Deep Learning From Trajectory Data: a Review of Deep Neural Networks and the Trajectory Data Representations to Train Them

Anita Graser[1], Anahid Jalali[1], Jasmin Lampert[1], Axel Weißenfeld[1] and Krzysztof Janowicz[2]

[1]AIT Austrian Institute of Technology, 1210 Vienna, Austria
[2]University of Vienna, Vienna, Austria

Abstract
Trajectory data combines the complexities of time series, spatial data, and (sometimes irrational) movement behavior. As data availability and computing power have increased, so has the popularity of deep learning from trajectory data. This paper aims to provide an overview of deep neural networks designed to learn from trajectory data, focusing on recent work published between 2020 and 2022. We take a data-centric approach and distinguish between deep learning models trained using dense trajectories (quasi-continuous tracking data), sparse trajectories (such as check-in data), and aggregated trajectories (crowd information).

Keywords
Deep learning, data engineering, movement data, trajectories

1. Introduction

Deep learning has become a popular approach for developing data-driven prediction, classification, and anomaly detection solutions. Work on deep learning from trajectory data is spread out over many domains, including but not limited to computer science, geography, geographic information science, urban planning, and ecology. Consequently, it covers many use cases and corresponding trajectory dataset types.

Trajectory datasets can be categorized according to the level of detail: from dense trajectories (quasi-continuous tracking data of individual movement) to sparse trajectories (such as check-in data of individuals), and finally, aggregated trajectories (crowd-level information, typically aggregated to edges/nodes in a mobility graph, to a grid, or to a set of points of interest)[1, 2, 3]. In many cases, the titles and abstracts of papers are not sufficient to determine which type of trajectory data was used to train the deep learning model. While many papers start with dense trajectories, most convert them into sparse trajectories [4, 5, 6, 7, 8, 9, 10] or even aggregate them to crowd-level [11, 12, 13]. Common approaches to turning dense trajectories into sparse trajectories include: converting them into a sequence of stop locations [4, 7] or a sequence of traversed regions (grid cells) [6, 8], or converting them to trajectory images [5, 9, 10].

In a related review of location encoding methods for GeoAI [14], the authors stress the analogy between NLP word-to-sentence relations and location-to-trajectory relations. This analogy has led to Word2Vec-inspired approaches encoding location into a location embedding using, for example, Location2Vec [15], Place2Vec [16], or POI2Vec [17]. Another recent review [18] is dedicated to deep learning for traffic flow prediction models, which are primarily trained on aggregated trajectory data. However, to the best of our knowledge, there is no review paper that provides an overview of the different neural network architectures used to learn from trajectory data.

The goal of this work is to provide a first overview of the current state of neural networks / deep learning trained with trajectory data, structured by 1. Use case category (travel time/crowd flow/location predictions, location/trajectory classifications, anomaly detection), 2. Neural network architecture (CNN, RNN, LSTM, GNN, ...), and 3. Trajectory data granularity (dense, sparse, aggregated) and representation. Therefore, this review does not include classic ML approaches and does not provide an exhaustive historical analysis of the field. Due to the page limit, this paper does not fit an exhaustive list of all relevant works published in recent years. However, we provide at least one paper for each use case and network combination we identified. We specifically reviewed publications at recent events, including SIGSpatial 2022[1] [6, 7, 19, 8, 20, 12, 21, 22], Sussex-Huawei Locomotion (SHL) Challenge 2021[2] at the ACM interna-

Proceedings of the Workshop on Big Mobility Data Analytics (BMDA) co-located with EDBT/ICDT 2023 Joint Conference (March 28-31, 2023), Ioannina, Greece
anita.graser@ait.ac.at (A. Graser); anahid.jalali@ait.ac.at (A. Jalali); jasmin.lampert@ait.ac.at (J. Lampert); axel.weissenfeld@ait.ac.at (A. Weißenfeld); krzysztof.janowicz@univie.ac.at (K. Janowicz)
0000-0001-5361-2885 (A. Graser)

[1]https://sigspatial2022.sigspatial.org/accepted-papers/
[2]http://www.shl-dataset.org/activity-recognition-challenge-2021/

	CNN	(C)RNN	LSTM	GRU	Transformer	GNN	Other
Location classification	• Yang (2018)					• Altan (2022)	• **Memory network**: Lyu (2022)
Arrival time prediction		• Wang (2018) • Buijse (2021)		• Buijse (2021)		• Derrow-Pinion (2021)	
Traffic volume / crowd flow prediction	• Wang (2022) • Lu (2021) • Kashyap (2022)*	• Kashyap (2022)*	• Buroni (2021) • Li (2021)		• Xue (2022)	• Gao (2022) • Li (2021) • Lippert (2022)	• **GAN**: Zhang (2020) • **SAE**: Kashyap (2022)*
Trajectory prediction (& imputation)			• Mehri (2021)	• Tritsarolis (2021) • Fan (2022)	• Carroll (2022) • Musleh (2022)		
(Sub)trajectory classification	• Chen (2020) • Yang (2022)	• Wang (2021)*	• Wang (2021)*		• Wang (2021)*		• **AdaNet**: Wang (2021)*
Next location (& destination) prediction	• Gao (2019)	• Feng (2018) • Liao (2018)		• Gao (2019)	• Hong (2022)		• **Hyper network**: Tenzer (2022)
Anomaly detection		• Nguyen (2022) • Sing (2022)	• Liatsikou (2021)				
Synthetic data generation			• Rao (2020)				• **GAN**: Rao (2020) • **MLP**: Simini (2021) • **VAE-like**: Zhang (2022)

Dataset categorization: Trajectories of individual movers: dense / quasi-continuous, sparse / check-in data or gridded data and aggregated trajectories of multiple movers: aggregate data / crowd-level

* review / summary paper

Figure 1: Overview of use cases and neural networks for trajectory data included in this review.

tional joint conference on pervasive and ubiquitous computing (UbiComp) [23], Traffic4cast challenge 2021[3] at NeurIPS [24], and Big Movement Data Analytics workshop BMDA 2021[4] at EDBT [11, 25, 26].

Even though we focus explicitly on deep learning, it is worth noting that deep learning may not always be the best approach [27]. In particular, the SHL Challenge summary [23] shows that regular machine learning models outperform deep learning models on all three metrics: F1 score, train time, and test time.

This review does not attempt to compare the performance of different deep-learning approaches. Even though there are some commonly used open datasets, such as the Porto taxi data[5], the T-Drive taxi dataset[6] and GeoLife dataset[7], and the Gowalla check-in data[8], cross-paper comparisons outside of dedicated data challenges are notoriously difficult. For example, "Despite the Porto dataset's original use as a standardized benchmark for open competition, design choices in subsequent work make cross-paper comparison difficult. Firstly, different papers often augment the dataset with their metadata not present in the original release, which may give some models an advantage over others independent of architecture or training design." [20]

Due to the large range of domains working on trajectory data analysis, the terminology used in different publications is not necessarily consistent. We, therefore, define the most important terms and abbreviations in a glossary at the end of this paper.

