)$, $p(r,t)$, $p(r,d)$, $p(r, d_{total})$, $p_{d_{total}}$, location frequency	[106]	-
	Kulkarni et al. [104]	-	2018	RNN, GAN	visitation frequency, statistical similarity, privacy tests	[106]	-
	Yin et al. [218]	-	2018	GAN, FC	reconstruction error, utility loss	[143]	-
	Liu et al. [115]	trajGANs	2018	GANs	-	-	-

(The leftmost column, rotated, reads "Trajectory Generation")

Table 3. List of selected papers tackling trajectory generation. For each paper, we describe the name of the corresponding proposed model (if any), the year of publication, the DL modules used in the proposed solution, the metrics used for performance evaluation, the link to the public datasets used for training and testing the model, and the link to a repository with the code (if any). Papers are sorted by year of publication in decreasing order.

datasets can be found in Appendix C.2. The distance between the distribution measures the performances of trajectory generators (e.g., using KL or JS divergence, see Appendix D.4 for details) of standard mobility metrics computed on a real dataset and the generated dataset. Appendix D.1 describes a set of standard mobility metrics commonly used to evaluate the realism of trajectory generators.

DL approaches. Most of the DL approaches to trajectory generation are based on GANs, while a few are based on VAEs. Table 3 reports a selection of the most relevant DL approaches to trajectory generation.

In their vision paper, Liu et al. [115] propose the trajGANs framework to address the potential and challenges of using GANs for trajectory generation. Similar to a typical GAN [67], a trajGAN consists of a generator G, which accepts a random vector z and generates a dense representation of synthetic trajectories samples, and a discriminator D, which classifies an input trajectory sample into "real" or "fake". Liu et al. [115] suggest the use of Recurrent Neural Networks (RNNs) to create dense representations of trajectories and transform between trajectories and distributional representations.

Ouyang et al. [128] represent a trajectory as a sequence of stays, each with a geographic location, start time and duration. Specifically, a trajectory is an $n_1 \times n_2 \times k$ matrix, where $n_1 \times n_2$ is the size of a squared tessellation, and k is the maximum number of stay repetitions for each location (set in the experiments to $k = 4$). A Wasserstein GAN [72] is used to train a trajectory generator. Both the generator and the discriminator are based on a CNN. The experiments are conducted on the MDC dataset [106] using a 64×64 squared tessellation on the city of Lausanne, Switzerland. The similarity between synthethic and real trajectories is evaluated using the JS divergence on the popularity $p(r)$ and temporal popularity $p(r,t)$ of locations, the staying patterns $p(r,d)$, the semantic importance $p_{d_{total}}(r)$, the semantic distance, and G-rank.

Song et al. [174] use a CNN with four layers within a GAN framework to generate trajectories represented as 512×512 matrices. In a data convolution process, the input 512×512 matrices are resized into a 32×32 matrix. Given the small size of the available (private) mobile phone datasets, the experiments include synthetic ones obtained by randomly shuffling the real trajectories. In a deconvolution process, the GAN's output (a 32×32 matrix) is resized back to a 512×512 matrix using the nearest neighbor function. No quantitative evaluation of the model's realism is provided.

Huang et al. [76] present SVAE (Sequential Variational Autoencoder), a generative model based on a combination of a Variational Autoencoder (VAE) and an LSTM, combining the ability of VAEs to construct a latent space that captures salient features of the training data with the ability of LSTMs to process sequential data. In a data processing phase, they force the input trajectories to have fixed timestamps, and in the experiments they evaluate

the realism of SVAE through the Mean Distance Error (MDE) between each step of real and synthetic trajectory pairs, finding that the reconstruction error of SVAE is smaller than 800 meters.

Kulkarni et al. [104] benchmark the performance of RNNs, SeqGAN [220], RGAN [52] and nonparametric copulas to generate synthetic trajectories. They compare the generated trajectories with real ones extracted from the MDC dataset [106] based on geographic and semantic similarity, statistical similarity, long-range dependencies and privacy tests. They find that copulas have an advantage over all other methods in terms of both model performance and computational time.

MoveSim [55] is a model-free GAN framework that integrates the domain knowledge of human mobility regularity. The generator consists of a self-attention-based sequential model to capture the temporal transitions in human mobility. The discriminator consists of a mobility regularity-aware loss to distinguish the generated trajectory from a fake one. The mobility regularities of spatial continuity and temporal periodicity are used to pre-train the generator and discriminator to accelerate the learning procedure. They conduct experiments on a private mobile phone dataset, using the base station as the spatial unit, and on GeoLife [232], projecting GPS coordinates into a grid. As for the temporal granularity set the basic time slot of a trajectory as half an hour of the day. The realism of MoveSim is evaluated based on the distribution of distances, radius of gyration, number of locations visited daily, G-rank, and I-rank using the JS divergence with respect to real trajectories.

Yin et al. [218] use a GAN-based framework to generate density distributions rather than trajectories, i.e., the number of users in each location at each time slot. Both the generator and the discriminator are implemented with FCs. Experiments are conducted on a dataset extracted from MoMo (spatial resolution 2km, time slots of 30 minutes) and on the dataset of taxis in San Francisco (squared tessellation of 50km×50km, time slots of two minutes) [143]. The proposed model is evaluated in terms of reconstruction error and utility loss outperforms the differential privacy approach in data utility and attack error.

TSG (Two-Stage GAN) [198] consists of two GANs with different objectives. The first GAN captures spatio-temporal patterns from the trajectories, mapped into a spatial tessellation leveraging CNNs (both for generator and discriminator). The generated matrices are then processed and the origin (enter point) and destination (exit point) of the trips are extracted. The second GAN extracts road information from road maps using CNNs, and uses LSTMs to generate a trajectory between an origin road and a destination road on the road network. The model performance is evaluated using the dataset of taxis in Porto [123] by measuring the JS divergence between the distribution of trajectory lengths and the frequencies of the top 50 visited places.

4.2 Flow Generation

Flow generation consists of generating the flows between a set of geographic locations, given some locations' characteristics (e.g., population, POIs, land use, distance to other locations) and without any information about the real flows, [9]. Flow generation is crucial to many aspects of our society, such as transport planning [51] and spatial economics [93, 138, 145] to reduce inequalities and to design more sustainable communities, and the modeling of epidemic spreading patterns [7, 28, 110, 225]. Solving flow generation requires capturing the spatial patterns of close and distant flows, dependencies in the mobility network, and the characteristics of the locations.

Problem definition. Given a tessellation $\mathcal{G}$ over a region A, the flow, $y(g_i, g_j)$, between locations g_i and g_j represents the number of people moving from g_i to g_j. The total outflow, O_i, from location g_i is the total number of people originating from location g_i, i.e., $O_i = \sum_j y(g_i, g_j)$. Flow generation aims to estimate $y(g_i, g_j) \forall i, j \in \mathcal{G}$, $i \neq j$, i.e., the flow between each pair of locations (g_i, g_j) in given their total outflows O_i and O_j.

DL vs Traditional approaches. Flow generation has attracted interest for a long time ago. Notably, in 1946 George K. Zipf proposed a model to estimate mobility flows, drawing an analogy with Newton's law of universal gravitation [236]. This model, known as the gravity model, is based on the assumption that the number of travelers between two locations (flow) increases with the locations' populations while decreases with the distance

between them [9]. Despite the significant results achieved by the gravity model, it suffers from several drawbacks, including the inability to capture the structure of the real flows accurately and the more significant variability of real flows than expected [166]. Moreover, generations are done without considering other factors (e.g., POIs, street networks and others). Other mechanistic approaches aim to extend the gravity model by capturing additional information like the radiation model. All the models mentioned cannot capture non-linear relationships in the data and rely on a limited set of parameters.
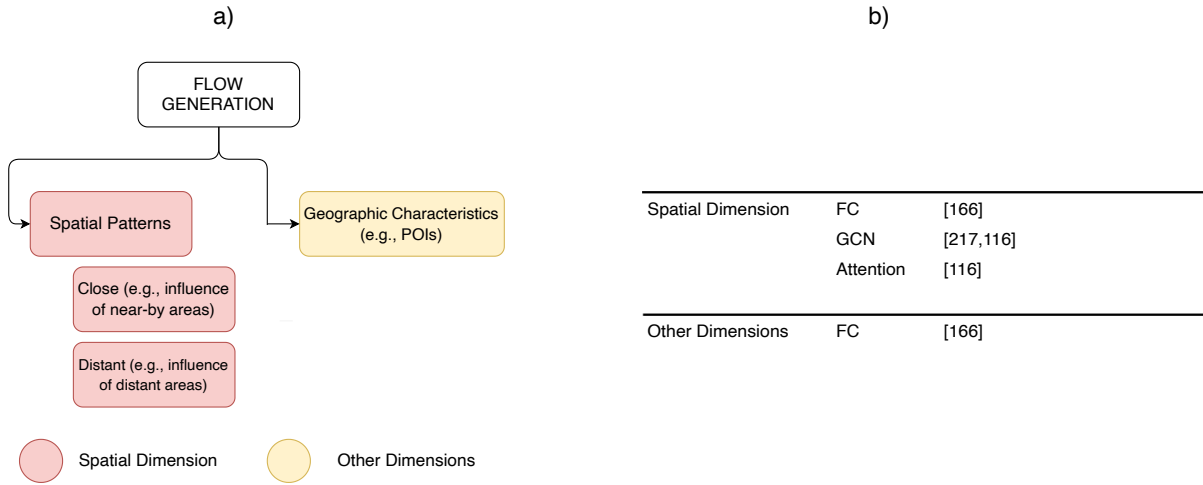

Fig. 5. **a)** The aspects flow generators should capture regarding the spatial dimension (red) and the social and geographic dimensions (yellow) of human mobility data. **b)** DL modules that allow to capture each dimension, with the reference to the selected papers in the literature that implement these modules.

In contrast, DL approaches can capture complex and non-linear relationships in the data and easily integrate additional information about locations such as population, POIs, population (Figure 5). For example, an FC can be fed with data representing POIs and, similarly, POIs can be a node feature in a graph neural network. Finally, CNNs can capture spatial relationships that traditional approaches cannot capture. Similarly, FC networks can be fed with additional features representing areas near the origin and the destination of a flow to characterize its spatial patterns.

Datasets and evaluation metrics. Flow generators are generally evaluated on commuting data from censuses from official statistics institutes [91, 166]. An alternative dataset is the set of traces by taxis in Beijing [217, 229] aggregated into flows (see Appendix C for details). Flow generation is commonly evaluated as the CPC (Common Part Of Commuters) between real and generated flows. Other metrics commonly used for this purpose are MAE, RMSE and MAPE. Additional information about these metrics can be found in Appendix D.3.

DL approaches. There is a limited literature tackling flow generation (Table 4).

DeepGravity [166] use FCs to extend the original gravity model with the ability to capture non-linear relationships and the possibility to integrate additional information to characterize the locations easily (e.g., population, POIs). The model is evaluated on commuting data from official statistics in the UK and Italy and a dataset of flows aggregated from individual trajectories in the US [91] using CPC and RMSE. The authors also provide meaningful explanations of the generated flows in terms of features of the origin and destination locations.

SI-GCN (Spatial Interaction GCN) [217] consists of three parts: *(i)* a layer for managing the spatial representation of the data (e.g., construct the local graph, negative sampling, features organization); followed by *(ii)* an encoder

	Reference	Name	Year	DL Modules	Evaluation	Dataset	Code (https://bit.ly)
F. G.	Yao et al. [217]	SI-GCN	2020	GCN	RMSE, MAPE, CPC	[229]	
	Simini et al. [166]	Deep Gravity	2020	FC	CPC	[91]	DeepGravity
	Liu et al. [116]	GMEL	2020	GNN, Attention	RMSE, MAE, CPC		GMEL-Code

Table 4. List of selected papers tackling flow generation. For each paper, we describe the name of the corresponding proposed model (if any), the year of publication, the DL modules used in the proposed solution, the metrics used for performance evaluation, the link to the public datasets used for training and evaluating the model, and the link to a repository with the code (if any). The papers are sorted by year of publication in decreasing order.

that uses graph convolutions to generate a representation in a latent space for all the geographical units; and *(iii)* a decoder in charge of generating the missing flows starting from the latent representation. SI-GCN is evaluated on the dataset of T-Drive [229] using RMSE, MAPE and CPC as evaluation metrics.

GML (Geocontextual Multitask Embedding Learner) [116] captures the spatial correlation from geographic contextual information and relies on two graph neural networks with attention (GAT). The role of the GATs is to learn an embedding representation that is then passed to a gradient boosting that is in charge of generating the flows. The model is evaluated on commuting data regarding New York City using MAE, RMSE, and CPC.

5 CONCLUSIONS

In this survey, we proposed a perspective on DL approaches to human mobility, focusing on next-location prediction, crowd flow prediction, trajectory generation, and flow generation. For each task, we highlighted the challenges related to solving it and how DL may help capture human mobility better than traditional models. We described a selection of relevant state-of-the-art solutions to each task. As an additional contribution to the community, we created a GitHub repository (bit.ly/DL4HM) where researchers can contribute to maintaining the list of relevant papers always up-to-date. Our survey reveals that predictive tasks (next-location and crowd flow prediction) are well established in the community and addressed by a significant variety of DL approaches. In contrast, the usage of DL for generative tasks (trajectory generation and flow generation) is more recent and should require more attention in the future.

Our overview of the state of the art of DL for human mobility reveals that existing solutions suffer from several limitations, and many relevant aspects need to be addressed in the future. In particular, we identify the following open challenges.

Geographic Transferability. Although DL models can capture complex mobility patterns automatically, they strictly depend on the data used for training and may not be *geographically transferable*, i.e., one model trained on a specific region can be used to predict locations or crowd flows or to generate synthetic trajectories on a distinct, non-overlapping region. Geographic transferability can be crucial in situations where there is a scarcity or even absence of mobility data for a region, and it poses several challenges related, for example, to the design of a suitable encoder of the mobility trajectories or flows. As a first tentative in this direction, RegionTrans [195] provides insights on how transferability can be tackled for crowd flow prediction. Still, more work is needed to address this open challenge.

Explainability. DL models are by nature *opaque*, i.e., they are black-boxes from which it is hard to reconstruct the reasoning that led to the generation of a trajectory or the prediction of a location or flow. Nonetheless, explainability is crucial for gaining a deeper understanding of mobility patterns and highlighting the presence of biases in the model's reasoning. It is important to develop mobility-related explanations that provide examples and counter-examples to validate trajectories and crowd flows from different perspectives. While models rely on many features, either external ones (e.g., weather data, POIs) or spatio-temporal ones, it is not clear what the role

of each feature is to the model's prediction or generation. Designing explainable DL models for human mobility is essential to gain knowledge that can be useful for possible users, such as policymakers and urban planners.

