Contents lists available at ScienceDirect

Journal of Cycling and Micromobility Research

journal homepage: www.sciencedirect.com/journal/journal-of-cycling-and-micromobility-research

Multi-operator free-floating GBFS trip destination prediction in public mobility sharing systems

Daniel Kerger *, Heiner Stuckenschmidt

Data and Web Science Group, University of Mannheim, B 6, 26, Mannheim, 68159, Baden-Württemberg, Germany

ARTICLE INFO

Keywords:
Shared mobility
Destination prediction
Machine learning
Urban mobility
XGBoost regression

ABSTRACT

Public mobility sharing systems are an important component of sustainable transport, particularly for last-mile travel. However, analysing trip patterns using open standards such as GBFS can be challenging due to vehicles frequently being assigned new identifiers and missing GPS trajectories, preventing a detailed tracking. To overcome this limitation, we present a machine learning pipeline that retrospectively predicts trip destinations within this circumstances—making it possible to partially recover travel patterns for GBFS data.

Our approach involves a three-step prediction pipeline: (1) candidate generation and reduction using spatial–temporal filtering; (2) multi-target regression via XGBoost to estimate destination coordinates; and (3) selection of the best-matching candidate. Our approach achieves an average accuracy of 77% across five German and 74% across five international cities within a tolerance of 500 metres. Compared to existing approaches, our method improves prediction accuracy by an average of 20% over methods that also do not use user-specific or GPS trajectory features.

These results demonstrate the feasibility of accurately predicting destinations in shared mobility despite rotating vehicle identifiers and missing trajectory data, thereby supporting improved system analysis and planning.

1. Introduction

Public mobility sharing systems, such as bike and e-scooter sharing, have become an integral part of sustainable urban transport, providing flexible, low-emission alternatives for short journeys and the final mile of a trip (Wang and Zhou, 2017; Huang and Xu, 2023). The widespread adoption of these systems has generated large volumes of publicly available data (Todd et al., 2021), enabling new opportunities for research into mobility behaviour (Wielinski et al., 2017; Si et al., 2020; Zhou, 2015), demand forecasting (Xu et al., 2018; Lin et al., 2018; Sathishkumar et al., 2020) and infrastructure planning (Félix et al., 2020; Griffin and Sener, 2016). In the context of rapid urbanisation and climate change, shared mobility services play a crucial role in reducing car dependency (Bissel and Becker, 2024; Vega-Gonzalo et al., 2024), lowering greenhouse gas emissions (Zhang and Mi, 2018) and improving access to transport in underserved areas (Lu et al., 2018). Their flexible deployment models, especially in free-floating systems without fixed stations, enable cities to adapt their infrastructure dynamically and support multimodal integration with public transit systems. Jaber et al. (2022), Tran et al. (2015)

Understanding how people use these services, particularly where trips begin and end, is essential for numerous applications. Urban planners require such insights for infrastructure development (e.g., the placement of bike lanes (Chahine et al., 2025) or stations (García-Palomares et al., 2012)), operators require them for fleet balancing and maintenance (De Chardon et al., 2016), and researchers depend on them for behavioural modelling and policy evaluation (Wielinski et al., 2017; Si et al., 2020; Zhou, 2015). However, analysing trip patterns from such systems remains a considerable challenge. Today, many cities and operators publish their data via open standards such as the General Bikeshare Feed Specification (GBFS) (MobilityData, 2025), which prioritises user privacy and data minimisation. Rotating vehicle identifiers, which are common in GBFS, prevent the linking of trip origins and destinations. The absence of continuous GPS trajectory data complicates or prevents the use of traditional methods for trajectory reconstruction and behavioural analysis.

While these limitations are essential from a data protection perspective, it comes at the cost of analytical granularity. Standard trip-based analysis techniques, which are commonly used in transportation research and data-driven urban analytics, rely on persistent identifiers to model demand Xu et al. (2018), Lin et al. (2018), Sathishkumar et al. (2020), infer behavioural patterns (Wielinski et al., 2017; Si et al.,

[*] Corresponding author.
E-mail addresses: daniel.kerger@students.uni-mannheim.de (D. Kerger), heiner.stuckenschmidt@uni-mannheim.de (H. Stuckenschmidt).

https://doi.org/10.1016/j.jcmr.2025.100105
Received 20 August 2025; Received in revised form 19 December 2025; Accepted 19 December 2025
Available online 31 December 2025

2020; Zhou, 2015), or detect anomalies (Liu et al., 2022a). Without such links, even basic tasks such as identifying the most common destinations or computing average trip distances become non-trivial. Moreover, advanced deep learning approaches — for instance, graph neural networks (GNNs) or convolutional neural networks (CNNs) combined with recurrent architectures such as long short-term memory networks (LSTMs) — typically rely on continuous GPS trajectory data to capture spatio-temporal dependencies and temporal variations in traffic flows. Since they are unavailable in GBFS, alternative approaches must be employed.