[3]https://www.iarai.ac.at/traffic4cast/2021-competition/

[4]https://www.datastories.org/bmda21/BMDA21Accepted.html

[5]https://www.kaggle.com/c/pkdd-15-predict-taxi-service-trajectory-i/data

[6]https://www.microsoft.com/en-us/research/publication/t-drive-trajectory-data-sample/

[7]https://www.microsoft.com/en-us/research/publication/geolife-gps-trajectory-dataset-user-guide/

[8]http://snap.stanford.edu/data/loc-Gowalla.html

2. Representing Trajectory Data For Deep Learning Use Cases

This review is structured around eight use case categories of deep learning from trajectory data, as shown in Figure 1. The following subsections describe the use cases, neural network designs used to address them, and trajectory data used to train these networks. Figure 2 provides an overview of the diversity of identified trajectory representations. More details on the trajectory datasets and the data engineering steps applied to the trajectory data before they are used as input to train the neural networks are summarized in Tables 1-3. Some works (e.g. [28, 29]) use additional data sources in combination with trajectory data to train their models. These additional data sources have been omitted from our review in favor of clarity and conciseness.

2.1. Location classification

This use case category covers the classification of locations using patterns derived from movement data. The classification of regionally dominant movement patterns may be of interest in and of itself [9] or help with the classification of POIs (e.g. ports [4]) or the classification of trip destinations [19].

To detect regionally dominant movement patterns, Yang et al. [9] use direction information and density maps to generate directional flow images. They convert the trajectories into images where each pixel contains the

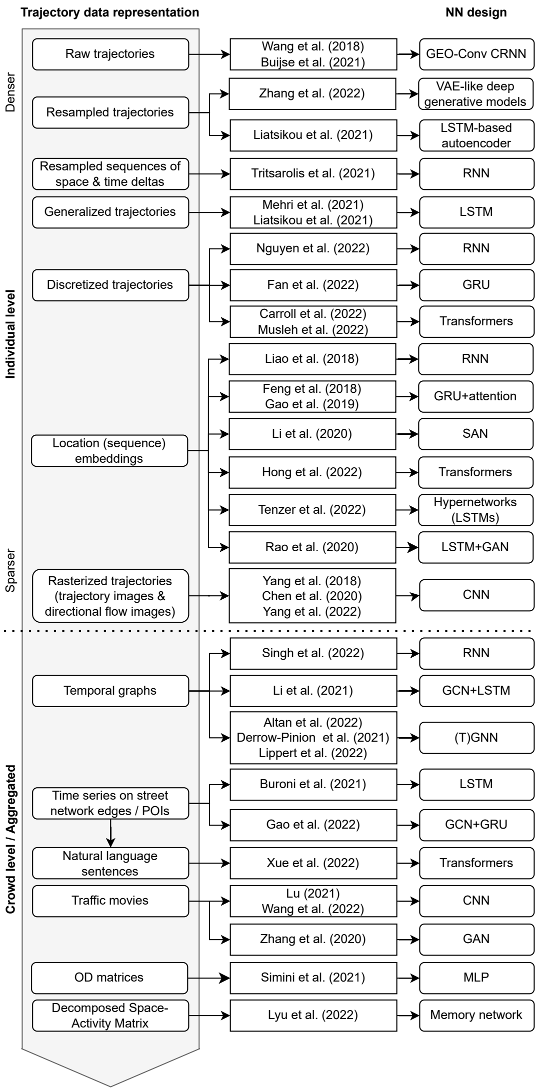

Figure 2: Overview of trajectory data representations used to train neural networks

directional flows. They use a CNN to classify the input image patterns and detect the dominant regional movements.

An approach that makes more use of temporal information is presented by Altan et al. [4]. They use a temporal GNN (TGNN) to distinguish gateway ports from actual ports using AIS vessel movement data. After extracting the ports (nodes) from the raw AIS messages using DBSCAN, they extract trips between consecutive ports and build a graph for each time step to generate the time-ordered daily graph sequence for the TGNN.

Lyu et al. [19] train a plug-in memory network to predict trip purposes based on destination locations. Their model is trained using activity, origin, and destination matrices derived from OD data using a non-negative Tucker decomposition scheme.

2.2. Arrival time prediction

This use case category covers the prediction of travel times or arrival times, such as arrival time prediction in train networks [30] and street networks [31, 32].

Since travel time often depends on historical travel times at a given time of day, recurrent mechanisms are commonly used [31, 32, 30]. For example, Derrow-Pinion et al. [31] train GNNs on aggregated trajectories to provide travel time predictions in Google Maps. The GNN graph consists of segment and supersegment-level embedding vectors. Nodes store street segment-level data (average real-time and historical segment travel speeds and times, segment length, and road class), while edges store supersegment-level data (real-time supersegment travel times).

In contrast, Wang et al. [32] introduces the GEO-convolutional network layer (*GEO-Conv*, also used by Buijse et la. [30]), which is trained on dense trajectories stating that "directly mapping the GPS coordinates into grid cells is not accurate enough to represent the original spatial information in the data". The proposed *GEO-Conv layer* takes dense trajectories as input and applies a non-linear mapping of each trajectory (latitude and longitude) point, followed by a GEO-Conv step with multiple kernels. The resulting feature map of local paths is appended with a final column of distances of the local paths.

2.3. Traffic volume prediction

This use case category covers traffic or crowd predictions of volumes or flows, e.g., predicting traffic volume on street segments [11, 18], human activity at specific POIs [28] and metropolitan areas [33, 24], or predicting animal movement dynamics [34].

Aggregated trajectory data in the form of traffic movies is provided in the *Traffic4Cast 2021* competition which challenged participants to predict traffic under conditions of temporal domain shift (Covid-19 pandemic) and spatial shift (transfer to entirely new cities). Lu [24] won this challenge using CNN (U-Net) and multi-task learning. Their multi-task learning approach randomly samples from all available cities and trains the U-Net model to jointly predict the future traffic states for different cities. Wang et al. [12] follow this traffic movie approach as well by aggregating individual-level trajectories into a grid with inflow referring to the total number of incoming traffic entering this region from other regions during a given time interval and outflow representing the total number of traffic leaving the region. Zhang et al. [13] also follow the traffic movie approach, creating temporal grids of average traffic speed and taxi inflow per cell.

Li et al. [33] build a graph for their GCN by aggregating CDR data and representing spatial statistical units as nodes and their relationship (physical distance, physical movement, phone calls) as edges. Similarly, Lippert et al. [34] build temporal graphs from bird migration data where nodes represent radar locations, and edges represent the flows between the Voronoi tessellation cells of the radar locations.