Privacy. DL models raise privacy issues both in the training and the prediction or generation phase. For example, in trajectory generation, evaluating the risk of re-identifying a real user from synthetic trajectories is crucial, especially when there is a scarcity of data to train the models. A synthetic trajectory may resemble a real one and a malicious adversary may use this information for re-identification. In the training phase, the risk of leaking private data is high regardless of the mobility task, as the portions of information used cannot be controlled directly. The extent training trajectories or crowd flows can be perturbed without degrading the realism of the generative models and the accuracy of predictive ones is an aspect that is barely investigated in the literature.

Tunability. Meaningful predictions and simulations require models that can be controlled along relevant mobility dimensions, such as geographic space, time granularity, presence of mobility restrictions or other events that may alter mobility, predictability of trajectories or crowd flows, and more. Current DL models have a limited degree of tunability, which limits their usability in practice. For example, it is not clear to what extent the models' realism or accuracy depends on the size or shape of the spatial tessellation, which may vary according to the user's needs. For example, suppose the decision-maker is interested in identifying the most active areas of a city in terms of daily mobility. In that case, the usage of administrative tiles may provide more interpretable results. In contrast, if they want to understand how the transportation and road networks affect and are affected by big events in a city, the usage of fine-grained grid-like tessellations may be helpful.

Interaction Dimension. Next-location predictors and trajectory generators assume the independence of the individuals' mobility, even though social purposes or collective needs can explain a significant portion of human movements. People's movements are not entirely independent of each other and they can lead to situations like traffic jams, accidents, or massive commuting patterns. Models that can include these aspects need to be designed to capture human mobility's complexity more comprehensively.

ACKNOWLEDGMENTS

REFERENCES

[1] Zain Ul Abideen, Heli Sun, Zhou Yang, Rana Zeeshan Ahmad, Adnan Iftekhar, and Amir Ali. 2021. Deep Wide Spatial-Temporal Based Transformer Networks Modeling for the Next Destination According to the Taxi Driver Behavior Prediction. *Applied Sciences* 11, 1 (2021), 17.

[2] Mohammed N Ahmed, Gianni Barlacchi, Stefano Braghin, Francesco Calabrese, Michele Ferretti, Vincent PA Lonij, Rahul Nair, Rana Novack, Jurij Paraszczak, and Andeep S Toor. 2016. A Multi-Scale Approach to Data-Driven Mass Migration Analysis.. In *SoGood@ ECML-PKDD.*

[3] Yi Ai, Zongping Li, Mi Gan, Yunpeng Zhang, Daben Yu, Wei Chen, and Yanni Ju. 2019. A deep learning approach on short-term spatiotemporal distribution forecasting of dockless bike-sharing system. *Neural Computing and Applications* 31, 5 (2019), 1665–1677.

[4] Laura Alessandretti, Ulf Aslak, and Sune Lehmann. 2020. The scales of human mobility. *Nature* 587, 7834 (2020), 402–407. https://doi.org/10.1038/s41586-020-2909-1

[5] Laura Alessandretti, Piotr Sapiezynski, Vedran Sekara, Sune Lehmann, and Andrea Baronchelli. 2018. Evidence for a conserved quantity in human mobility. *Nature Human Behaviour* 2, 7 (2018), 485–491.

[6] Daniel Ashbrook and Thad Starner. 2002. Learning significant locations and predicting user movement with GPS. In *Proceedings. Sixth International Symposium on Wearable Computers,*. IEEE, 101–108.

[7] Duygu Balcan, Bruno Gonçalves, Hao Hu, José J Ramasco, Vittoria Colizza, and Alessandro Vespignani. 2010. Modeling the spatial spread of infectious diseases: The GLobal Epidemic and Mobility computational model. *Journal of computational science* 1, 3 (2010), 132–145.

[8] Yi Bao, Zhou Huang, Linna Li, Yaoli Wang, and Yu Liu. 2020. A BiLSTM-CNN model for predicting users' next locations based on geotagged social media. *International Journal of Geographical Information Science* (2020), 1–22.

[9] Hugo Barbosa, Marc Barthelemy, Gourab Ghoshal, Charlotte R James, Maxime Lenormand, Thomas Louail, Ronaldo Menezes, José J Ramasco, Filippo Simini, and Marcello Tomasini. 2018. Human mobility: Models and applications. *Physics Reports* 734 (2018), 1–74.

[10] Hugo Barbosa, Fernando B de Lima-Neto, Alexandre Evsukoff, and Ronaldo Menezes. 2015. The effect of recency to human mobility. *EPJ Data Science* 4 (2015), 1–14.

[11] Gianni Barlacchi, Christos Perentis, Abhinav Mehrotra, Mirco Musolesi, and Bruno Lepri. 2017. Are you getting sick? Predicting influenza-like symptoms using human mobility behaviors. *EPJ Data Science* 6, 1 (2017), 27.

[12] Gianni Barlacchi, Alberto Rossi, Bruno Lepri, and Alessandro Moschitti. 2017. Structural semantic models for automatic analysis of urban areas. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 279–291.

[13] Armando Bazzani, Bruno Giorgini, Sandro Rambaldi, Riccardo Gallotti, and Luca Giovannini. 2010. Statistical laws in urban mobility from microscopic GPS data in the area of Florence. *Journal of Statistical Mechanics: Theory and Experiment* 2010, 05 (2010), P05001.

[14] Citi Bike. 2013. *Citi Bike System Data - NYC*. https://www.citibikenyc.com/system-data

[15] Capital Bikeshare. 2011. *Capital Bikeshare - Washington DC*. https://www.capitalbikeshare.com/system-data

[16] V. Bindschaedler and R. Shokri. 2016. Synthesizing Plausible Privacy-Preserving Location Traces. In *2016 IEEE Symposium on Security and Privacy (SP)*. 546–563.

[17] Justine I. Blanford, Zhuojie Huang, Alexander Savelyev, and Alan M. MacEachren. 2015. Geo-Located Tweets. Enhancing Mobility Maps and Capturing Cross-Border Movement. *PLOS ONE* 10, 6 (2015), 1–16.

[18] Vincent D. Blondel, Adeline Decuyper, and Gautier Krings. 2015. A survey of results on mobile phone datasets analysis. *EPJ Data Science* 4, 1 (2015), 10.

[19] M Bohm, Mirco Nanni, and Luca Pappalardo. 2021. Quantifying the presence of air pollutants over a road network in high spatio-temporal resolution. In *Climate Change AI, NeurIPS Workshop*.

[20] George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. 2015. *Time series analysis: forecasting and control*. John Wiley & Sons.

[21] Lorenzo Bracciale, Marco Bonola, Pierpaolo Loreti, Giuseppe Bianchi, Raul Amici, and Antonello Rabuffi. 2014. CRAWDAD dataset roma/taxi (v. 2014-07-17).

[22] Dirk Brockmann, Lars Hufnagel, and Theo Geisel. 2006. The scaling laws of human travel. *Nature* 439, 7075 (2006), 462–465.

[23] Ingrid Burbey and Thomas L Martin. 2012. A survey on predicting personal mobility. *International Journal of Pervasive Computing and Communications* (2012).

[24] F. Calabrese, G. Di Lorenzo, L. Liu, and C. Ratti. 2011. Estimating Origin-Destination Flows Using Mobile Phone Location Data. *IEEE Pervasive Computing* (2011), 36–44.

[25] Francesco Calabrese, Giusy Di Lorenzo, and Carlo Ratti. 2010. Human mobility prediction based on individual and collective geographical preferences. In *13th international IEEE conference on intelligent transportation systems*. 312–317.

[26] Luca Canzian and Mirco Musolesi. 2015. Trajectories of depression: unobtrusive monitoring of depressive states by means of smartphone mobility traces analysis. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. 1293–1304.

[27] Justin David Carlson. 2010. *Mapping Large, Urban Environments with GPS-Aided SLAM*. Ph.D. Dissertation. Carnegie Mellon University.

[28] Serhan Cevik. 2020. Going Viral: A Gravity Model of Infectious Diseases and Tourism Flows. (2020).

[29] Shwu-Jing Chang, Gong-Ying Hsu, Jia-Ao Yang, Kuan-Ning Chen, Yung-Fang Chiu, and Fu-Tong Chang. 2010. Vessel traffic analysis for maritime Intelligent Transportation System. In *2010 IEEE 71st Vehicular Technology Conference*. IEEE, 1–4.

[30] Guangshuo Chen, Aline Carneiro Viana, Marco Fiore, and Carlos Sarraute. 2019. Complete trajectory reconstruction from sparse mobile phone data. *EPJ Data Science* 8, 1 (2019), 30.

[31] J. Chen, Z. Xiao, D. Wang, W. Long, and V. Havyarimana. 2019. Stay of Interest: A Dynamic Spatiotemporal Stay Behavior Perception Method for Private Car Users. In *2019 IEEE 21st International Conference on High Performance Computing and Communications*. 1526–1532.

[32] Qi Chen, Wei Wang, Fangyu Wu, Suparna De, Ruili Wang, Bailing Zhang, and Xin Huang. 2019. A survey on an emerging area: Deep learning for smart city data. *IEEE Transactions on Emerging Topics in Computational Intelligence* 3, 5 (2019), 392–410.

[33] Yile Chen, Cheng Long, Gao Cong, and Chenliang Li. 2020. Context-aware Deep Model for Joint Mobility and Time Prediction. In *Proceedings of the 13th International Conference on Web Search and Data Mining*. 106–114.

[34] Eunjoon Cho, Seth A Myers, and Jure Leskovec. 2011. Friendship and mobility: user movement in location-based social networks. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. 1082–1090.

[35] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 1724–1734.

[36] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078* (2014).

[37] Edward Choi, Mohammad Taha Bahadori, Jimeng Sun, Joshua Kulas, Andy Schuetz, and Walter Stewart. 2016. Retain: An interpretable predictive model for healthcare using reverse time attention mechanism. In *Advances in Neural Information Processing Systems*. 3504–3512.

[38] Jan K Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio. 2015. Attention-based models for speech recognition. In *Advances in neural information processing systems*. 577–585.

[39] Giuliano Cornacchia and Luca Pappalardo. 2021. STS-EPR: Modelling individual mobility considering the spatial, temporal, and social dimensions together. *Procedia Computer Science* 184 (2021), 258–265. https://doi.org/10.1016/j.procs.2021.03.035 The 12th International Conference on Ambient Systems, Networks and Technologies (ANT) / The 4th International Conference on Emerging Data and Industry 4.0 (EDI40) / Affiliated Workshops.

[40] Balázs Cs Csáji, Arnaud Browet, Vincent A Traag, Jean-Charles Delvenne, Etienne Huens, Paul Van Dooren, Zbigniew Smoreda, and Vincent D Blondel. 2013. Exploring the mobility of mobile phone users. *Physica A: statistical mechanics and its applications* 392, 6 (2013), 1459–1473.

[41] Yilan Cui, Xing Xie, and Yi Liu. 2018. Social media and mobility landscape: Uncovering spatial patterns of urban human mobility with multi source data. *Frontiers of Environmental Science & Engineering* 12, 5 (2018), 7.

[42] Susan L Cutter, Joseph A Ahearn, Bernard Amadei, Patrick Crawford, Elizabeth A Eide, Gerald E Galloway, Michael F Goodchild, Howard C Kunreuther, Meredith Li-Vollmer, Monica Schoch-Spana, et al. 2013. Disaster resilience: A national imperative. *Environment: Science and Policy for Sustainable Development* 55, 2 (2013), 25–29.

[43] Genan Dai, Xiaoyang Hu, Youming Ge, Zhiqing Ning, and Yubao Liu. 2021. Attention based simplified deep residual network for citywide crowd flows prediction. *Frontiers of Computer Science* 15, 2 (2021), 1–12.

[44] Alexandre De Brébisson, Étienne Simon, Alex Auvolat, Pascal Vincent, and Yoshua Bengio. 2015. Artificial neural networks applied to taxi destination prediction. In *Proceedings of the 2015th International Conference on ECML PKDD Discovery Challenge*. 40–51.

[45] Yves-Alexandre de Montjoye, Sébastien Gambs, Vincent Blondel, Geoffrey Canright, Nicolas de Cordes, Sébastien Deletaille, Kenth Engø-Monsen, Manuel Garcia-Herranz, Jake Kendall, Cameron Kerry, Gautier Krings, Emmanuel Letouzé, Miguel Luengo-Oroz, Nuria Oliver, Luc Rocher, Alex Rutherford, Zbigniew Smoreda, Jessica Steele, Erik Wetter, Alex "Sandy" Pentland, and Linus Bengtsson. 2018. On the privacy-conscientious use of mobile phone data. *Scientific Data* 5, 1 (2018), 180286.

[46] Pierre Deville, Catherine Linard, Samuel Martin, Marius Gilbert, Forrest R Stevens, Andrea E Gaughan, Vincent D Blondel, and Andrew J Tatem. 2014. Dynamic population mapping using mobile phone data. *Proceedings of the National Academy of Sciences* 111, 45 (2014), 15888–15893.

[47] Bowen Du, Hao Peng, Senzhang Wang, Md Zakirul Alam Bhuiyan, Lihong Wang, Qiran Gong, Lin Liu, and Jing Li. 2019. Deep irregular convolutional residual LSTM for urban traffic passenger flows prediction. *IEEE Transactions on Intelligent Transportation Systems* 21, 3 (2019), 972–985.

[48] Nathan Eagle and Alex Sandy Pentland. 2009. Eigenbehaviors: identifying structure in routine. *Behavioral Ecology and Sociobiology* 63, 11 (2009), 1689–1689.

[49] Patrick Ebel, Ibrahim Emre Göl, Christoph Lingenfelder, and Andreas Vogelsang. 2020. Destination Prediction Based on Partial Trajectory Data. *arXiv preprint arXiv:2004.07473* (2020).

[50] Zeinab Ebrahimpour, Wanggen Wan, Ofelia Cervantes, Tianhang Luo, and Hidayat Ullah. 2019. Comparison of main approaches for extracting behavior features from crowd flow analysis. *ISPRS International Journal of Geo-Information* 8, 10 (2019), 440.

[51] Sven Erlander and Neil F Stewart. 1990. *The gravity model in transportation analysis: theory and extensions*. Vol. 3. Vsp.

[52] Cristóbal Esteban, Stephanie L Hyland, and Gunnar Rätsch. 2017. Real-valued (medical) time series generation with recurrent conditional gans. *arXiv preprint arXiv:1706.02633* (2017).

[53] Clement Farabet, Camille Couprie, Laurent Najman, and Yann LeCun. 2013. Learning Hierarchical Features for Scene Labeling. *IEEE Trans. Pattern Anal. Mach. Intell.* 35, 8 (2013), 1915–1929.

[54] Jie Feng, Yong Li, Chao Zhang, Funing Sun, Fanchao Meng, Ang Guo, and Depeng Jin. 2018. Deepmove: Predicting human mobility with attentional recurrent networks. In *Proceedings of the 2018 world wide web conference*. 1459–1468.

[55] Jie Feng, Zeyu Yang, Fengli Xu, Haisu Yu, Mudan Wang, and Yong Li. 2020. Learning to Simulate Human Mobility. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (Virtual Event, CA, USA) *(KDD '20)*. Association for Computing Machinery, New York, NY, USA, 3426–3433. https://doi.org/10.1145/3394486.3412862

[56] Z. Feng and Y. Zhu. 2016. A Survey on Trajectory Data Mining: Techniques and Applications. *IEEE Access* 4 (2016), 2056–2067.

[57] V. Fernandez Arguedas, G. Pallotta, and M. Vespe. 2018. Maritime Traffic Networks: From Historical Positioning Data to Unsupervised Maritime Traffic Monitoring. *IEEE Transactions on Intelligent Transportation Systems* 19, 3 (2018), 722–732.

[58] Michele Ferretti, Gianni Barlacchi, Luca Pappalardo, Lorenzo Lucchini, and Bruno Lepri. 2018. Weak nodes detection in urban transport systems: Planning for resilience in Singapore. In *2018 IEEE 5th international conference on data science and advanced analytics (DSAA)*. IEEE, 472–480.

[59] Marco Fiore, Panagiota Katsikouli, Elli Zavou, Mathieu Cunche, Franccoise Fessant, Dominique Le Hello, Ulrich Matchi Aivodji, Baptiste Olivier, Tony Quertier, and Razvan Stanica. 2019. Privacy in trajectory micro-data publishing : a survey. *arXiv: Cryptography and*

Security (2019).

[60] Riccardo Gallotti, Armando Bazzani, Mirko Degli Esposti, and Sandro Rambaldi. 2013. Entropic measures of individual mobility patterns. *Journal of Statistical Mechanics: Theory and Experiment* 2013, 10 (2013), P10022.

[61] Riccardo Gallotti, Armando Bazzani, Sandro Rambaldi, and Marc Barthelemy. 2016. A stochastic model of randomly accelerated walkers for human mobility. *Nature Communications* 7, 1 (2016), 12600.

[62] Sébastien Gambs, Marc-Olivier Killijian, and Miguel Núñez del Prado Cortez. 2010. Show me how you move and I will tell you who you are. In *Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Security and Privacy in GIS and LBS*. 34–41.

[63] Sébastien Gambs, Marc-Olivier Killijian, and Miguel Núñez del Prado Cortez. 2012. Next place prediction using mobility markov chains. In *Proceedings of the First Workshop on Measurement, Privacy, and Mobility*. 1–6.

[64] Qiang Gao, Fan Zhou, Goce Trajcevski, Kunpeng Zhang, Ting Zhong, and Fengli Zhang. 2019. Predicting human mobility via variational attention. In *The World Wide Web Conference*. 2750–2756.

[65] Marta C Gonzalez, Cesar A Hidalgo, and Albert-Laszlo Barabasi. 2008. Understanding individual human mobility patterns. *nature* 453, 7196 (2008), 779–782.

[66] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. 2016. *Deep learning*. Vol. 1. MIT press Cambridge.

[67] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative Adversarial Nets. In *Proceedings of the 27th International Conference on Neural Information Processing Systems*. 2672–2680.

[68] Alex Graves, Navdeep Jaitly, and Abdel-rahman Mohamed. 2013. Hybrid speech recognition with deep bidirectional LSTM. In *2013 IEEE workshop on automatic speech recognition and understanding*. IEEE, 273–278.

[69] Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural networks* 18, 5-6 (2005), 602–610.

[70] Clark L Gray and Valerie Mueller. 2012. Natural disasters and population mobility in Bangladesh. *Proceedings of the National Academy of Sciences* 109, 16 (2012), 6000–6005.

[71] Aditya Grover and Jure Leskovec. 2016. Node2vec: Scalable Feature Learning for Networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Association for Computing Machinery, 855–864.

[72] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. 2017. Improved Training of Wasserstein GANs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*. 5769–5779.

[73] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[74] Andrea Hess, Karin Anna Hummel, Wilfried N Gansterer, and Günter Haring. 2015. Data-driven human mobility modeling: a survey and engineering guidance for mobile networking. *ACM Computing Surveys (CSUR)* 48, 3 (2015), 1–39.

[75] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[76] D. Huang, X. Song, Z. Fan, R. Jiang, R. Shibasaki, Y. Zhang, H. Wang, and Y. Kato. 2019. A Variational Autoencoder Based Generative Model of Urban Human Mobility. In *2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*. 425–430.

[77] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. 2017. Densely Connected Convolutional Networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2261–2269.

[78] D H HUBEL and T N WIESEL. 1959. Receptive fields of single neurones in the cat's striate cortex. *The Journal of physiology* 148, 3 (1959), 574–591.

[79] D. H. Hubel and T. N. Wiesel. 1962. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of Physiology* 160, 1 (1962), 106–154.

[80] Tomoharu Iwata and Hitoshi Shimizu. 2019. Neural collective graphical models for estimating spatio-temporal population flow from aggregated data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 3935–3942.

[81] Kasthuri Jayarajah, Andrew Tan, and Archan Misra. 2018. Understanding the Interdependency of Land Use and Mobility for Urban Planning. In *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers*. Association for Computing Machinery, 1079–1087.

[82] Renhe Jiang, Zekun Cai, Zhaonan Wang, Chuang Yang, Zipei Fan, Quanjun Chen, Kota Tsubouchi, Xuan Song, and Ryosuke Shibasaki. 2021. DeepCrowd: A Deep Model for Large-Scale Citywide Crowd Density and Flow Prediction. *IEEE Transactions on Knowledge and Data Engineering* (2021).

[83] Renhe Jiang, Xuan Song, Zipei Fan, Tianqi Xia, Quanjun Chen, Satoshi Miyazawa, and Ryosuke Shibasaki. 2018. Deepurbanmomentum: An online deep-learning system for short-term urban mobility prediction. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

[84] Shan Jiang, Gaston A. Fiore, Yingxiang Yang, Joseph Ferreira, Emilio Frazzoli, and Marta C. González. 2013. A Review of Urban Computing for Mobile Phone Traces: Current Methods, Challenges and Opportunities. In *Proceedings of the 2nd ACM SIGKDD International Workshop on Urban Computing*.

[85] Shan Jiang, Yingxiang Yang, Siddharth Gupta, Daniele Veneziano, Shounak Athavale, and Marta C. Gonzalez. 2016. The TimeGeo modeling framework for urban mobility without travel surveys. *Proceedings of the National Academy of Sciences* 113 (2016), 201524261.

[86] Weiwei Jiang and Jiayun Luo. 2021. Graph Neural Network for Traffic Forecasting: A Survey. arXiv:cs.LG/2101.11174
[87] Wenwei Jin, Youfang Lin, Zhihao Wu, and Huaiyu Wan. 2018. Spatio-Temporal Recurrent Convolutional Networks for Citywide Short-Term Crowd Flows Prediction.
[88] Raja Jurdak, Kun Zhao, Jiajun Liu, Maurice AbouJaoude, Mark Cameron, and David Newth. 2015. Understanding Human Mobility from Twitter. *PLOS ONE* 10, 7 (2015), 1–16.
[89] Yiannis Kamarianakis and Poulicos Prastacos. 2003. Forecasting traffic flow conditions in an urban network: Comparison of multivariate and univariate approaches. *Transportation Research Record* 1857, 1 (2003), 74–84.
[90] Yiannis Kamarianakis and Poulicos Prastacos. 2005. Space–time modeling of traffic flow. *Computers & Geosciences* 31, 2 (2005), 119–133.
[91] Yuhao Kang, Song Gao, Yunlei Liang, Mingxiao Li, Jinmeng Rao, and Jake Kruse. 2020. Multiscale dynamic human mobility flow dataset in the US during the COVID-19 epidemic. *Scientific data* 7, 1 (2020), 1–13.
[92] Dmytro Karamshuk, Chiara Boldrini, Marco Conti, and Andrea Passarella. 2011. Human mobility models for opportunistic networks. *IEEE Communications Magazine* 49, 12 (2011), 157–165.
[93] David Karemera, Victor Iwuagwu Oguledo, and Bobby Davis. 2000. A gravity model analysis of international migration to North America. *Applied economics* 32, 13 (2000), 1745–1755.
[94] L. Khaidem, M. Luca, F. Yang, A. Anand, B. Lepri, and W. Dong. 2020. Optimizing Transportation Dynamics at a City-Scale Using a Reinforcement Learning Framework. *IEEE Access* 8 (2020), 171528–171541.
[95] Asifullah Khan, Anabia Sohail, Umme Zahoora, and Aqsa Saeed Qureshi. 2020. A survey of the recent architectures of deep convolutional neural networks. *Artificial Intelligence Review* (2020).
[96] Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013).
[97] Niko Kiukkonen, Jan Blom, Olivier Dousse, Daniel Gatica-Perez, and Juha Laurila. 2010. Towards rich mobile phone datasets: Lausanne data collection campaign. *Proc. ICPS, Berlin* 68 (2010).
[98] J. F. Kolen and S. C. Kremer. 2001. *Gradient Flow in Recurrent Nets: The Difficulty of Learning LongTerm Dependencies.* 237–243.
[99] Dejiang Kong and Fei Wu. 2018. HST-LSTM: A Hierarchical Spatial-Temporal Long-Short Term Memory Network for Location Prediction.. In *IJCAI*. 2341–2347.
[100] Vartika Koolwal and Krishna Kumar Mohbey. 2020. A comprehensive survey on trajectory-based location prediction. *Iran Journal of Computer Science* 3, 2 (2020), 65–91.
[101] Moritz UG Kraemer, Chia-Hung Yang, Bernardo Gutierrez, Chieh-Hsi Wu, Brennan Klein, David M Pigott, Louis Du Plessis, Nuno R Faria, Ruoran Li, William P Hanage, et al. 2020. The effect of human mobility and control measures on the COVID-19 epidemic in China. *Science* 368, 6490 (2020), 493–497.
[102] Mark A Kramer. 1991. Nonlinear principal component analysis using autoassociative neural networks. *AIChE journal* 37, 2 (1991), 233–243.
[103] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger (Eds.). Curran Associates, Inc., 1097–1105.
[104] Vaibhav Kulkarni, Natasa Tagasovska, Thibault Vatter, and Benoit Garbinato. 2018. Generative models for simulating mobility trajectories. *arXiv preprint arXiv:1811.12801* (2018).
[105] Shengjie Lai, Andrea Farnham, Nick W Ruktanonchai, and Andrew J Tatem. 2019. Measuring mobility, disease connectivity and individual risk: a review of using mobile phone data and Health for travel medicine. *Journal of travel medicine* 26, 3 (2019).
[106] Juha K Laurila, Daniel Gatica-Perez, Imad Aad, Olivier Bornet, Trinh-Minh-Tri Do, Olivier Dousse, Julien Eberle, Markus Miettinen, et al. 2012. *The mobile data challenge: Big data for mobile computing research.* Technical Report.
[107] Sangsoo Lee and Daniel B Fambro. 1999. Application of subset autoregressive integrated moving average model for short-term freeway traffic volume forecasting. *Transportation Research Record* 1678, 1 (1999), 179–188.
[108] Maxime Lenormand, Aleix Bassolas, and José J Ramasco. 2016. Systematic comparison of trip distribution laws and models. *Journal of Transport Geography* 51 (2016), 158–169.
[109] Wei Li, Wei Tao, Junyang Qiu, Xin Liu, Xingyu Zhou, and Zhisong Pan. 2019. Densely Connected Convolutional Networks With Attention LSTM for Crowd Flows Prediction. *IEEE Access* 7 (2019), 140488–140498.
[110] Xinhai Li, Huidong Tian, Dejian Lai, and Zhibin Zhang. 2011. Validation of the gravity model in predicting the global spread of influenza. *International journal of environmental research and public health* 8, 8 (2011), 3134–3143.
[111] Yuan Liao, Sonia Yeh, and Gustavo S Jeuken. 2019. From individual to collective behaviours: exploring population heterogeneity of human mobility based on social media data. *EPJ Data Science* 8, 1 (2019), 34.
[112] Ziqian Lin, Jie Feng, Ziyang Lu, Yong Li, and Depeng Jin. 2019. Deepstn+: Context-aware spatial-temporal neural network for crowd flow prediction in metropolis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 1020–1027.
[113] Lingbo Liu, Jiajie Zhen, Guanbin Li, Geng Zhan, Zhaocheng He, Bowen Du, and Liang Lin. 2020. Dynamic Spatial-Temporal Representation Learning for Traffic Flow Prediction. *IEEE Transactions on Intelligent Transportation Systems* (2020).

[114] Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. 2016. Predicting the next location: A recurrent model with spatial and temporal contexts. In *Thirtieth AAAI conference on artificial intelligence*.

[115] Xi Liu, Hanzhou Chen, and Clio Andris. 2018. trajGANs: Using generative adversarial networks for geo-privacy protection of trajectory data (Vision paper). In *Location Privacy and Security Workshop*. 1–7.

[116] Zhicheng Liu, Fabio Miranda, Weiting Xiong, Junyan Yang, Qiao Wang, and Claudio Silva. 2020. Learning geo-contextual embeddings for commuting flow prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 808–816.

[117] Massimiliano Luca, Gianni Barlacchi, Nuria Oliver, and Bruno Lepri. 2021. Leveraging Mobile Phone Data for Migration Flows. arXiv:cs.CY/2105.14956

[118] Jianming Lv, Qing Li, Qinghui Sun, and Xintong Wang. 2018. T-CONV: A convolutional neural network for multi-scale taxi trajectory prediction. In *2018 IEEE international conference on big data and smart computing (bigcomp)*. 82–89.