Finally, Buroni et al. [11] provide a tutorial using vehicle counts derived from GPS tracks to build and train a Direct LSTM encoder-decoder model. The model is trained to predict counts of vehicles per network edge per time step for the Belgian motorway network. Similarly, Gao et al. [28] use their GPS tracks to count vehicles per POI per time step (hourly) to train a GCN+GRU model that predicts these visit counts. And Xue et al. [21] propose a translator called *mobility prompting* which converts daily POI visit counts into natural language sentences so they can use (and fine-tune) pre-trained NLP models such as Bert, RoBERTa, GPT-2, and XLNet to predict these visit counts.

2.4. Trajectory prediction/imputation

This use case category covers the prediction of trajectories in artificial [35], urban [6], and maritime environments [36, 26], as well as imputation of trajectories [8].

Mehri et al. [36] generalize AIS trajectories using context-aware piecewise linear segmentation before feeding them into their LSTM three vertices at a time. This enables their model to perform short-term trajectory predictions with high spatial detail. Tritsarolis et al. [26], on the other hand, represent trajectories by their composition of differences in space Δx, Δy, and time Δt for the input of their RNN-based models to predict the vessel's position at time $\Delta t + 1$.

Fan et al. [6] discretize mobile phone GPS trajectories using the H3 hexagonal grid and use the grid cell sequences to train their GRU. The resulting model is used to predict cell sequences which are afterward used to search for similar high-resolution trajectories, which are returned as the final trajectory prediction.

Carroll et al. [35] use synthetic discrete movement sequences in a minimalistic grid world environment. Their transformers are trained on trajectories as sequences of states, actions, and return-to-go tokens to predict trajectories. Another work using transformers to impute trajectories is Musleh et al. [8]. They propose TrajBERT, a model trained using H3-discretized (tokenized) GPS tracks. They "down-sample the trajectories by dropping three-quarters of the points of each trajectory and then run TrajBERT to fill the gaps by imputing the missing points".

2.5. (Sub)trajectory classification

This use case category considers the classification of complete trajectories [10] or sub-trajectories [5, 23] in order to learn more about different vessel and human movement patterns.

By splitting trajectories into sub-trajectories, more fine-grained analyses are possible. Typical applications include the detection of movement types, such as ship maneuvers [5] or the detection of transportation and locomotion modes of smartphone users [23]. Chen et al. [5] generate colour-coded trajectory images from ship AIS data, where each pixel is assigned one of three colors according to the movement type (static, normal, maneuvering). These trajectory images are used to train a CNN-based ship maneuver classifier. To identify different movement modes (i.e. still, walk, run, bike, car, bus, train, and subway) from smartphone data, the SHL Challenge winner uses an AdaNet algorithm, a Tensorflow-based framework for learning NN models and ensembling models to obtain even better models [46, 23].

An example of the classification of complete trajectories is the recognition of ship types introduced by Yang et al. [10]. It relies on the same technique as [5] for transforming the raw AIS trajectory data into colour-coded trajectory images. The resulting images show characteristic trajectory patterns, which can be used to identify the ship vessel type with a CNN classifier.

2.6. Next location / final destination prediction

This use case category covers the prediction of the next locations or final destinations of trips [38, 37, 7, 39, 40, 41, 21, 20]. Besides GPS tracks, a commonly used data source in this category are social media check-ins (e.g., from Foursquare). The task then becomes to predict the next check-in location (e.g., a POI).

Attention mechanisms have proven to be a popular approach for next location prediction. Gao et al. [37] train *VANext*, a semi-supervised network trajectory convolutional network, on check-in data. They convert each individual user's trajectory (check-in/POI sequence) into

[9] https://www.kaggle.com/datasets/giobbu/belgium-obu
[11] https://coast.noaa.gov/htdata/CMSP/AISDataHandler/2017/index.html
[12] https://zenodo.org/record/4498410
[13] https://sites.google.com/site/yangdingqi/home/foursquare-dataset?pli=1
[14] http://www.start.umd.edu/gtd/
[15] https://github.com/bigdata-ufsc/petry-2020-marc/tree/master/data/foursquare_nyc
[16] https://drive.google.com/file/d/1rLJz5E0igbrmAnmnDmazdBl97UuQ0sch/view?usp=sharing

Table 1
Trajectory datasets and data engineering; public datasets are printed in bold

Ref	NN design	Trajectory data	Data engineering
Location classification			
Yang et al. (2018) [9]	CNN	Synthetic data (manually drawn trajectories, rotated in a data augmentation step)	Trajectories are converted to **directional flow images (DFI)** (resolution: 10×10)
Altan et al. (2022) [4]	TGNN	Ship AIS tracks in Halifax, Canada, covering 15 ports, 10 vessels, for 4 months (total: 513k AIS records)	Trajectories are converted to **temporal (daily) graphs** of ports (nodes with associated port visit frequencies, waiting times, and speed statistics) and trips between consecutive ports (edges)
Lyu et al. (2022) [19]	Memory	OD data from Tokyo Metro travel survey including travel mode, time, and purpose	ODs are converted into a **Space-Activity Matrix** which is then decomposed into an activity, an origin, and a destination matrix
Arrival time prediction			
Wang et al. (2018) [32]	CRNN	Taxi GPS tracks in Chengdu and Beijing with a density of 2.6 to 5.5 GPS records per km	**Trajectories** are fed into a *GEO-Conv* layer
Buijse et la. (2021) [30]	CRNN+GRU	Train GPS tracks in the Netherlands covering 350k train trips	**Trajectories** are fed into a *GEO-Conv* layer
Derrow-Pinion et al. (2021) [31]	GNN	Google Maps road segments with travel times/speeds	Road network graph edges are subdivided into shorter segments (modeled as **GNN graph** nodes) with associated aggregated travel time/speed information; segments are combined into supersegments (GNN graph edges)
Traffic volume prediction			
Lu (2021) [24]	CNN (U-Net)	Traffic movies for 10 cities in 2019+2020 with 8 dynamic channels encoding traffic speed and volume per direction and 9 static channels encoding the properties of the road maps	The multi-task learning randomly samples from all available city **traffic movies** (resolution: 495×436)
Wang et al. (2022) [12]	CNN	Taxi trajectories from TaxiBJ in Beijing for 17 months and bike trajectories from BikeNYC in New York City for 6 months	Trajectories are converted to **flow/traffic movies** (32×32 for TaxiBJ & 16×8 for BikeNYC) with two dynamic channels encoding inflows and outflows
Li et al. (2021) [33]	GCN+LSTM	Sparse CDR trajectories in Senegal of 100,000 individuals for one year	Trajectories are converted into a **movement graph** with edges representing the number of transitions from one cell phone tower node to the next
Buroni et al. (2021) [11]	LSTM	**Lorry GPS tracks**[9] in Belgium with 30s reporting interval, anonymous IDs (reset daily), timestamp, latitude, longitude, speed, and direction	Trajectories are matched to street network to **count vehicles per network edge per time step**
Gao et al. (2022) [28]	GCN+GRU	Taxi GPS tracks in Xi'an, China covering 7.7k taxis for 3 months with a sampling interval of 5–30 s	Trajectories are used to **count vehicles per POI (n=100) per time step (hourly)**
Lippert et al. (2022) [34]	Recurrent GNN	Bird migration data in the form of simulated trajectories and measurements from the European weather radar network	Trajectories are converted **temporal graphs** with to flows (edges) between Voronoi tessellation cells of radar locations (nodes).
Xue et al. (2022) [21]	Transformers	SafeGraph daily POI visit counts in NYC, Dallas and Miami	Historical visitation data are translated into natural language sentences to fine-tune pretrained NLP models
Zhang et al. (2020) [13]	GAN	Taxi GPS tracks in Shenzhen, China for 6 months	Trajectories are converted to **traffic movies** (resolution: 40×50, hourly) with average traffic speed and taxi inflow