[119] Jean Damascène Mazimpaka and Sabine Timpf. 2016. Trajectory data mining: A review of methods and applications. *Journal of Spatial Information Science* 2016, 13 (2016), 61–99.

[120] D. J. Mir, S. Isaacman, R. Cáceres, M. Martonosi, and R. N. Wright. 2013. DP-WHERE: Differentially private modeling of human mobility. In *2013 IEEE International Conference on Big Data*. 580–588.

[121] Anna Monreale, Fabio Pinelli, Roberto Trasarti, and Fosca Giannotti. 2009. Wherenext: a location predictor on trajectory pattern mining. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. 637–646.

[122] CK Moorthy and BG Ratcliffe. 1988. Short term traffic forecasting using time series methods. *Transportation planning and technology* 12, 1 (1988), 45–56.

[123] Luis Moreira-Matias, Joao Gama, Michel Ferreira, Joao Mendes-Moreira, and Luis Damas. 2013. Predicting taxi–passenger demand using streaming data. *IEEE Transactions on Intelligent Transportation Systems* 14, 3 (2013), 1393–1402.

[124] Lablack Mourad, Heng Qi, Yanming Shen, and Baocai Yin. 2019. ASTIR: Spatio-Temporal Data Mining for Crowd Flow Prediction. *IEEE Access* 7 (2019), 175159–175165.

[125] Anastasios Noulas, Salvatore Scellato, Renaud Lambiotte, Massimiliano Pontil, and Cecilia Mascolo. 2012. A Tale of Many Cities: Universal Patterns in Human Urban Mobility. *PLOS ONE* 7, 5 (2012), 1–10.

[126] MM Nyhan, I Kloog, R Britter, C Ratti, and P Koutrakis. 2019. Quantifying population exposure to air pollution using individual mobility patterns inferred from mobile phone data. *Journal of exposure science & environmental epidemiology* 29, 2 (2019), 238.

[127] Nuria Oliver, Bruno Lepri, Harald Sterly, Renaud Lambiotte, Sébastien Deletaille, Marco De Nadai, Emmanuel Letouzé, Albert Ali Salah, Richard Benjamins, Ciro Cattuto, et al. 2020. Mobile phone data for informing public health actions across the COVID-19 pandemic life cycle.

[128] Kun Ouyang, Reza Shokri, David S Rosenblum, and Wenzhuo Yang. 2018. A Non-Parametric Generative Model for Human Trajectories.. In *IJCAI*. 3812–3817.

[129] Xi Ouyang, Chaoyun Zhang, Pan Zhou, Hao Jiang, and Shimin Gong. 2016. Deepspace: An online deep learning framework for mobile big data to understand human mobility patterns. *arXiv preprint arXiv:1610.07009* (2016).

[130] Luca Pappalardo, Giuliano Cornacchia, Victor Navarro, Loreto Bravo, and Leo Ferres. 2020. A dataset to assess mobility changes in Chile following local quarantines. arXiv:physics.soc-ph/2011.12162

[131] Luca Pappalardo, Leo Ferres, Manuel Sacasa, Ciro Cattuto, and Loreto Bravo. 2020. An individual-level ground truth dataset for home location detection. arXiv:2010.08814

[132] Luca Pappalardo, Leo Ferres, Manuel Sacasa, Ciro Cattuto, and Loreto Bravo. 2021. Evaluation of home detection algorithms on mobile phone data using individual-level ground truth. *EPJ Data Science* 10, 1 (2021), 29. https://doi.org/10.1140/epjds/s13688-021-00284-9

[133] Luca Pappalardo, Salvatore Rinzivillo, Zehui Qu, Dino Pedreschi, and Fosca Giannotti. 2013. Understanding the patterns of car travel. *The European Physical Journal Special Topics* 215, 1 (2013), 61–73.

[134] Luca Pappalardo and Filippo Simini. 2018. Data-driven generation of spatio-temporal routines in human mobility. *Data Mining and Knowledge Discovery* 32, 3 (2018), 787–829.

[135] Luca Pappalardo, F Simini, G Barlacchi, and R Pellungrini. 2019. scikit-mobility: A Python library for the analysis, generation and risk assessment of mobility data. *arXiv preprint arXiv:1907.07062* (2019).

[136] Luca Pappalardo, Filippo Simini, Salvatore Rinzivillo, Dino Pedreschi, Fosca Giannotti, and Albert-László Barabási. 2015. Returners and explorers dichotomy in human mobility. *Nature Communications* 6, 1 (2015), 8166.

[137] Luca Pappalardo, Maarten Vanhoof, Lorenzo Gabrielli, Zbigniew Smoreda, Dino Pedreschi, and Fosca Giannotti. 2016. An analytical framework to nowcast well-being using mobile phone data. *International Journal of Data Science and Analytics* (2016), 75–92.

[138] Roberto Patuelli, Aura Reggiani, Sean P Gorman, Peter Nijkamp, and Franz-Josef Bade. 2007. Network analysis of commuting flows: A comparative static approach to German data. *Networks and Spatial Economics* 7, 4 (2007), 315–331.

[139] Roberto Pellungrini, Luca Pappalardo, Francesca Pratesi, and Anna Monreale. 2017. A Data Mining Approach to Assess Privacy Risk in Human Mobility Data. *ACM Transactions on Intelligent Systems and Technologies* 9, 3 (2017).

[140] R. Pellungrini, L. Pappalardo, F. Simini, and A. Monreale. 2020. Modeling Adversarial Behavior Against Mobility Data Privacy. *IEEE Transactions on Intelligent Transportation Systems* (2020), 1–14.

[141] Emanuele Pepe, Paolo Bajardi, Laetitia Gauvin, Filippo Privitera, Brennan Lake, Ciro Cattuto, and Michele Tizzoni. 2020. COVID-19 outbreak response, a dataset to assess mobility changes in Italy following national lockdown. *Scientific data* 7, 1 (2020), 1–7.

[142] Christos Perentis, Michele Vescovi, Chiara Leonardi, Corrado Moiso, Mirco Musolesi, Fabio Pianesi, and Bruno Lepri. 2017. Anonymous or Not? Understanding the Factors Affecting Personal Mobile Data Disclosure. *ACM Trans. Internet Technol.* 17, 2 (2017).

[143] Michal Piorkowski, Natasa Sarafijanovic-Djukic, and Matthias Grossglauser. 2009. CRAWDAD dataset epfl/mobility (v. 2009-02-24).

[144] Rafael Prieto Curiel, Luca Pappalardo, Lorenzo Gabrielli, and Steven Richard Bishop. 2018. Gravity and scaling laws of city to city migration. *PLOS ONE* 13, 7 (2018), 1–19.

[145] Rafael Prieto Curiel, Luca Pappalardo, Lorenzo Gabrielli, and Steven Richard Bishop. 2018. Gravity and scaling laws of city to city migration. *PLOS ONE* 13, 7 (07 2018), 1–19. https://doi.org/10.1371/journal.pone.0199892

[146] F. Rebelo, C. Soares, and R. J. F. Rossetti. 2015. TwitterJam: Identification of mobility patterns in urban centers based on tweets. In *2015 IEEE First International Smart Cities Conference (ISC2)*. 1–6.

[147] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems*. 91–99.

[148] Yibin Ren, Huanfa Chen, Yong Han, Tao Cheng, Yang Zhang, and Ge Chen. 2020. A hybrid integrated deep learning model for the prediction of citywide spatio-temporal flow volumes. *International Journal of Geographical Information Science* 34, 4 (2020), 802–823.

[149] Rafael Reuveny. 2007. Climate change-induced migration and violent conflict. *Political geography* 26, 6 (2007), 656–673.

[150] S. Rinzivillo, L. Gabrielli, M. Nanni, L. Pappalardo, D. Pedreschi, and F. Giannotti. 2014. The purpose of motion: Learning activities from Individual Mobility Networks. In *2014 International Conference on Data Science and Advanced Analytics (DSAA)*. 312–318.

[151] Maria Riveiro, Giuliana Pallotta, and Michele Vespe. 2018. Maritime anomaly detection: A review. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 8, 5 (2018), e1266.

[152] Can Rong, Jie Feng, and Yong Li. 2019. Deep learning models for population flow generation from aggregated mobility data. In *Adjunct Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2019 ACM International Symposium on Wearable Computers*. 1008–1013.

[153] Alberto Rossi, Gianni Barlacchi, Monica Bianchini, and Bruno Lepri. 2019. Modelling Taxi Drivers' Behaviour for the Next Destination Prediction. *IEEE Transactions on Intelligent Transportation Systems* (2019).

[154] Alessio Rossi, Luca Pappalardo, Paolo Cintia, F. Marcello Iaia, Javier Fernández, and Daniel Medina. 2018. Effective injury forecasting in soccer with GPS training data and machine learning. *PLOS ONE* 13, 7 (2018), 1–15.

[155] Luca Rossi, Matthew J. Williams, Christopher Stich, and Mirco Musolesi. 2015. Privacy and the City: User Identification and Location Semantics in Location-Based Social Networks. In *Proceedings of the 9th International AAAI Conference on Web and Social Media*.

[156] N. W. Ruktanonchai, J. R. Floyd, S. Lai, C. W. Ruktanonchai, A. Sadilek, P. Rente-Lourenco, X. Ben, A. Carioli, J. Gwinn, J. E. Steele, O. Prosper, A. Schneider, A. Oplinger, P. Eastham, and A. J. Tatem. 2020. Assessing the impact of coordinated COVID-19 exit strategies across Europe. *Science* 369, 6510 (2020), 1465–1470.

[157] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. 1986. *Learning Internal Representations by Error Propagation*. MIT Press, 318–362.

[158] T. Russo, L. D'Andrea, A. Parisi, M. Martinelli, A. Belardinelli, F. Boccoli, I. Cignini, M. Tordoni, and S. Cataudella. 2016. Assessing the fishing footprint using data integrated from different tracking devices: Issues and opportunities. *Ecological Indicators* 69 (2016), 818 – 827.

[159] BA Sabarish, R Karthi, and T Gireeshkumar. 2015. A survey of location prediction using trajectory mining. In *Artificial Intelligence and Evolutionary Algorithms in Engineering Systems*. Springer, 119–127.

[160] Christian M. Schneider, Vitaly Belik, Thomas Couronné, Zbigniew Smoreda, and Marta C. González. 2013. Unravelling daily human mobility motifs. *Journal of The Royal Society Interface* 10, 84 (2013), 20130246.

[161] M. Schuster and K. K. Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing* 45, 11 (1997), 2673–2681.

[162] Sungyong Seo, Jing Huang, Hao Yang, and Yan Liu. 2017. Interpretable convolutional neural networks with dual local and global attention for review rating prediction. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*. 297–305.

[163] Alex Sherstinsky. 2020. Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. *Physica D: Nonlinear Phenomena* 404 (2020), 132306.

[164] Yan Shi, Haoran Feng, Xiongfei Geng, Xingui Tang, and Yongcai Wang. 2019. A Survey of Hybrid Deep Learning Methods for Traffic Flow Prediction. In *Proceedings of the 2019 3rd International Conference on Advances in Image Processing*. Association for Computing Machinery, 133–138.

[165] Seungjae Shin, Hongseok Jeon, Chunglae Cho, Seunghyun Yoon, and Taeyeon Kim. 2020. User Mobility Synthesis based on Generative Adversarial Networks: A Survey. In *2020 22nd International Conference on Advanced Communication Technology (ICACT)*. IEEE, 94–103.

[166] Filippo Simini, Gianni Barlacchi, Massimiliano Luca, and Luca Pappalardo. 2020. Deep Gravity: enhancing mobility flows generation with deep neural networks and geographic information. arXiv:cs.LG/2012.00489

[167] Filippo Simini, Marta C. González, Amos Maritan, and Albert-László Barabási. 2012. A universal model for mobility and migration patterns. *Nature* 484, 7392 (2012), 96–100.

[168] Karen Simonyan and Andrew Zisserman. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *Proceedings of the 3rd International Conference on Learning Representations*, Yoshua Bengio and Yann LeCun (Eds.).

[169] Alina Sîrbu, Gennady Andrienko, Natalia Andrienko, Chiara Boldrini, Marco Conti, Fosca Giannotti, Riccardo Guidotti, Simone Bertoli, Jisu Kim, Cristina Ioana Muntean, et al. 2020. Human migration: the big data perspective. *International Journal of Data Science and Analytics* (2020), 1–20.

[170] B. H. Soleimani, E. N. De Souza, C. Hilliard, and S. Matwin. 2015. Anomaly detection in maritime data based on geometrical analysis of trajectories. In *2015 18th International Conference on Information Fusion (Fusion)*. 1100–1105.

[171] G. Solmaz and D. Turgut. 2019. A Survey of Human Mobility Models. *IEEE Access* 7 (2019), 125711–125731.

[172] Chaoming Song, Tal Koren, Pu Wang, and Albert-László Barabási. 2010. Modelling the scaling properties of human mobility. *Nature Physics* 6, 10 (2010), 818–823.

[173] Chaoming Song, Zehui Qu, Nicholas Blumm, and Albert-László Barabási. 2010. Limits of predictability in human mobility. *Science* 327, 5968 (2010), 1018–1021.

[174] H. Y. Song, M. S. Baek, and M. Sung. 2019. Generating Human Mobility Route Based on Generative Adversarial Network. In *2019 Federated Conference on Computer Science and Information Systems (FedCSIS)*. 91–99.

[175] Xuan Song, Quanshi Zhang, Yoshihide Sekimoto, Ryosuke Shibasaki, Nicholas Jing Yuan, and Xing Xie. 2016. Prediction and Simulation of Human Mobility Following Natural Disasters. *ACM Transactions on Intelligent Systems and Technologies* 8, 2 (2016).

[176] Victor Soto, Vanessa Frias-Martinez, Jesus Virseda, and Enrique Frias-Martinez. 2011. Prediction of socioeconomic levels using cell phone records. In *International Conference on User Modeling, Adaptation, and Personalization*. 377–388.

[177] University of Maryland Start. 2009. *Global Terrorism Database*. http://www.start-dev.umd.edu/gtd/

[178] Iain D Stewart, Chris A Kennedy, Angelo Facchini, and Renata Mele. 2018. The electric city as a solution to sustainable urban development. *Journal of Urban Technology* 25, 1 (2018), 3–20.

[179] Arkadiusz Stopczynski, Vedran Sekara, Piotr Sapiezynski, Andrea Cuttone, Mette My Madsen, Jakob Eg Larsen, and Sune Lehmann. 2014. Measuring Large-Scale Social Networks with High Resolution. *PLOS ONE* 9, 4 (2014), 1–24.

[180] Junkai Sun, Junbo Zhang, Qiaofei Li, Xiuwen Yi, Yuxuan Liang, and Yu Zheng. 2020. Predicting citywide crowd flows in irregular regions using multi-view graph convolutional networks. *IEEE Transactions on Knowledge and Data Engineering* (2020).

[181] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*. 3104–3112.

[182] Jinjun Tang, Jian Liang, Tianjian Yu, Yong Xiong, and Guoliang Zeng. 2021. Trip destination prediction based on a deep integration network by fusing multiple features from taxi trajectories. *IET Intelligent Transport Systems* (2021).

[183] Chujie Tian, Xinning Zhu, Zheng Hu, and Jian Ma. 2020. Deep spatial-temporal networks for crowd flows prediction by dilated convolutions and region-shifting attention mechanism. *Applied Intelligence* (2020), 1–14.

[184] New York City TLC. 2009. *New York City Taxi & Limousine Commission*. https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page

[185] Eran Toch, Boaz Lerner, Eyal Ben-Zion, and Irad Ben-Gal. 2019. Analyzing large-scale human mobility data: a survey of machine learning methods and applications. *Knowledge and Information Systems* 58, 3 (2019), 501–523.

[186] Marcello Tomasini, Basim Mahmood, Franco Zambonelli, Angelo Brayner, and Ronaldo Menezes. 2017. On the effect of human mobility to the design of metropolitan mobile opportunistic networks of sensors. *Pervasive and Mobile Computing* 38 (2017), 215 – 232.

[187] Jameson Toole, Carlos Herrera-Yague, Christian Schneider, and Marta C. Gonzalez. 2015. Coupling Human Mobility and Social Ties. *Journal of the Royal Society Interface* 12 (2015).

[188] Alexander Toshev and Christian Szegedy. 2014. DeepPose: Human Pose Estimation via Deep Neural Networks. In *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, 1653–1660.

[189] International Telecommunication Union. 2019. *Measuring digital development Facts and figures*. Technical Report. International Telecommunication Union.

[190] WMO UNISDR. 2012. Disaster risk and resilience. *Thematic Think Piece, UN System Task Force on the Post-2015 UN Development Agenda* (2012).

[191] Michele Vespe, Maurizio Gibin, Alfredo Alessandrini, Fabrizio Natale, Fabio Mazzarella, and Giacomo C. Osio. 2016. Mapping EU fishing activities using ship tracking data. *Journal of Maps* 12, sup1 (2016), 520–525.

[192] Vasiliki Voukelatou, Lorenzo Gabrielli, Ioanna Miliou, Stefano Cresci, Rajesh Sharma, Maurizio Tesconi, and Luca Pappalardo. 2020. Measuring objective and subjective well-being: dimensions and data sources. *International Journal of Data Science and Analytics* (2020).

[193] Di Wang, Tomio Miwa, and Takayuki Morikawa. 2020. Big Trajectory Data Mining: A Survey of Methods, Applications, and Services. *Sensors* 20, 16 (2020), 4571.

[194] Jinzhong Wang, Xiangjie Kong, Feng Xia, and Lijun Sun. 2019. Urban human mobility: Data-driven modeling and prediction. *ACM SIGKDD Explorations Newsletter* (2019), 1–19.

[195] Leye Wang, Xu Geng, Xiaojuan Ma, Feng Liu, and Qiang Yang. 2018. Cross-city transfer learning for deep spatio-temporal prediction. *arXiv preprint arXiv:1802.00386* (2018).

[196] Senzhang Wang, Jiannong Cao, Hao Chen, Hao Peng, and Zhiqiu Huang. 2020. SeqST-GAN: Seq2Seq Generative Adversarial Nets for Multi-Step Urban Crowd Flow Prediction. *ACM Transactions on Spatial Algorithms and Systems (TSAS)* 6, 4 (2020), 1–24.

[197] Senzhang Wang, Jiannong Cao, and Philip Yu. 2020. Deep learning for spatio-temporal data mining: A survey. *IEEE Transactions on Knowledge and Data Engineering* (2020).

[198] Xingrui Wang, Xinyu Liu, Ziteng Lu, and Hanfang Yang. 2021. Large Scale GPS Trajectory Generation Using Map Based on Two Stage GAN. *Journal of Data Science* 19, 1 (2021), 126–141.

[199] Yan Wang and John E Taylor. 2018. Coupling sentiment and human mobility in natural disasters: a Twitter-based study of the 2014 South Napa Earthquake. *Natural hazards* 92, 2 (2018), 907–925.

[200] Yuan Wang, Dongxiang Zhang, Ying Liu, Bo Dai, and Loo Hay Lee. 2019. Enhancing transportation systems via deep learning: A survey. *Transportation research part C: emerging technologies* 99 (2019), 144–163.

[201] Billy M Williams and Lester A Hoel. 2003. Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results. *Journal of transportation engineering* 129, 6 (2003), 664–672.

[202] Jianxin Wu. 2017. Introduction to convolutional neural networks. *National Key Lab for Novel Software Technology. Nanjing University. China* 5 (2017), 23.

[203] Ruizhi Wu, Guangchun Luo, Junming Shao, L. Tian, and Chengzong Peng. 2018. Location prediction on trajectory data: A review. *Big Data Min. Anal.* 1 (2018), 108–127.

[204] Peng Xie, Tianrui Li, Jia Liu, Shengdong Du, Xin Yang, and Junbo Zhang. 2020. Urban flow prediction from spatiotemporal data using machine learning: A survey. *Information Fusion* 59 (2020), 1–12.

[205] SHI Xingjian, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. 2015. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In *Advances in neural information processing systems*. 802–810.

[206] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*. 2048–2057.

[207] Shuai Xu, Xiaoming Fu, Jiuxin Cao, Bo Liu, and Zhixiao Wang. 2020. Survey on user location prediction based on geo-social networking data. *World Wide Web* (2020), 1–44.

[208] Takahiro Yabe, Kota Tsubouchi, and Yoshihide Sekimoto. 2017. CityFlowFragility: Measuring the Fragility of People Flow in Cities to Disasters Using GPS Data Collected from Smartphones. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 1, 3, Article 117 (Sept. 2017), 17 pages. https://doi.org/10.1145/3130982

[209] Takahiro Yabe, Kota Tsubouchi, Akihito Sudo, and Yoshihide Sekimoto. 2016. A Framework for Evacuation Hotspot Detection after Large Scale Disasters Using Location Data from Smartphones: Case Study of Kumamoto Earthquake. In *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems* (Burlingame, California) *(SIGSPACIAL '16)*. Association for Computing Machinery, New York, NY, USA, Article 44, 10 pages. https://doi.org/10.1145/2996913.2997014

[210] Zhaojin Yan, Yijia Xiao, Liang Cheng, Rong He, Xiaoguang Ruan, Xiao Zhou, Manchun Li, and Ran Bin. 2020. Exploring AIS data for intelligent maritime routes extraction. *Applied Ocean Research* 101 (2020), 102271.

[211] Bing Yang, Yan Kang, Hao Li, Yachuan Zhang, Yan Yang, and Lan Zhang. 2020. Spatio-temporal expand-and-squeeze networks for crowd flow prediction in metropolis. *IET Intelligent Transport Systems* 14, 5 (2020), 313–322.

[212] Dingqi Yang, Benjamin Fankhauser, Paolo Rosso, and Philippe Cudre-Mauroux. 2020. Location Prediction over Sparse User Mobility Traces Using RNNs: Flashback in Hidden States!. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*. 2184–2190.

[213] Dingqi Yang, Bingqing Qu, Jie Yang, and Philippe Cudre-Mauroux. 2019. Revisiting User Mobility and Social Relationships in LBSNs: A Hypergraph Embedding Approach. 2147–2157.

[214] Di Yao, Chao Zhang, Jianhui Huang, and Jingping Bi. 2017. Serm: A recurrent model for next location prediction in semantic trajectories. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. 2411–2414.

[215] Huaxiu Yao, Xianfeng Tang, Hua Wei, Guanjie Zheng, and Zhenhui Li. 2019. Revisiting spatial-temporal similarity: A deep learning framework for traffic prediction. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 33. 5668–5675.

[216] Huaxiu Yao, Xianfeng Tang, Hua Wei, Guanjie Zheng, Yanwei Yu, and Zhenhui Li. 2018. Modeling spatial-temporal dynamics for traffic prediction. *arXiv preprint arXiv:1803.01254* (2018).

[217] Xin Yao, Yong Gao, Di Zhu, Ed Manley, Jiaoe Wang, and Yu Liu. 2020. Spatial Origin-Destination Flow Imputation Using Graph Convolutional Networks. *IEEE Transactions on Intelligent Transportation Systems* (2020).

[218] Dan Yin and Qing Yang. 2018. GANs based density distribution privacy-preservation on mobility data. *Security and Communication Networks* 2018 (2018).

[219] Xueyan Yin, Genze Wu, Jinze Wei, Yanming Shen, Heng Qi, and Baocai Yin. 2020. A Comprehensive Survey on Traffic Prediction. *arXiv preprint arXiv:2004.08555* (2020).

[220] Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. 2017. Seqgan: Sequence generative adversarial nets with policy gradient. In *Thirty-first AAAI conference on artificial intelligence*.

[221] Hao Yuan, Xinning Zhu, Zheng Hu, and Chunhong Zhang. 2020. Deep Multi-View Residual Attention Network for Crowd Flows Prediction. *Neurocomputing* (2020).

[222] Chao Zhang, Jiawei Han, Lidan Shou, Jiajun Lu, and Thomas LLa Porta. 2014. Splitter: Mining fine-grained sequential patterns in semantic trajectories. *Proceedings of the VLDb Endowment* 7, 9 (2014), 769–780.

[223] Chao Zhang, Keyang Zhang, Quan Yuan, Luming Zhang, Tim Hanratty, and Jiawei Han. 2016. Gmove: Group-level mobility modeling using geo-tagged social media. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1305–1314.

[224] Junbo Zhang, Yu Zheng, and Dekang Qi. 2017. Deep spatio-temporal residual networks for citywide crowd flows prediction. In *Thirty-First AAAI Conference on Artificial Intelligence*.

[225] Yahua Zhang, Anming Zhang, and Jiaoe Wang. 2020. Exploring the roles of high-speed train, air and coach services in the spread of COVID-19 in China. *Transport Policy* 94 (2020), 34 – 42. https://doi.org/10.1016/j.tranpol.2020.05.012

[226] K. Zhao, S. Tarkoma, S. Liu, and H. Vo. 2016. Urban human mobility data mining: An overview. In *2016 IEEE International Conference on Big Data (Big Data)*. 1911–1920.

[227] Liang Zhao. 2020. Event Prediction in Big Data Era: A Systematic Survey. *arXiv preprint arXiv:2007.09815* (2020).

[228] Xin Zheng, Jialong Han, and Aixin Sun. 2018. A survey of location prediction on twitter. *IEEE Transactions on Knowledge and Data Engineering* 30, 9 (2018), 1652–1671.

[229] Yu Zheng. 2011. T-Drive trajectory data sample.

[230] Yu Zheng. 2015. Trajectory data mining: an overview. *ACM Transactions on Intelligent Systems and Technology (TIST)* (2015), 1–41.

[231] Yu Zheng, Licia Capra, Ouri Wolfson, and Hai Yang. 2014. Urban computing: concepts, methodologies, and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)* 5, 3 (2014), 1–55.

[232] Yu Zheng, Xing Xie, Wei-Ying Ma, et al. 2010. GeoLife: A collaborative social networking service among user, location and trajectory. *IEEE Data Eng. Bull.* 33, 2 (2010), 32–39.

[233] Yirong Zhou, Hao Chen, Jun Li, Ye Wu, Jiangjiang Wu, and Luo Chen. 2019. ST-Attn: Spatial-Temporal Attention Mechanism for Multi-step Citywide Crowd Flow Prediction. In *2019 International Conference on Data Mining Workshops (ICDMW)*. IEEE, 609–614.

[234] Yirong Zhou, Ye Wu, Jiangjiang Wu, Luo Chen, and Jun Li. 2018. Refined Taxi Demand Prediction with ST-Vec. In *2018 26th International Conference on Geoinformatics*. IEEE, 1–6.

[235] Wen-Yuan Zhu, Wen-Chih Peng, Ling-Jyh Chen, Kai Zheng, and Xiaofang Zhou. 2015. Modeling User Mobility for Location Promotion in Location-Based Social Networks. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1573–1582.

[236] George Kingsley Zipf. 1946. The P 1 P 2/D hypothesis: on the intercity movement of persons. *American sociological review* 11, 6 (1946), 677–686.

[237] Ali Zonoozi, Jung-jae Kim, Xiao-Li Li, and Gao Cong. 2018. Periodic-CRN: A Convolutional Recurrent Model for Crowd Density Prediction with Recurring Periodic Patterns.. In *IJCAI*. 3732–3738.

A SPATIAL TESSELLATIONS

Spatial tessellations may be computed using several open-source tools. Figure 6 shows examples of tessellations in New York City.

Library scikit-mobility [135] allows for creating a squared spatial tessellation, given a reference geographic area and a tile size expressed in meters.

S2 Geometry is an open-source project that represents spatial data on a three-dimensional sphere. It provides efficient and scalable spatial indexing techniques to carry out operations such as testing relationships among objects, measuring centroids, distances, and more. S2 Geometry decomposes the unit sphere into a hierarchy of cells (tiles), each of which is a quadrilateral bounded by four geodesics. The top level of the hierarchy is obtained by projecting the six faces of a cube into the unit sphere, and lower levels are obtained by subdividing each cell into four sub-cells recursively. Each cell in the hierarchy has a level, defined as the number of times the cell has been subdivided (starting with a face cell). Cells' levels range from 0 to 30. The smallest cells at level 30 are called leaf cells; there are 6×4^{30} cells in total, each about 1cm across on the Earth's surface.