Table 2
Trajectory datasets and data engineering; public datasets are printed in bold – continued

Ref	NN design	Trajectory data	Data engineering
Trajectory prediction/imputation			
Mehri et al. (2021) [36]	LSTM	**AIS data from NOAA**[10] for the US East Coast, containing 58.5mio messages from 10.7k vessels over 2 months	Trajectory is generalized using context-aware piecewise linear segmentation. The LSTM is trained on three vertices at a time
Tritsarolis et al. (2021) [26]	RNN	**Ship AIS tracks in Piraeus**[11] containing 138k records from 246 fishing vessels and **GeoLife**[7]	Trajectories are resampled to regular 1 minute intervals and converted into sequences of differences in space Δx, Δy, and time Δt to predict the vessel's position in the next future step
Fan et al. (2022) [6]	GRU	Mobile phone GPS tracks in the Kanto area of Tokyo covering 220k users with a minimum reporting period of 5 minutes for 2 months	Trajectories are **discretized using the H3** hexagonal grid
Carroll et al. (2022) [35]	Transformers	Synthetic data in a minimalistic grid world environment	Transformers are trained on trajectories as sequences of states, actions, and return-to-go tokens
Musleh et al. (2022) [8]	Transformers	GPS tracks in San Francisco from the GISCUP'17 dataset with 5M records	Trajectories are **discretized using the H3** hexagonal grid (tokenization) followed by the creation of spatial embeddings
(Sub)trajectory classification			
Chen et al. (2020) [5]	CNN	**Ship AIS tracks in Tianjin, China** covering 23k trips	Trajectories are converted into **trajectory images** with pixels coloured according to movement type (static, maneuvering, normal)
Yang et al. (2022) [10]	CNN	**Ship AIS tracks in Northern America** of the U.S. National Oceanic and Atmospheric Administration's Office of Coastal Management with 259k records after pre-processing	Trajectories are converted into **trajectory images** with the same method as described in [5].
Wang et al. (2021) [23]	(C)RNN, LSTM, Transformers, AdaNet	SHL dataset[12]: asynchronously sampled radio data of smartphones with up to 2,812 hours of labeled data, e.g. GPS reception and location, Wifi reception and GSM cell tower scans	[SHL challenge summary paper] Mostly hand-crafted feature engineering as input, but also two teams using raw trajectory data
Next location / final destination prediction			
Gao et al. (2019) [37]	GRU & CNN	Foursquare check-ins in New York and Singapore and Gowalla data in Houston and California with an average of 229k records of 3k users	Location (<id, lon, lat>) sequences are converted into **sequences embeddings** using a causal embedding method
Feng et al. (2018) [38]	Attention-GRU	**Foursquare check-ins**[13] with 294k records covering 15k users, other mobile application location records (search & check-ins) with 15mio records covering 5k users, and CDR with 491k records covering 1k users	Location sequences are converted into **embeddings** using two independent attention mechanisms which are then fed into *DeepMove's* GRUs and a historical attention module
Li et al. (2020) [39]	SAN	**Foursquare check-ins**, Tweets, Yelp and NYC data	Location sequences are converted into **embeddings**
Liao et al. (2018) [40]	RNN	**Foursquare check-ins** in NYC and Tokyo with an average of 400k records of 1.7k users	Location sequences and activity/location graphs are converted into **embeddings** for *MCARNN*
Liu et al. (2016) [41]	RNN	**Gowalla check-ins**[8] and **Global Terrorism Database (GTD) incidents**[14]	Location sequences are converted into time-specific and distance-specific transition matrices for *ST-RNN*
Hong et al. (2022) [7]	Transformers	Mobile phone GPS tracks in Switzerland from the Green Class (GC) study covering 139 participants for a year and from the yumuv study covering 498 participants for 3 months	Location sequences are generated from GPS tracks by first filtering stay locations with a stay duration >25min and then spatially aggregating stays into locations. These location sequences are converted into location, time, day, and mode embeddings which are fed into the transformers
Tenzer et al. (2022) [20]	Hyper network (LSTMs)	**Porto taxi tracks**[5] covering 1.7mio trips by 442 taxis in Porto, Portugal for 12 months	Sequences of trajectory points are converted to **sequences of spatial embeddings** via a geospatial encoding mechanism

Table 3
Trajectory datasets and data engineering; public datasets are printed in bold – continued

Ref	NN design	Trajectory data	Data engineering
Anomaly detection			
Liatsikou et al. (2021) [25]	LSTM-based AE	**Porto taxi tracks**[5]	Trajectories are down-sampled to 60s and represented as a **sequence of vectors** (lat, lon) and clipped to the first nine points to fit the autoencoder requirements
Nguyen et al. (2022) [42]	Probabilistic RNN	Ship AIS tracks from a single receiver in Ushant, France	Trajectories are down-sampled to 600s and converted to a **four hot vectors** (lat, lon, SOG, COG) with the resolutions of 0.01° for lat/lon, 1 knot for SOG, and 5° for COG.
Singh et al. (2022) [43]	RNN regression model	Ship AIS tracks in the Baltic sea region and near Bremerhaven, Germany for two months (including a comparison between satellite-based and coastal AIS)	Trajectories are resampled and interpolated at the 60s and converted into a **graph** with nodes representing turning points for the vessel trajectories and edges representing the sea lanes traveled by vessels
Synthetic data generation			
Rao et al. (2020) [44]	LSTM-GAN	**Foursquare check-ins**[15] in NYC covering 193 users with 3k trajectories and 67k records	Trajectories are processed by a trajectory encoding model covering trajectory point encodings (location, temporal and categorical attributes) and trajectory padding (to ensure that all trajectories have the same length)
Zhang et al. (2022) [22]	VAE-like deep generative models	**Porto taxi tracks**[5]; **T-Drive**[6] data consisting of 10.3k taxis for one week; and **Gowalla check-ins**[8]	A coordinate encoding MLP converts two-dimensional points into a high-dimensional representation. Then, a Bidirectional LSTM is used to encode all representations with forward and backward information for a time step
Simini et al. (2021) [45]	MLP	England & Italy commuting flows, **NY State flows**[16] including origin & destination geographic unit and estimated population flows between two geographic units	Input to the model is of the origin & destination location as well as the distance between origin and destination. The output of the model is the probability to observe a trip between two locations.

sequence embeddings using a causal embedding method (similar to a high-order Markov Process). The resulting embeddings are the input for their GRU to learn the trajectory patterns. They further apply attention to the embeddings for predicting the user's next POI. Feng et al. [38] tailor two attention mechanisms to generate independent latent vectors from large and sparse trajectories. These embeddings are then fed into their *DeepMove* GRUs and a historical attention module. The learned attention weights can intuitively explain the prediction based on the user's history of movement behavior. Li et al. [39] introduce a spatio-temporal self-attention network (*STSAN*). They generate trajectory embeddings by concatenating the temporal (activity sequence), spatial (distance matrix of locations), and location attentions (location sequence and their categories). They feed these embeddings through a softmax layer and predict the user's next POI. They use a federated learning setting to tackle the heterogeneity problem. Liao et al. [40] generate embeddings from location sequences as well as graph embeddings from location-location and activity-location graphs and train their *MCARNN* multi-task context-aware recurrent neural network to solve both activity and location prediction tasks.

Other works use neural networks for dimensionality reduction and for creating embeddings. Liu et al. [41] incorporate time and distance-specific transition matrices as temporal and spatial embeddings generated by RNNs. Hong et al. [7] reduce the dimensions of trajectories using a multilayered embedding approach for transformers to predict next location and travel mode. Tenzer et al. [20] generate two geospatial and temporal embeddings by 1. combining the random picking and the nearest neighbor to create sequences of spatial embeddings and 2. using a sinusoidal embedding to convert the timesteps to temporal vectors. They train a hyper network to learn to change its weights in response to these embeddings.

2.7. Anomaly detection

This use case category covers anomalous trajectory detection. Since the definition of anomalies is often context-dependent, ground truth labeled data is rare. Therefore, anomaly detection approaches often resort to trying to identify trajectories that are different compared to previously observed trajectories based on some spatial, spatiotemporal, or other metrics. Alternatively, researchers resort to using synthetically generated anomalies [25].

Liatsikou et al. [25] developed an LSTM-based network for the automatic detection of movement anomalies, such as the detection of synthetic anomalies in taxi trajectories. Since the autoencoder requires inputs of a certain fixed length, all trajectories are clipped to nine points (and shorter ones discarded).

The GeoTrackNet [42] is a model for maritime trajectory anomaly detection, which consists of a probabilistic RNN-based (Recurrent Neural Network) representation of AIS tracks and a contrario detection [47]. Detected anomalies were evaluated by AIS experts.

Singh et al. [43] present an anomaly detection system based on RNN regression models to detect anomalous trajectories, on-off switching, and unusual turns. Again, a quantitative accuracy analysis is not feasible due to the lack of ground truth data.

2.8. Synthetic data generation

This category covers the generation of synthetic movement data, such as synthetic trajectories [22, 44] and synthetic flows [45].

Rao et al. [44] focus on GeoAI-trajectory privacy protection. For this, they develop an end-to-end deep LSTM-TrajGAN model to generate privacy-preserving synthetic trajectory data for data sharing and publication.

Simini et al. [45] developed an MLP model (denoted Deep Gravity) to generate mobility flow probabilities. They evaluated Deep Gravity on mobility flows in England, Italy, and New York State and achieved a good performance even for regions with no data available for training.

Zhang et al. [22] propose an end-to-end trajectory generation model for generating synthetic trajectories. The design of the model is VAE-like encoders (e.g., Global-semantics encoder: MLPs & Bidirectional LSTM) and decoders (e.g., a prior generator based on variational recurrent structure generates noise at time t by considering the noise at the previous time step).

3. Conclusion and outlook

In this work, we reviewed deep learning-based research focussing on mobility data. In most cases, even if trajectory data is used in the process, it is not ingested directly for training the neural networks. Instead, data engineering steps are applied that convert trajectories into more compact representations of individual trajectories (sparse trajectories) or aggregations of multiple trajectories. This aggregated trajectory data is commonly presented as time series of vectors, graphs, or images (movies).

On the deep learning side, we expect the popularity of GNNs to increase. For example, the Traffic4cast challenge (in its 4th year, 2022) is moving from (image/video-based) traffic forecasting to graph-based representations. Additionally, AutoML methods (e.g., AdaNet used by the SHL Challenge winner [23]) will allow users with limited DL expertise to build competitive DL models.

As far as data engineering and development is concerned, we expect further uptake of trajectory analysis libraries, such as *Trackintel*[17] (e.g., used by [7]), *MovingPandas*[18] (e.g., used by [36]) and *scikit-mobility*[19] (e.g., used by [45]) since these libraries implement many common trajectory generalization, aggregation, and analysis methods and aim at a long(er) term availability. This is an important next step, as the implementations summarized in Figure 2.8 have not been substantially updated/maintained since being published. This does not only reduce the likelihood of reuse but also will lead to security issues down the road.

Future research should address the issues of model transferability, benchmark availability, and model explainability. Current work rarely addresses the issue of model transferability. Since most existing global ML models "cannot perform well locally, or be transferred to study similar problems in other regions"[48], transferability should be considered when evaluating or comparing models. Additionally, developed models, even for the same application and trajectory type, are difficult to evaluate (e.g., due to the lack of ground truth for anomaly detection) and to compare due to different datasets and applied metrics. Therefore, more open datasets are needed. Finally, to better understand the why and how of using neural networks for a specific application, explainability should play a more crucial role in model development.

References

[1] G. Andrienko, N. Andrienko, P. Bak, D. Keim, S. Kisilevich, S. Wrobel, A conceptual framework and taxonomy of techniques for analyzing movement, Journal of Visual Languages & Computing 22 (2011) 213–232. URL: https://www.sciencedirect.com/science/article/pii/S1045926X11000139. doi:https://doi.org/10.1016/j.jvlc.2011.02.003.