The H3 geospatial indexing system consists of a hexagonal tiling of the sphere with hierarchical indexes. The hexagonal grid system is created on the planar faces of a sphere-circumscribed icosahedron, and the grid cells are then projected to the surface of the sphere using a specific projection. The H3 grid is constructed by recursively creating increasingly higher precision hexagon grids until the desired resolution is achieved. The first H3 resolution (resolution 0) consists of 122 base cells, and each subsequent resolution is created splitting each cell into seven children recursively. H3 provides 15 finer grid resolutions in addition to resolution 0. The finest resolution, resolution 15, has cells with an area of less than $1m^2$.
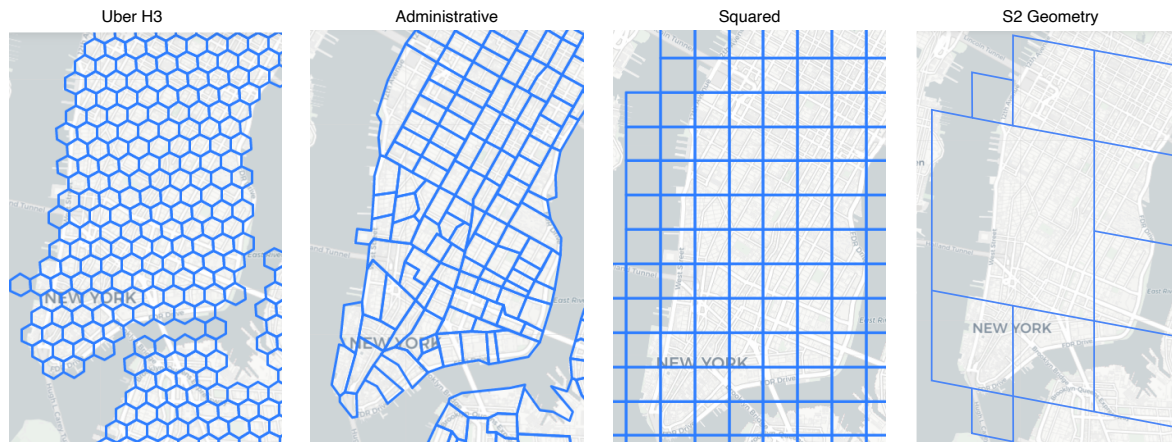

Fig. 6. Examples of spatial tessellations constructed over New York City. On the left, the city is tessellated in hexagons using H3 (resolution 6). In the second image, the area is split according to the administrative boundaries described in the GeoJSON file downloaded from New York City's open data portal. In the third image, we use scikit-mobility to split an area into a squared grid with tiles of 1km×1km. In the last image, we make a tessellation using S2 Geometry.

B DEEP LEARNING MODULES

B.1 Recurrent Neural Networks

A fully connected neural network (FC) consists of a series of fully-connected layers. All the neurons in one layer are connected to the neurons in the next layer. An FC layer, therefore, is a function that, given an input $x \in \mathcal{R}^m$,

map values in $\mathcal{R}^m \to \mathcal{R}^n$ where m and n are the number of neurons in two consecutive layers. FCs are universal approximators (i.e., can learn any representation function) [66] but they are not able to deal with sequential data.

Recurrent Neural Networks (RNNs) [157] can efficiently deal with sequential data such as time series, in which values are ordered by time, or sentences in natural language, in which the order of the words is crucial to shaping its meaning. An RNN consists of a sequence of gates $G = \{G_0, \ldots, G_{n-1}\}$, each one producing an hidden state h_i based on the current input x_i and the output from the previous gate h_{i-1} (Figure 7). In Vanilla RNNs, a gate G_i is implemented by using a hyperbolic tangent function (tanh), which takes as input x_i and h_{i-1} and computes the current state h_i. RNNs suffer from the vanishing gradient problem [98] and cannot propagate information found at early steps, losing relevant information at the beginning of a sequence when it is time to analyze its end [98]. Long-short-term Memory networks (LSTMs) [75] and Gated Recurrent Units (GRUs) [35] are two gate implementations that mitigate this problem (Figure 7).
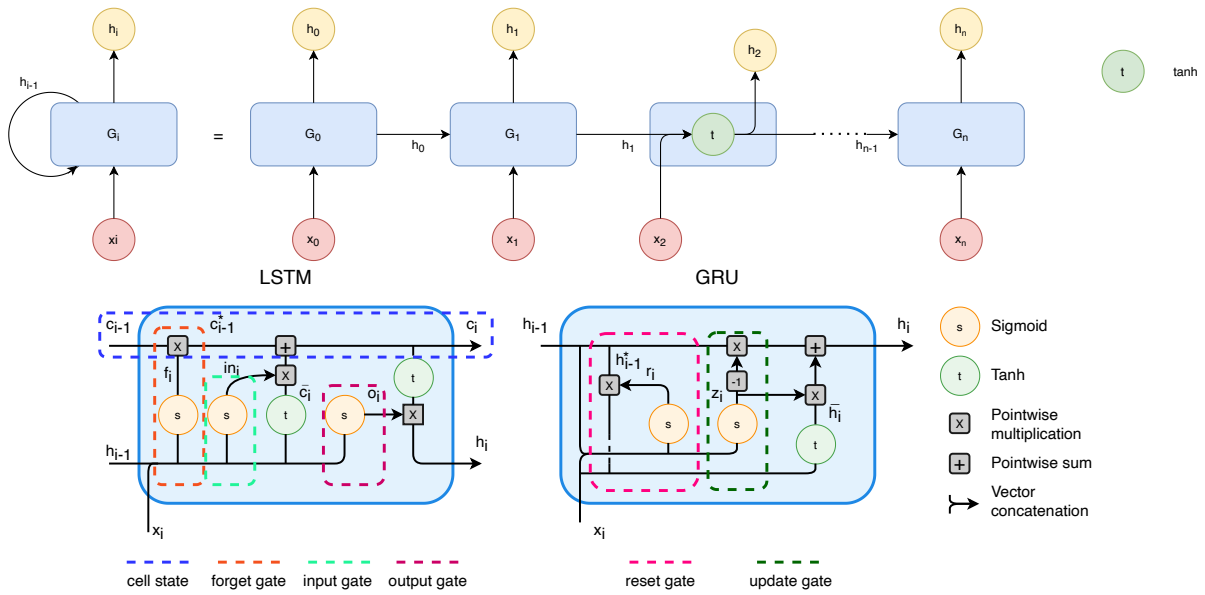

Fig. 7. (Top) An example of a Recurrent Neural Network (RNN). The right term of the equation corresponds to the unrolled version of the network on the left term. In general, an RNN takes as input a sequence $X = \langle x_0, x_1, \ldots x_n \rangle$ and produces an output h_i that, at a certain moment i, is based on the current input x_i and the output of the previous gate h_{i-1}. Usually, in Vanilla RNNs, the tanh function (green circle) is used to combine the current input x_i and the previous hidden state h_{i-1}. (Bottom) The structure of an LSTM gate (left) and a GRU gate (right). In the LSTM gate, we highlight with a dashed line the cell state (in blue), the forget gate (in orange), the input gate (in green), and the output gate (in purple). In the GRU gate, we highlight with the dashed line the reset gate (in red) and the update gate (in green).

In LSTMs, the cell state c_i carries the data through the network, while the internal gates remove useless information from the flow or add relevant knowledge to the cell state. There are three internal gates: forget, input, and output gate (Figure 7). The forget gate passes the information of the previous hidden state h_{i-1} and of the current input x_i to a sigmoid activation function which ranges in $[0, 1]$. The closer the sigmoid's output (f_i) is to 0, the less likely the information is to be considered relevant. The sigmoid's output f_i is then pointwise multiplied with previous cell state c_{i-1} forming c_{i-1}^*. Values of c_{i-1}^* close to 0 are not taken into consideration by the current cell state. The input gate passes h_{i-1} and x_i to a sigmoid activation function and to a tanh activation

function. The sigmoid function outputs in_i, i.e., the relevance of the new data (0 is not relevant; 1 is relevant). The tanh outputs, $\overline{c}_i \in [-1, 1]$, is multiplied with in_i, and the resulting product is pointwise summed with c^*_{i-1} to obtain the new cell state c_i. The output gate generates the hidden state h_i. The previous hidden state h_{i-1} and the current input x_i are passed to a sigmoid activation function, which generates o_i. This value is pointwise multiplied with the output of a tanh applied on the new cell state c_i, to form the new hidden state h_i. The hidden state h_i and the cell state c_i are passed to the next gate of the recurrent network. In some works, bidirectional LSTMs (Bi-LSTMs) [161] are adopted instead of LSTMs. Bi-LSTMs duplicate the recurrent layer: the sequence is provided as input to one layer and its reverse as input to the other layer. Analyzing time dependencies in both directions gives an advantage in scenarios like speech recognition, in which the context is essential to interpret the meaning of a sentence [68, 69].

In GRUs, the relevant information is propagated throughout the network using hidden states only (Figure 7). GRUs have two types of internal gates. The reset gate decides how much to forget and consists of a sigmoid function that takes as input the hidden layer of the previous step h_{i-1} and the current input x_i. The output of the sigmoid, r_i, is pairwise multiplied with h_{i-1}, generating h^*_{i-1}. Depending on h^*_{i-1}, we can determine which past information to forget (low values) or keep (high values). The update gate establishes whether the past information is relevant for future predictions and, therefore, should be propagated to the next steps. It is composed of a sigmoid function that takes as input h_{i-1} and x_i and outputs z_i. The next hidden state, h_i, is obtained by: *(i)* computing $\hat{h}_i$, which is the output of a tanh function of h^*_{i-1} and the current input x_i; *(ii)* multiplying $1 - z_i$ with h_{i-1}; and *(iii)* adding the resulting sum to the product between z_i and $\hat{h}_i$.

Further mathematical details and formalizations can be found in [163] for RNNs and LSTMs and [36] for GRUs.

RNNs are widely used in next-location prediction (Section 3.1), often in combination with attention mechanisms, to capture the temporal relationships in individual trajectories. Moreover, RNNs are often combined with convolutional neural networks in crowd flow prediction (Section 3.2) and trajectory generation (Section 4.1), to capture temporal and spatial patterns at the same time.

B.2 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are widely used in computer vision for their efficacy in object recognition [147, 168], image classification and segmentation [53, 103], movement or event recognition [188], and more [95]. Similarly to the visual cortex [78, 79], a CNN is a network made of neurons that react only to certain stimuli in a restricted region of the visual field [103].

CNNs alternate two types of layers: *(i)* the convolutional layers reduce the size of the matrix by applying a kernel function, or filter, that keeps all the relevant information; *(ii)* the pooling layers reduce the spatial size of the convoluted features to decrease the computational power required to process the data. Figure 8 shows an example of a CNN architecture with two convolutional layers and two pooling layers. Usually, at the end of the CNN, an FC is used to compute the output.

The convolutional layers apply one or more filters to the input matrix A to extract relevant features and summarize the characteristics of an $i \times j$ area of A into a single value. Specifically, given an input matrix A of size $n \times m$, a filter is a mathematical operation between the original matrix A and another matrix B of size $k \times l$, with $k < n$ and $l < m$. The filter is generally applied with a sliding window mechanism called stride.

The pooling layers aim at downsampling the input matrix, replacing each portion of it with summary statistics. In practice, pooling layers are either max pooling or average pooling. Max pooling returns the maximum value from the part of the matrix convoluted by a given size and stride filter. Average pooling returns the average of the values produced by the filter operations. Pooling layers create representations of the matrix that are approximately invariant to small translations of the input.
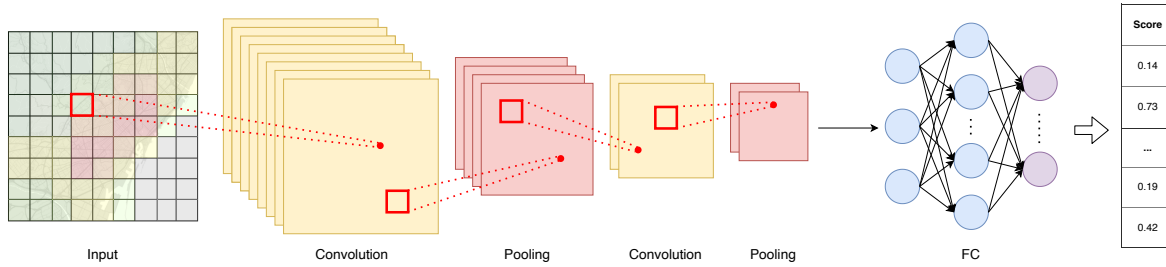
Fig. 8. Example of architecture of a CNN. The neural network consists of two convolutions and two pooling layers. After the convolution/pooling layers, there is an FC that outputs the prediction. For instance, to classify a sample into one of n classes, the last layer is usually fed into an FC that outputs a $1 \times n$ vector containing the probabilities of the sample to belong to a specific class.

A significant limitation of CNNs, especially relevant for networks with many layers, is the vanishing gradient issue [98]. The so-called residual units are a solution to this problem [73]. A residual unit implements the skip connection also known as identity connection. Given a network with $l_1, \ldots, l_k$ layers, the output of a layer l_i is added to the output of the layer l_{i+j} so to preserve the loss of information. The variable j represents the skip size, and it is usually smaller than 4.

A mathematical formalization of CNNs can be found in [202].

Differently from RNNs, where the input is processed sequentially, CNNs are not designed to process sequences, and they are used mainly in crowd flow prediction (Section 3.2): the evolution of the incoming (outgoing) flows within a city is represented as a sequence of matrices. CNNs are used to capture the dynamics of the spatial dependencies among the areas of a city. CNNs are often combined with RNNs to capture temporal dependencies (e.g., ConvLSTM [205]). CNNs are also used in trajectory generation (Section 4.1): real trajectories are represented as images and used to train a Generative Adversarial Network (GAN), in which two CNNs are used as generator and discriminator to generate realistic trajectories.

B.3 Attention Mechanism

Attention mechanisms are based on the idea that, when dealing with a large amount of information, our brains focus on the most significant parts and consider all the others as background information. Initially introduced for natural language processing (e.g., machine translation [36, 181] and speech recognition tasks [38]), the usage of attention mechanisms rapidly extended to computer vision, healthcare, and recommendation systems [37, 162, 206]. Given the current input x and a context, an attention mechanism produces a score for each element of x. These scores are usually computed using adequate activation functions (e.g., softmax) and organized in a vector s. A context vector $\hat{c}$ is computed as the pairwise multiplication between s and x. The context plays a crucial role in the establishment of which features should be the most important to the model.