[17] https://github.com/mie-lab/trackintel
[18] http://movingpandas.org
[19] https://scikit-mobility.github.io/scikit-mobility/

Authors	Model/System Name	Code Repository	Stars*	ML Library
Feng et al. (2018)	DeepMove	https://github.com/vonfeng/DeepMove	122	PyTorch
Wang et al. (2018)	DeepTTE with GEO-Conv layer	https://github.com/UrbComp/DeepTTE	118	PyTorch
Simini et al. (2021)	DeepGravity	https://github.com/scikit-mobility/DeepGravity	39	PyTorch
Zhang et al. (2020)	Curb-GAN	https://github.com/Curb-GAN/Curb-GAN	22	PyTorch
Hong et al. (2022)		https://github.com/mie-lab/location-mode-prediction	3	PyTorch
Lippert et al. (2022)	FluxRGNN	https://github.com/FionaLippert/FluxRGNN https://zenodo.org/record/6921595	3	PyTorch
Buijse et al. (2021)	Deep Train Arrival Time Estimator	https://github.com/basbuijse/train-arrival-time-estimator	2	PyTorch
Fan et al. (2022)		https://github.com/fanzipei/crowd-context-prediction/tree/master	0	PyTorch
Xue et al. (2022)	AuxMobLCast	https://github.com/cruiseresearchgroup/AuxMobLCast	0	PyTorch
Zhang et al. (2022)	TrajGen	https://github.com/tongjiyiming/TrajGen	0	PyTorch
Li et al. (2021)		https://figshare.com/articles/dataset/Prediction_of_human_activity_intensity_using_the_interactions_in_physical_and_social_spaces_through_graph_convolutional_networks/11829306/1	107	TensorFlow
Nguyen (2021)	GeoTrackNet	https://github.com/CIA-Oceanix/GeoTrackNet	55	TensorFlow
Singh et al. (2022)	Uncertainty EDL Graph	https://github.com/sansastra/uncertainty_edl_graph	5	Tensorflow
Rao et al. (2020)	LSTM-TrajGAN	https://github.com/GeoDS/LSTM-TrajGAN	28	Keras
Chen et al. (2020)	CNN-SMMC	https://github.com/rechardchen123/Ship_movement_classification_from_AIS	11	Keras
Buroni (2021)	Tutorial on traffic forecasting with DL	https://www.kaggle.com/code/giobbu/lstm-encoder-decoder-tensorflow	9	Keras
Liatsikou (2021)	BMDA anomaly detection	https://github.com/marialiatsikou/BMDA_anomaly_detection	0	Keras

* The stars column lists the number of Github stars, Kaggle upvotes, or Figshare/Zenodo downloads of the respective repository (as of December 2022)

Figure 3: Neural network implementations published together with their reviewed papers, ordered by ML library and stars.

[2] M. P. Konzack, Trajectory analysis: Bridging algorithms and visualization, Ph.D. thesis, Technische Universiteit Eindhoven, 2018.

[3] Y. Zheng, Trajectory Data Mining: An Overview, ACM Transactions on Intelligent Systems and Technology 6 (2015) 1–41. URL: http://dl.acm.org/citation.cfm?doid=2764959.2743025. doi:10.1145/2743025.

[4] D. Altan, M. Etemad, D. Marijan, T. Kholodna, Discovering Gateway Ports in Maritime Using Temporal Graph Neural Network Port Classification, 2022. URL: http://arxiv.org/abs/2204.11855.

[5] X. Chen, Y. Liu, K. Achuthan, X. Zhang, A ship movement classification based on Automatic Identification System (AIS) data using Convolutional Neural Network, Ocean Engineering 218 (2020) 108182. URL: https://linkinghub.elsevier.com/retrieve/pii/S0029801820311124. doi:10.1016/j.oceaneng.2020.108182.

[6] Z. Fan, X. Yang, W. Yuan, R. Jiang, Q. Chen, X. Song, R. Shibasaki, Online trajectory prediction for metropolitan scale mobility digital twin, in: Proc. of the 30th Intern. Conf. on Advances in Geographic Information Systems, ACM, Seattle Washington, 2022, pp. 1–12. URL: https://dl.acm.org/doi/10.1145/3557915.3561040. doi:10.1145/3557915.3561040.

[7] Y. Hong, H. Martin, M. Raubal, How do you go where?: improving next location prediction by learning travel mode information using transformers, in: Proc. of the 30th Intern. Conf. on Advances in Geographic Information Systems, ACM, Seattle Washington, 2022, pp. 1–10. URL: https://dl.acm.org/doi/10.1145/3557915.3560996. doi:10.1145/3557915.3560996.

[8] M. Musleh, Towards a unified deep model for trajectory analysis, in: Proc. of the 30th Intern. Conf. on Advances in Geographic Information Systems, ACM, Seattle Washington, 2022, pp. 1–2. URL: https:

//dl.acm.org/doi/10.1145/3557915.3565529. doi:10.1145/3557915.3565529.

[9] C. Yang, G. Gidofavi, Classification of regional dominant movement patterns in trajectories with a convolutional neural network, in: Martin Raubal, Shaowen Wang, Mengyu Guo, David Jonietz, Peter Kiefer (Eds.), Spatial Big Data and Machine Learning in GIScience, Melbourne, Australia, 2018.

[10] T. Yang, X. Wang, Z. Liu, Ship Type Recognition Based on Ship Navigating Trajectory and Convolutional Neural Network, Journal of Marine Science and Engineering 10 (2022) 84. URL: https://www.mdpi.com/2077-1312/10/1/84. doi:10.3390/jmse10010084.

[11] G. Buroni, G. Bontempi, K. Determe, A tutorial on network-wide multi-horizon traffic forecasting with deep learning., in: Workshop Proc. of the EDBT/ICDT 2021 Joint Conference, Nicosia, Cyprus, 2021. URL: https://ceur-ws.org/Vol-2841/BMDA_6.pdf.

[12] C. Wang, Y. Liang, G. Tan, Periodic residual learning for crowd flow forecasting, in: Proc. of the 30th Intern. Conf. on Advances in Geographic Information Systems, ACM, Seattle Washington, 2022, pp. 1–10. URL: https://dl.acm.org/doi/10.1145/3557915.3560947. doi:10.1145/3557915.3560947.

[13] Y. Zhang, Y. Li, X. Zhou, X. Kong, J. Luo, Curb-GAN: Conditional Urban Traffic Estimation through Spatio-Temporal Generative Adversarial Networks, in: Proc. of the 26th ACM SIGKDD Intern. Conf. on Knowledge Discovery & Data Mining, ACM, Virtual Event CA USA, 2020, pp. 842–852. URL: https://dl.acm.org/doi/10.1145/3394486.3403127. doi:10.1145/3394486.3403127.

[14] G. Mai, K. Janowicz, Y. Hu, S. Gao, B. Yan, R. Zhu, L. Cai, N. Lao, A review of location encoding for GeoAI: methods and applications, Intern. Journal of Geographical Information Science 36 (2022) 639–673. URL: https://www.tandfonline.com/doi/full/10.1080/13658816.2021.2004602. doi:10.1080/13658816.2021.2004602.