B.4 Generative Models

Dimensionality reduction consists in transforming data from a high-dimensional space to a low-dimensional one retaining important properties and information of the original data [102]. In general, this is done by employing a module called *encoder* (i.e., a neural network), which reduces the space of the original features generating a *latent space*; and a *decoder* (i.e., a neural network), which transforms the latent space back to the original space. AutoEncoders (AEs) [102] reduce the reconstruction error while decoding. AEs are designed to encode a sample always in the same way and, therefore, cannot be considered a generative model as they are not
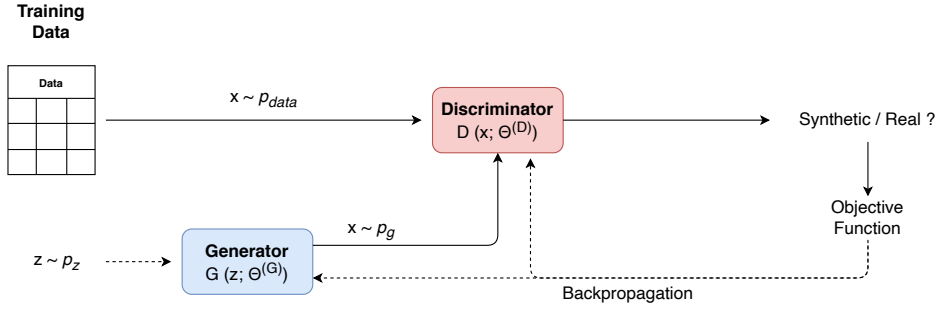
Fig. 9. Visual representation of a Generative Adversarial Network (GAN). A GAN is composed of a generator G, and a discriminator D. The generator is a differentiable function $G(z; \Theta^{(G)})$ which outputs the new data according to a distribution p_g, where $\Theta^{(G)}$ are the parameters of the generative model. The discriminator represents a differentiable function $D(x; \Theta^{(D)})$, where $\Theta^{(D)}$ are the parameters of the discriminative model, which produces the probability that x comes from the distribution of training data p_{data}. The aim is to obtain a generator that is a good estimator of p_{data}. When this occurs, the discriminator is "fooled" and can no longer distinguish the samples from p_{data} from those from p_g.

capable of decoding (i.e., generate) data with different charateristics with repsect to the original data. Variational AutoEncoders (VAEs) [96] solve this problem by encoding a sample as a distribution instead of a set of fixed values. In this way, the decoder can be used to generate data that are similar to the original ones (e.g., realistic as they should follow similar patterns) but diverse.

Generative adversarial networks (GANs) [67] are an example of generative models, i.e., any model that takes a training set, consisting of samples drawn from a distribution, and learns to represent an estimate of that distribution. A generative mobility model can generate synthetic spatio-temporal trajectories that realistically reproduce mobility patterns. Recently, GANs are being used to generate synthetic mobility trajectories [115, 128].

The basic idea of a GAN is to set up a game between a generator (e.g., a neural network) and a discriminator (e.g. a classifier) (Figure 9). The generator creates samples from the distribution p_g that are intended to come from the same distribution as the training data p_{data} and hence similar to the original ones. The discriminator examines the generated samples to determine whether they are real or fake. In other words, the generator is trained to fool the discriminator by generating samples that are indistinguishable from real ones.

Mathematically, the generator and the discriminator are two functions that are differentiable with respect to both the inputs and the parameters. The discriminator is a function D that takes x (e.g., a trajectory) as input and uses $\Theta^{(D)}$ as parameters. The generator is a function G that takes z (e.g., set of trajectories) as input and uses $\Theta^{(G)}$ as parameters. D wishes to minimize a cost function $J(D)(\Theta^{(D)}, \Theta^{(G)})$ while controlling only $\Theta^{(D)}$. Similarly, G wishes to minimize a cost function $J(G)(\Theta^{(D)}, \Theta^{(G)})$ while controlling only $\Theta^{(G)}$. This scenario is a game the solution of which is a Nash equilibrium [66], i.e., a tuple $(\Theta^{(D)}, \Theta^{(G)})$ that is a local minimum of $J(D)$ with respect to $\Theta^{(D)}$ and a local minimum of $J(G)$ with respect to $\Theta^{(G)}$.

The training process consists of two simultaneous Stochastic Gradient Descent (SGD) [66]. SGD is a stochastic approximation of gradient descent optimization that replaces the actual gradient with an estimate, thus achieving faster iterations in trade for a lower convergence rate. Two gradient steps are made simultaneously: one updating $\Theta^{(D)}$ to reduce $J(D)$ and one updating $\Theta^{(G)}$ to reduce $J(G)$. Usually, the cost function used for the discriminator and the generator is the cross-entropy [66], defined as: $H(P, Q) = -E_P[logQ]$, where P, Q are two distributions and E_P is the expected value of $logQ$ according to P.

A formalization of the concept of GAN can be found in [67]

C DATA

The last decade has witnessed the emergence of massive datasets of digital traces that portray human movements at an unprecedented scale and detail. Examples include tracks generated by GPS devices embedded in smartphones [232] or private vehicles [133]; mobile phone records [18]; and geotagged posts from social media platforms [125]. Unfortunately, most of these datasets are proprietary and not publicly available, making research on human mobility hard to reproduce. In this Section, we discuss the peculiarities of various mobility data sources and, for each of them, provide a reference to a list of public datasets commonly used to train and test models presented in this survey (Table 5).

C.1 Mobile Phone Records

Mobile phones are ubiquitous, with coverage in most countries that reaches almost 100% of the population [18, 189]. Telco companies record the activity of mobile phone users for billing and operational purposes, hence storing an enormous amount of information on where, when, and with whom users communicate [18]. Every time a user engages a telecommunication interaction – calls, text messages, data connections – the operator assigns a Radio Base Station (RBS) to deliver the communication through the network. Since the position and the coverage area of each RBS are known, a user's telecommunication interaction reflects their geographic location. Each interaction generates several mobile data formats [18, 131].

Every time a user makes/receives a call or sends/receives an SMS, a new Call Detail Record (CDR) is created. A CDR is a tuple $(u_o, u_i, t, A_o, A_i, d)$, where u_o and u_i are the identifiers of the caller and the callee, and t is a timestamp of when the call starts. In turn, A_o and A_i are the RBSs that manage the outgoing call and the receiving call, and d is the call duration. An individual's mobility can be reconstructed from CDRs assuming a movement between the RBSs of any two consecutive records [18]. Aggregated mobility, such as flows, can be inferred by counting the number of users that move, in a given time window, between two RBSs or spatial aggregations of them (e.g., neighborhoods or municipalities) [24]. CDRs are sparse in time, i.e., a user's position is known when they make or receive a call or a text message only, leading to sparse and incomplete mobility trajectories. Notwithstanding, they are the most common format in human mobility studies [10, 18, 40, 65, 136, 167].

When a user uploads or downloads data from the Internet using their phone's connection, they generate an eXtended Detail Record (XDR), a tuple (u, t, A, k) where A is the RBS that serves the connection, and k the amount of uploaded/downloaded information. XDRs are less common in the literature than CDRs, mainly because the advent of the mobile data connection is relatively recent. XDRs partially overcome the problem of sparsity present in CDRs [30], because mobile connections are more frequent than calls and text messages.

For both CDRs and XDRs, the spatial granularity is at the level of RBS, i.e., the user position is approximated with the location of the RBS used for the telecommunication activity. This approximation implies that the user's position within an RBS's coverage area is unknown and that user tracking depends on the spatial distribution of antennas on the territory, which depends on population density. Nevertheless, mobile phone data may cover a large sample size on a national scale. An advantage of mobile phone data is its multidimensionality: CDRs also provide information about the social interactions between users; both CDRs and XDRs may be accompanied by socio-demographic information about the users (e.g., age, sex).

C.1.1 Available datasets. Since mobile phone data contain sensitive information [45, 139], they are not typically publicly available, and any data collected by a specific group of researchers may not be shared with other groups, making reproducibility difficult.

C.2 GPS traces

Global Navigation Satellite Systems (GNSS) use satellites to provide geo-spatial positioning, allowing electronic receivers to determine their location (longitude, latitude, and altitude) and time, using signals transmitted along a

line of sight from satellites. The US Global Positioning System (GPS) is the most popular one, and GPS receivers are ubiquitous in many tools of everyday life, such as mobile phones [5, 179, 232], vehicles [61, 133, 136], vessels [158, 170, 191], and wearable devices [154]. On mobile phones, GPS receivers are activated by apps that require the user's position (e.g., Google Maps). On vehicles, GPS devices automatically turn on when the vehicle starts, sending positions to a server with a frequency of a few seconds. The precision of GPS receivers varies from a few centimeters to meters, depending on the device's quality and the errors generated by the system [27]. GPS cannot track the devices in enclosed spaces, such as buildings and tunnels. A typical GPS trace is a set of tuples (u, t, lat, lng) where u is a user, t is a timestamp of the measurement and lat, lng are the position's latitude and longitude. GPS traces may require several preprocessing tasks aimed to mitigate errors and extract meaningful semantics. For example, since GPS traces are dense sequences of spatio-temporal points, they do not explicitly define semantic locations, which must be inferred through specific preprocessing techniques [56, 230, 231].

C.2.1 Available datasets. GeoLife [232] is a publicly available dataset describing the GPS trajectories of 182 users over 4.5 years, collected using different devices (e.g., GPS receivers and mobile phones) every 1-5 seconds or 5-10 meters. GeoLife covers a broad range of users' outdoor movements, from life routines like go home and go to work to entertainment and sports activities. Each point in a GeoLife's trajectory contains latitude, longitude, altitude, and the timestamp. Moreover, the user's transportation mode is also available for most of the trajectories.

Other public datasets provide information about the trips of GPS-equipped taxis in several cities. Piorkowski et al. [143] provide the trajectories, sampled on average every 10 seconds, of taxis in San Francisco, in May 2008. Each point of a trajectory consists of the taxi's identifier, the latitude, the longitude, the timestamp, and the occupancy. Moreira et al. [123] (ECML/PKDD Challenge) provide the trips of taxis in Porto, Portugal, in which points contain the latitude, the longitude and a timestamp indicating when the trip started. Data are collected approximately every 15 seconds. For each trip, the dataset provides several auxiliary information, such as the trip's typology (e.g., dispatched from the central, demanded to the operator, requested to the driver), the stand from which the taxi departed, and an identifier of the passenger's phone number. The Taxi and Limousine Commission of New York City collected a dataset on yellow and green taxis operating in the city starting from 2009 [184]. The dataset provides information on pick-up and drop-off dates/times and locations, trip distances, itemized fares, rate types, payment types, and driver-reported passenger counts.

The T-Drive dataset [229] describes the trajectories of about 10,000 taxis in Beijing, China, for one week. Points are sampled every 177 seconds and contain the taxi's identifier, the latitude, the longitude, and the timestamp. Zhang et al. [224] create a squared tessellation on New York City and Beijing and, for each tile, provide the incoming and the outgoing flows. Beijing's flows are captured based on taxis' GPS signal; the New York City ones are based on the city's bike-sharing system.

Similarly, Jiang et al. [82] generated a tessellation of 450×450 meters over Tokyo and Osaka, Japan and, for each tile, they provide people density and incoming and outgoing flows. The mobility is captured using GPS devices.

Citi Bike by Lyft [14] provides a dataset describing the trips between bike-sharing stations in New York City. Each record describes the stations where the trip started and ended when it took place and the stations' coordinates. There are similar open datasets for other cities, such as Washington DC [15].

The Mobile Data Challenge dataset (MDC) [106] describes the trajectories of 185 participants of a data collection campaign in Lausanne, collected from mobile phones [97]. The dataset offers various information (e.g., calendar logs, data from accelerometers), including two files describing the individuals' mobility through GPS receivers and Wireless Local Area Networks (WLANs). Each record in the first file describes the user's identifier, latitude, longitude, altitude, timestamp, speed, heading direction, and accuracy. Records in the second file describe the user's identifier, a timestamp, the first three bytes of the device's MAC address, and the access points' location.

Recently, to take countermeasures for the COVID-19 pandemic, companies such as Cuebiq and SafeGraph are providing free access upon request to their data.[1]

C.3 Social Media Data

Users' posts on social media (e.g., photos, text, videos) can be associated with a geographic location and time (geotagged). The presence of spatial and temporal information allows the reconstruction of users' trajectories from the sequence of published posts. Some platforms like Twitter provide either the post's precise geotag (i.e., a latitude and longitude pair, a format recently removed) or the position of a predefined location suggested by the platform (e.g., a city, an area, a restaurant). In other platforms like Foursquare and Facebook, users can check-in in predefined locations called venues, i.e., POIs that provide information about social, cultural, and infrastructural components of a geographic area (e.g., cities, shops, museums). A venue is associated with a physical location (latitude and longitude pair) and textual information (a description of the place or the activities related to the place) and can follow a hierarchical categorization that provides different levels of detail about the activities (e.g., Food, Asian Restaurant, Chinese Restaurant) [12].

In general, a geotagged record describes the posting user's identifier, the resource identifier (e.g., post, tweet, photo), the time of posting, and, depending on the platform, the venue identifier/category or a location as a string (e.g., Statue of Liberty, New York) or a latitude/longitude pair.

For most of the social media platforms, geotagged posts are downloadable through their APIs. APIs impose limitations on the number of downloadable posts and queries per day or require authorization from the platform's users to download the data. Users' location is available only when they post something or check-in into a venue, leading to a data sparsity problem. Nevertheless, social media data bring the advantage that an objective definition of location is available, facilitating the data preprocessing phase [41].

C.3.1 Available datasets. Datasets about check-ins on social media platforms that are not active anymore, such as Gowalla and Brightkite, are freely available. Gowalla was a location-based social network platform in which, similarly to Foursquare, the users were allowed to check-in in the so-called spots (venues) through a website or the app. The related dataset [34] consists of more than six million check-ins over one year and a half from February 2009 to October 2010. Each check-in describes the user identifier, the location identifier, the latitude and longitude pair, and a timestamp. The dataset also provides information about the users' friendship network, which contains about 200,000 nodes and one million edges [34]. In Brightkite, another platform that is not active anymore, users were allowed to check-in in POIs, to specify who is nearby at the moment and who went to a POI before. The dataset contains almost 4.5 million check-ins from April 2008 to October 2010 and the users' friendship network with about 60,000 nodes and 220,000 edges [34].

A dataset collected through Foursquare APIs is introduced in [54] and contains check-ins of 16,000 users over one year in New York City.

Twitter provides several open datasets, in which location is usually expressed as a semantic point of interest either suggested by the platform (e.g., Empire State Building, NYC) or typed by the users (e.g., Home), or as a latitude and longitude pair. Geotagged tweets may be retrieved directly using Twitter APIs. For example, Zhang et al. [223] (GMove) provide a Twitter dataset describing 1.4 million tweets from August to November 2014 covering the area of Los Angeles.