[15] M. Zhu, W. Chen, J. Xia, Y. Ma, Y. Zhang, Y. Luo, Z. Huang, L. Liu, Location2vec: A Situation-Aware Representation for Visual Exploration of Urban Locations, IEEE Transactions on Intelligent Transportation Systems 20 (2019) 3981–3990. URL: https://ieeexplore.ieee.org/document/8664648/. doi:10.1109/TITS.2019.2901117.

[16] B. Yan, K. Janowicz, G. Mai, S. Gao, From ITDL to Place2Vec: Reasoning About Place Type Similarity and Relatedness by Learning Embeddings From Augmented Spatial Contexts, in: Proc. of the 25th ACM SIGSPATIAL Intern. Conf. on Advances in Geographic Information Systems, ACM, Redondo Beach CA USA, 2017, pp. 1–10. URL: https://dl.acm.org/doi/10.1145/3139958.3140054. doi:10.1145/3139958.3140054.

[17] S. Feng, G. Cong, B. An, Y. M. Chee, POI2Vec: geographical latent representation for predicting future visitors, in: Proc. of the Thirty-First AAAI Conf. on Artificial Intelligence, 2017, pp. 102–108.

[18] A. A. Kashyap, S. Raviraj, A. Devarakonda, S. R. Nayak K, S. K V, S. J. Bhat, Traffic flow prediction models – A review of deep learning techniques, Cogent Engineering 9 (2022) 2010510. URL: https://www.tandfonline.com/doi/full/10.1080/23311916.2021.2010510. doi:10.1080/23311916.2021.2010510.

[19] S. Lyu, T. Han, Y. Nishiyama, K. Sezaki, T. Kusakabe, A plug-in memory network for trip purpose classification, in: Proc. of the 30th Intern. Conf. on Advances in Geographic Information Systems, ACM, Seattle Washington, 2022, pp. 1–12. URL: https://dl.acm.org/doi/10.1145/3557915.3560969. doi:10.1145/3557915.3560969.

[20] M. Tenzer, Z. Rasheed, K. Shafique, N. Vasconcelos, Meta-learning over time for destination prediction tasks, in: Proc. of the 30th Intern. Conf. on Advances in Geographic Information Systems, ACM, Seattle Washington, 2022, pp. 1–10. URL: https://dl.acm.org/doi/10.1145/3557915.3560980. doi:10.1145/3557915.3560980.

[21] H. Xue, B. P. Voutharoja, F. D. Salim, Leveraging language foundation models for human mobility forecasting, in: Proc. of the 30th Intern. Conf. on Advances in Geographic Information Systems, ACM, Seattle Washington, 2022, pp. 1–9. URL: https://dl.acm.org/doi/10.1145/3557915.3561026. doi:10.1145/3557915.3561026.

[22] L. Zhang, L. Zhao, D. Pfoser, Factorized deep generative models for end-to-end trajectory generation with spatiotemporal validity constraints, in: Proc. of the 30th Intern. Conf. on Advances in Geographic Information Systems, ACM, Seattle Washington, 2022, pp. 1–12. URL: https://dl.acm.org/doi/10.1145/3557915.3560994. doi:10.1145/3557915.3560994.

[23] L. Wang, M. Ciliberto, H. Gjoreski, P. Lago, K. Murao, T. Okita, D. Roggen, Locomotion and Transportation Mode Recognition from GPS and Radio Signals: Summary of SHL Challenge 2021, in: Adjunct Proc. of the 2021 ACM Intern. Joint Conf. on Pervasive and Ubiquitous Computing, ACM, Virtual USA, 2021, pp. 412–422. URL: https://dl.acm.org/doi/10.1145/3460418.3479373. doi:10.1145/3460418.3479373.

[24] Y. Lu, Learning to Transfer for Traffic Forecasting via Multi-task Learning (2021). URL: https://arxiv.org/abs/2111.15542. doi:10.48550/ARXIV.2111.15542.

[25] M. Liatsikou, S. Papadopoulos, L. Apostolidis,

I. Kompatsiaris, A Denoising Hybrid Model for Anomaly Detection in Trajectory Sequences., in: Workshop Proc. of the EDBT/ICDT 2021 Joint Conference, Nicosia, Cyprus, 2021. URL: https://ceur-ws.org/Vol-2841/BMDA_4.pdf.

[26] A. Tritsarolis, E. Chondrodima, P. Tampakis, A. Pikrakis, Online Co-movement Pattern Prediction in Mobility Data, in: Workshop Proc. of the EDBT/ICDT 2021 Joint Conference, Nicosia, Cyprus, 2021. URL: https://ceur-ws.org/Vol-2841/BMDA_15.pdf.

[27] L. Grinsztajn, E. Oyallon, G. Varoquaux, Why do tree-based models still outperform deep learning on tabular data?, 2022. URL: http://arxiv.org/abs/2207.08815.

[28] Y. Gao, Y. Chiang, X. Zhang, M. Zhang, Traffic volume prediction for scenic spots based on multi-source and heterogeneous data, Transactions in GIS 26 (2022) 2415–2439. URL: https://onlinelibrary.wiley.com/doi/10.1111/tgis.12975. doi:10.1111/tgis.12975.

[29] J. Rao, S. Gao, M. Li, Q. Huang, A privacy-preserving framework for location recommendation using decentralized collaborative machine learning, Transactions in GIS 25 (2021) 1153–1175. URL: https://onlinelibrary.wiley.com/doi/10.1111/tgis.12769. doi:10.1111/tgis.12769.

[30] B. J. Buijse, V. Reshadat, O. W. Enzing, A Deep Learning-Based Approach for Train Arrival Time Prediction, in: H. Y. et al. (Ed.), Intelligent Data Engineering and Automated Learning – IDEAL 2021, volume 13113, Springer, Cham, 2021, pp. 213–222. URL: https://link.springer.com/10.1007/978-3-030-91608-4_22. doi:10.1007/978-3-030-91608-4_22, series Title: Lecture Notes in Computer Science.

[31] A. Derrow-Pinion, J. She, D. Wong et al., ETA Prediction with Graph Neural Networks in Google Maps, in: Proc. of the 30th ACM Intern. Conf. on Information & Knowledge Management, ACM, Virtual Event Queensland Australia, 2021, pp. 3767–3776. URL: https://dl.acm.org/doi/10.1145/3459637.3481916. doi:10.1145/3459637.3481916.

[32] D. Wang, J. Zhang, W. Cao, J. Li, Y. Zheng, When Will You Arrive? Estimating Travel Time Based on Deep Neural Networks, Proc. of the AAAI Conference on Artificial Intelligence 32 (2018). URL: https://ojs.aaai.org/index.php/AAAI/article/view/11877. doi:10.1609/aaai.v32i1.11877.