A comprehensive list of Twitter datasets is available at github.com/shaypal5/awesome-twitter-data.

---

[1]Cuebiq: cuebiq.com/visitation-insights-covid19/, SafeGraph: safegraph.com/covid-19-data-consortium. For instance, [91] is a recently published dataset based on SafeGraph. It contains aggregated daily mobility flows at different spatial aggregations such as country to country, state to state and census tract to census tract flows.

	Ref.	Name	Items	Time span	Area	Used By	Task	Link (https://bit.ly/)
GPS traces	[232]	GeoLife	182	4.5 Years	Asia	[55]	Traj. Gen.	Geolife
	[229]	T-Drive	10k	1 Week	Beijing, China	[80]	Flow. Pred.	T-Drive-Data
	[82]	DeepCrowd	-	4 months	Tokyo and Osaka, Japan	[82]	Crowd Flow Pred.	DeepCrowd
	[224]	ST-ResNet taxis	-	4, 6 months	Beijing, China	[47, 87, 109, 113, 148, 180, 221, 224, 237]	Crowd Flow Pred.	ST-ResNet
	[224]	ST-ResNet bikes	-	6 months	New York City, USA	[47, 87, 109, 113, 148, 180, 221, 224, 237]	Crowd Flow Pred.	ST-ResNet
	[143]	Taxi San Francisco	500	30 days	San Francisco, USA	[49, 153, 218]	Next-Loc., Traj. Gen.	TaxiSF
	[123]	ECML-PKDD taxi	441	9 months	Porto, Portugal	[44, 49, 118, 153]	Next-Loc.	TaxiPorto
	[184]	Taxi New York City	-	From 2009	New York City	[153, 180, 183]	Next-Loc.	TaxiNYC-2
	[106]	MDC	185	2 years	Lausanne, Switzerland	[104, 129]	Traj. Gen	MDC-2
	[91]	COVID 19 US Flows	-	From 2019	United States	[166]	Flow Gen.	USFlows
check-ins	[34]	Gowalla	196k	20 months	California & Nevada, USA	[64, 114, 212]	Next-Loc.	GowallaData
	[34]	Brightkite	58k	30 months	-	[64, 114, 212]	Next-Loc.	Brightkite
	[54]	DeepMove	16k	1 Year	New York City	[54]	Next-Loc.	DeepMove
	[223]	GMove	1.4M	4 Months	Los Angeles	[214]	Next-Loc.	SERM-Repo
	[14]	New York City bikes	-	from 2013	New York City, USA	[112, 180, 183, 216]	Crowd Flow Pred.	BikeNYCData
	[15]	Washington DC bikes	-	from 2010	Washington DC , USA	[180]	Crowd Flow Pred.	BikeWashington

Table 5. Public mobility datasets used in the selected papers. For each dataset, we provide a reference to the paper introducing it, the number of items (users or points) in the dataset (symbol "-" indicates that the dataset is aggregated, that the number is not available, or that the dataset is continuously updated), its time span, the geographic area covered, the list of selected papers that use it, the mobility tasks the dataset is used for, and the link to download it.

D   EVALUATION METRICS

D.1   Distance metrics

The *Haversine distance* is the distance on the spherical earth[2] between two points p_1 and p_2:

$$d_h(p_1, p_2) = 2R\left(\sqrt{\frac{a(p_1, p_2)}{a(p_1, p_2) - 1}}\right); \quad a(p_1, p_2) = sin^2\left(\frac{\phi_2 - \phi_1}{2}\right) + cos(\phi_1)cos(\phi_2)sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right) \tag{3}$$

where R is the earth radius and λ_i and ϕ_i, with $i = 1, 2$, are the longitude and the latitude of p_i, respectively. The Haversine distance range in $[0, \infty]$ and lower values indicate better performance. The *equirectangular distance* is defined as:

$$d_{eq}(p_1, p_2) = R\sqrt{\left(\left(\lambda_{p_2} - \lambda_{p_1}\right)cos\left(\frac{\phi_{p_2} - \phi_{p_1}}{2}\right)\right)^2 + \left(\phi_{p_2} - \phi_{p_1}\right)^2} \tag{4}$$

where λ_{p_i} and ϕ_{p_i} are the longitude and latitude of point p_i, respectively.

D.2   Classification metrics

Accuracy (ACC) indicates how many of the locations an individual will visit are correctly predicted. The k-accuracy (ACC@k) is often used instead of ACC: predictors output a list of all possible locations an individual can visit next, ranked from the most to the least likely, and ACC@k is the fraction of times the real location

---

[2]Flat earthers can simply use the Euclidean distance, defined as follows: given two points $p_1 = (x_1, y_1)$ and $p_2 = (x_2, y_2)$, $d_{flatearth} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$

is among the k most likely locations predicted by the model, i.e., the percentage that a list of predictions with length k covers the ground truth location (ACC = ACC@1).

Precision measures how accurate the predictor is on the positive class. Recall measures the True Positive Rate (TPR), i.e., the fraction of positive instances correctly predicted by the model. F1-score summarizes the performance of a model and it is computed as the harmonic mean of Precision and Recall. Given the number of True Positives (TPs), False Positives (FPs), and False Negatives (FNs), we define Precision, Recall, and F1-Score as:

$$Precision = \frac{TP}{(TP + FP)}; \quad Recall = \frac{TP}{(TP + FN)}; \quad F1 = 2\frac{Precision \times Recall}{Precision + Recall} \tag{5}$$

The Receiver Operating Characteristic Curve (ROC) visualizes a classifier's performance by plotting the Recall against the False Positive Rate (FPR). FPR is the ratio of FPs over the sum of FPs and True Negatives (TNs). AUC ($\in [0, 1]$) measures the area under the ROC. It is scale-invariant and provides an aggregate measure of performance across all possible classification thresholds. The higher the AUC, the better the model, where AUC=0.50 indicates the performance of a model that makes predictions at random.

D.3 Error metrics

Commonly used error metrics are Mean Average Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and MAPE (Mean Average Percentage Error), defined as follows:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|; \quad \text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2; \quad \text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}; \quad \text{MAPE} = \left(\frac{1}{n}\sum_{i=1}^{n}\frac{|y_i - \hat{y}_i|}{|y_i|}\right) * 100 \tag{6}$$

where $\hat{y}_i$ indicates the predicted value, y_i indicates the actual value, and n is the number of predictions. All metrics range in $[0, \infty]$ and lower values indicate better performance. Since MAE uses the error's absolute value, it does not consider whether the model overestimates or underestimates the actual value. MSE weighs large errors more than MAE, and it is sensitive to outliers. RMSE weighs the errors more than MAE, hence penalizing models that produce large errors. As the values are squared, the RMSE is expressed in the same unit as the predicted one.

The Sørensen-Dice index, also called Common Part of Commuters (CPC) [9, 108]. It is a well-established measure to compute the similarity between real flows, y^r, and generated flows, y^g:

$$CPC = \frac{2\sum_{i,j} min(y^g(l_i, l_j), y^r(l_i, l_j))}{\sum_{i,j} y^g(l_i, l_j) + \sum_{i,j} y^r(l_i, l_j)} \tag{7}$$

CPC is always positive and contained in the closed interval $(0, 1)$ with 1 indicating a perfect match between the generated flows and the ground truth and 0 highlighting bad performance with no overlap.

D.4 Divergence metrics

The Kullback-Leibler (KL) divergence measures how different a probability distribution is from a reference probability distribution. It is used to assess the performance of a generative mobility model by calculating the extent to which synthetic trajectories and real trajectories are similar with respect to relevant mobility patterns. Formally, given two discrete probability distributions P and Q, defined on the same probability space X, the KL divergence from P to Q is defined as:

$$D_{\text{KL}}(P||Q) = \sum_{x \in X} P(x)log\left(\frac{P(x)}{Q(x)}\right). \tag{8}$$

Formally, given two probability distributions P and Q, KL divergence is the expectation of the logarithmic difference between the probabilities of P and Q, where the expectation is taken using the probabilities of P. KL

divergence is always non-negative ($D_{KL}(P||Q) \geq 0$) and not symmetric, i.e., $D_{KL}(P||Q) \neq D_{KL}(Q||P)$. P and Q are the same distribution if $D_{KL}(P||Q) = 0$.

The Jensen-Shannon (JS) divergence is a measure to assess the similarity between two distributions. It is based on the KL divergence but it is symmetric ($JS(P||Q) = JS(Q||P)$) and ranges in $[0, 1]$. Formally, given two probability distributions P and Q, and $M = \frac{1}{2}(P||Q)$, we define the JS divergence as:

$$D_{JS}(P||Q) = \frac{1}{2}D_{KL}(P||M) + \frac{1}{2}D_{KL}(Q||M). \tag{9}$$

The JS divergence is used, as an alternative to KL, to assess the performance of generative mobility models.

E HUMAN MOBILITY PATTERNS

Human movements, far from being random, follows well-defined statistical patterns [9, 194]. In this Section, we revise the most relevant spatial (Section E.1) and temporal (Section E.2) patterns of human mobility.

E.1 Spatial patterns

Displacements. The distance between two consecutive locations visited by an individual is called jump length or displacement [22, 65]. The term location usually indicates a spatial point in which an individual spent a minimum amount of time reflecting human behavioral tendencies that motivate people to move between two places. Formally, a jump length $\Delta r = d(s_i, s_{i+1})$ is the distance between two spatio-temporal points s_i and s_{i+1} in a trajectory $T_u = \langle s_1, s_2, ..., s_n \rangle$. A truncated power-law well approximates the empirical distribution $P(\Delta r)$ within a population of individuals, with the value of the exponent slightly varying based on the type of data and the spatial scale [22, 65].

Radius of gyration. The characteristic distance traveled by an individual u during a period of time can be quantified by their radius of gyration [65], defined as $r_g(u) = \sqrt{\frac{1}{n_u} \sum_{i=1}^{n_u} d(s_i, s_{cm})^2}$, where n_u is the number of points in T_u, $s_i \in T_u$ and $s_{cm} = \frac{1}{n_u} \sum_{i=1}^{n_u} s_i$ is the position vector of the center of mass of the set of points in T_u. A truncated power-law well approximates the distribution of r_g [65, 133]. At a collective level, the evolution in time of the average r_g of individuals follows a logarithmic curve $\langle r_g(t) \rangle \sim \alpha + \beta \ln t$ [65, 172].

The k-radius of gyration of an individual u is defined as the radius over their k most frequented locations [136], $r_g^{(k)}(u) = \sqrt{\frac{1}{N_k} \sum_{i=1}^{k} n_i d(s_i - s_{cm}^{(k)})^2}$, where $s_i \in T_u$, N_k is the sum of the visits to u's k most frequented locations, and $s_{cm}^{(k)} = \frac{1}{N_k} \sum_{i=1}^{k} s_i$ is the center of mass computed on u's k most frequented locations. The comparison of r_g and $r_g^{(k)}$ over an entire population revealed the existence of a returners and explorers dichotomy [136].

Mobility entropy. The temporal-uncorrelated entropy of an individual u characterizes the predictability of their spatial movements, and it is defined as $S_{unc}(u) = -\sum_{i=1}^{n_u} p_u(i) \log_2 p_u(i)$, where n_u is the number of distinct locations visited by u and $p_u(i)$ is the probability that u visits location i [48, 173]. The real entropy of an individual u considers also the order in which the places were visited and the time spent at each location, and it is defined as $S(u) = -\sum_{T'_u \subset T_u} P(T'_u) \log_2 P(T'_u)$ [173], where $P(T'_u)$ is the probability of finding a particular time-ordered subsequence T'_u in the trajectory T_u. The distribution of both S_{unc} and S are peaked and in particular $P(S)$ peaks around $S \approx 0.8$, indicating that the real spatio-temporal uncertainty in a typical user's whereabouts is $2^{0.8} = 1.74$, i.e., fewer than two locations [173].

I-rank and G-rank. Location ranks identify the importance of a location to an individual's mobility: the most visited location (likely home or work) has rank 1, the second most visited location (e.g., school or local shop) has rank 2, and so on. The visitation frequency of locations $P(L)$, or I-rank, follows a Zipf law: the probability of finding an individual at a location of rank L is well approximated by $P(L) \sim 1/L$ [65, 133]. Similarly, the collective

visitation frequency of a location $P(r)$, or G-rank, indicates the popularity of locations according to how people visit them on the geographic space [128, 134].

Semantic importance. For an individual's trajectory T_u, we define $p_{d_{\text{total}}}(r) = \frac{d_{total}(r)}{\sum_{r'} d_{total}(r')}$, where $d_{total}(r)$ is the total stay duration in location r interpreted as r's semantic importance [16, 128]. The semantic distance between two trajectories T_u and T_v is the distance between the distribution of the $p_{d_{total}}(r)$ for u and v.

Mean Distance Error. Given two equal-sized sets of trajectories $\mathcal{T} = \{T_1, \ldots, T_N\}$ and $\hat{\mathcal{T}} = \{\hat{T}_1, \ldots, \hat{T}_N\}$, the Mean Distance Error (MDE) between $\mathcal{T}$ and $\hat{\mathcal{T}}$ is defined as $MDE = \frac{\sum_i^N d(T_i, \hat{T}_i)}{N}$, where d is the distance between two points [76].

Mobility networks. In an individual mobility network (IMN), nodes represent locations and directed edges represent an individual's trips between locations [150, 160]. The vast majority of individuals' trips can be described with a limited number of daily motifs which represent the underlying regularities in daily movements [160]. Individual trajectories may be aggregated to study the flows of individuals between locations at different spatio-temporal scales. Flows can typically be described by an Origin-Destination (OD) matrix, or mobility network, which has a specific structure and dynamics [167].

E.2 Temporal metrics

Waiting time and circadian rhythm. The waiting time Δt is the elapsed time between two consecutive points in the mobility trajectory of an individual u, or equivalently as the time spent in a location: $\Delta t = t_i - t_{i-1}$. Empirically the distribution of waiting times is well approximated by a truncated power-law [172] The movements of individuals are not distributed uniformly during the hours of the day but follow a circadian rhythm [92, 134]; people tend to be stationary during the night hours while prefer moving at specific times of the day, for example, to reach the workplace or return home.

Temporal location patterns. The temporal popularity $p(r, t)$ measures the visiting probability for a location r at any time t [128]. The staying patterns $p(r, d)$ measures the probability of visiting a location r for a duration d [128].