[33] M. Li, S. Gao, F. Lu, K. Liu, H. Zhang, W. Tu, Prediction of human activity intensity using the interactions in physical and social spaces through graph convolutional networks, Intern. Journal of Geographical Information Science 35 (2021) 2489–2516. URL: https://www.tandfonline.com/doi/full/10.1080/13658816.2021.1912347. doi:10.1080/13658816.2021.1912347.

[34] F. Lippert, B. Kranstauber, P. D. Forré, E. E. van Loon, Learning to predict spatiotemporal movement dynamics from weather radar networks, Methods in Ecology and Evolution (2022) 2041–210X.14007. URL: https://onlinelibrary.wiley.com/doi/10.1111/2041-210X.14007. doi:10.1111/2041-210X.14007.

[35] M. Carroll, J. Lin, O. Paradise, R. Georgescu, M. Sun, D. Bignell, S. Milani, K. Hofmann, M. Hausknecht, A. Dragan, S. Devlin, Towards Flexible Inference in Sequential Decision Problems via Bidirectional Transformers, 2022. URL: http://arxiv.org/abs/2204.13326.

[36] S. Mehri, A. A. Alesheikh, A. Basiri, A Contextual Hybrid Model for Vessel Movement Prediction, IEEE Access 9 (2021) 45600–45613. URL: https://ieeexplore.ieee.org/document/9380635/. doi:10.1109/ACCESS.2021.3066463.

[37] Q. Gao, F. Zhou, G. Trajcevski, K. Zhang, T. Zhong, F. Zhang, Predicting Human Mobility via Variational Attention, in: The World Wide Web Conf. on - WWW '19, ACM Press, San Francisco, CA, USA, 2019, pp. 2750–2756. URL: http://dl.acm.org/citation.cfm?doid=3308558.3313610. doi:10.1145/3308558.3313610.

[38] J. Feng, Y. Li, C. Zhang, F. Sun, F. Meng, A. Guo, D. Jin, DeepMove: Predicting Human Mobility with Attentional Recurrent Networks, in: Proc. of the 2018 World Wide Web Conf. on World Wide Web - WWW '18, ACM Press, Lyon, France, 2018, pp. 1459–1468. URL: http://dl.acm.org/citation.cfm?doid=3178876.3186058. doi:10.1145/3178876.3186058.

[39] A. Li, S. Wang, W. Li, S. Liu, S. Zhang, Predicting Human Mobility with Federated Learning, in: Proc. of the 28th Intern. Conf. on Advances in Geographic Information Systems, ACM, Seattle WA USA, 2020, pp. 441–444. URL: https://dl.acm.org/doi/10.1145/3397536.3422270. doi:10.1145/3397536.3422270.

[40] D. Liao, W. Liu, Y. Zhong, J. Li, G. Wang, Predicting activity and location with multi-task context aware recurrent neural network, in: Proc. of the 27th Intern. Joint Conf. on Artificial Intelligence, 2018, pp. 3435–3441.

[41] Q. Liu, S. Wu, L. Wang, T. Tan, Predicting the next location: A recurrent model with spatial and temporal contexts, in: Thirtieth AAAI conference on artificial intelligence, 2016.

[42] D. Nguyen, R. Vadaine, G. Hajduch, R. Garello, R. Fablet, *GeoTrackNet* —A Maritime Anomaly Detector Using Probabilistic Neural Network Representation of AIS Tracks and *A Contrario* Detection, IEEE Transactions on Intelligent Transportation Systems

23 (2022) 5655–5667. URL: https://ieeexplore.ieee.org/document/9353410/. doi:10.1109/TITS.2021.3055614.

[43] S. K. Singh, J. S. Fowdur, J. Gawlikowski, D. Medina, Leveraging Graph and Deep Learning Uncertainties to Detect Anomalous Maritime Trajectories, IEEE Transactions on Intelligent Transportation Systems 23 (2022) 23488–23502. URL: https://ieeexplore.ieee.org/document/9839418/. doi:10.1109/TITS.2022.3190834.

[44] J. Rao, S. Gao, Y. Kang, Q. Huang, LSTM-TrajGAN: A Deep Learning Approach to Trajectory Privacy Protection, in: 11th Intern. Conf. on Geographic Information Science (GIScience 2021)-Part I, Poznan/online, 2020, pp. 12:1–12:17. URL: http://arxiv.org/abs/2006.10521.

[45] F. Simini, G. Barlacchi, M. Luca, L. Pappalardo, A Deep Gravity model for mobility flows generation, Nature Communications 12 (2021) 6576. URL: https://www.nature.com/articles/s41467-021-26752-4. doi:10.1038/s41467-021-26752-4.

[46] C. Cortes, X. Gonzalvo, V. Kuznetsov, M. Mohri, S. Yang, AdaNet: Adaptive Structural Learning of Artificial Neural Networks, in: D. Precup, Y. Teh (Eds.), Proc. of the 34th Intern. Conf. on Machine Learning, volume 70 of *Proc. of Machine Learning Research*, PMLR, 2017, pp. 874–883. URL: https://proceedings.mlr.press/v70/cortes17a.html.

[47] A. Desolneux, L. Moisan, J.-M. Morel, From gestalt theory to image analysis: a probabilistic approach, volume 34, Springer Science & Business Media, 2007.

[48] K. Janowicz, R. Zhu, J. Verstegen, G. McKenzie, B. Martins, L. Cai, Six GIScience Ideas That Must Die, AGILE: GIScience Series 3 (2022) 1–8. URL: https://agile-giss.copernicus.org/articles/3/7/2022/. doi:10.5194/agile-giss-3-7-2022.

Acknowledgments

This work is mainly funded by the EU's Horizon Europe research and innovation program under Grant No. 101070279 MobiSpaces and No. 101021797 STARLIGHT.

Glossary

- AE – Autoencoder
- AIS – Automatic Identification System
- CDR – Call Detail Records
- CNN – Convolutional Neural Network
- COG – Course over ground
- DNN – Deep Neural Network
- DL – Deep Learning
- GAN – Generative Adversarial Network
- GeoAI – Geospatial Artificial Intelligence
- GIS – Geographic Information Science
- GNN – Graph Neural Network
- GPS – Global Positioning System, often used synonymously for all GNSS (incl. Galileo, GLONASS, and Beidou)
- LSTM – Long Short-Term Memory
- MLP – Multilayer Perceptron
- NLP – Natural Language Processing
- OBU – On-board Unit
- OD – Origin-Destination
- POI – Point of Interest
- RNN – Recurrent Neural Network
- SAE – Stacked Autoencoder
- SAN – Self-Attention Network
- SOG – Speed over ground
- VAE – Variational AutoEncoder