Therefore, our work addresses this issue by proposing a machine learning-based pipeline for predicting trip destinations retrospectively under the aforementioned constraints of limited trajectory data and rotating vehicle identifiers presented by GBFS. Unlike previous studies with the same GPS trajectory constraint, which often rely on persistent identifiers or user-specific features, such as travel history, our work focuses on a setting where only GBFS-compliant features are available and no prior knowledge of individual users is required. This complies with various data privacy regulations and enables the future adaptation of our pipeline for the nearly 1300 public sharing networks that already use GBFS today (MobilityData, 2025).

We present a three-step prediction pipeline that generates potential destinations based on spatio-temporal plausibility and system constraints, estimates their geographic coordinates using multi-target regression and selects a destination based on spatial proximity. To evaluate generalisation and real-world applicability, we conducted experiments using data from two public mobility operators across five German cities, spanning more than two years (April 2023 to May 2025), as well as five additional datasets from other cities around the world.

This study contributes to both machine learning in transportation science and the broader discourse on responsible data use in smart cities. It demonstrates that effective analytical methods can be developed even under privacy-preserving conditions, thereby aligning technical innovation with ethical data governance.

In summary, this paper makes the following contributions:

1. **First demonstration of trip reconstruction with rotating IDs within GBFS.** To the best of our knowledge, this is the first time that it has been demonstrated that it is possible to accurately reconstruct trips in GBFS datasets that rotate vehicle identifiers and do not provide GPS trajectories.
2. **Multi-operator and multi-city generalisation.** We demonstrate that our approach maintains strong predictive performance across operators and cities for future adaptations to other cities.
3. **Improved prediction accuracy with minimal data requirements.** Our approach outperforms existing approaches by an average of 20% when no user-specific data is used and no GPS trajectory data is available.
4. **Fully open-source pipeline.** We are releasing our full pipeline alongside a representative dataset under an Apache 2.0 licence, which will enable reproducibility and further research, available on GitHub.[1]

Our findings enable a wide range of analyses that were previously considered impossible on publicly available GBFS datasets. Accurate reconstruction of trip destinations makes it possible to conduct studies of traffic flow, demand distribution, vehicle rebalancing strategies and the spatial–temporal dynamics of shared mobility. Such analyses are central to transportation planning and policy, but were previously limited to datasets with persistent identifiers or privileged internal access. By making them possible using rotating public data, our work paves the way for more extensive and inclusive mobility research.

The rest of the paper is structured as follows: Section 2 discusses related work on trip destination prediction in both station-based and free-floating mobility systems and introduces the GBFS. Section 3 describes the methodology, including data preprocessing, the candidate filtering mechanism, and our predictive model. Section 4 presents empirical results and analyses model performance across operators and cities. Finally, Section 5 summarises key findings and outlines directions for future work.

2. Related work

Destination prediction is a well-studied area in contexts such as taxi services (Zhang et al., 2017; Yang et al., 2020-02; Song et al., 2020-09; Liao et al., 2022-05), car navigation (Tanaka et al., 2009; Terada et al., 2006), metro systems (Wu et al., 2025; Cheng et al., 2022; Liu et al., 2022b), railway (Noursalehi et al., 2021; Roos et al., 2016; Jiang et al., 2022), or multi-purpose services (Chen et al., 2010-12-01). Further studies conducted in the area of public mobility sharing systems, and existing ones are often limited to specific cities or datasets (Liu et al., 2024, 2019; Jiang et al., 2019; Du et al., 2018; Dai et al., 2018), that do not imply the limitation of GBFS. By contrast, Xu et al. (2022) were among the first to provide algorithms for dealing with different vehicle identifier types in the GBFS. However, they were unable to provide reliable algorithms for creating origin–destination pairs from rotating feeds; instead, they provided a list of vehicle rentals and returns.

In general, destination prediction in public mobility sharing systems can be categorised according to two main criteria: the type of network and the prediction approach. Networks can be station-based, in which trips start and end at predefined stations, or free-floating, in which trips can start and end at any location within a predefined area. Prediction approaches can be divided into classification tasks, where the aim is to predict predefined categories (e.g. stations or regions), and regression tasks, where the aim is to predict continuous values, such as geographical coordinates.

In addition to these main categorisations, there are other research areas related to predicting trip destinations: González and Melo-Riquelme (2016-05-01), Faghih-Imani and Eluru (2015) predicted the motivation behind bike-sharing trips using surveys, while Faghih-Imani and Eluru (2020-04-01) incorporated sociological data from the 2010 US Census to map rentals to social factors such as working, housing and shopping trips. Furthermore, Zhang and Yu (2016-11) discussed multi-rental trips incorporating lends and returns at multiple stations to allow longer routes on free plans.

2.1. Trip destination prediction in station-based networks

In station-based networks, predicting the destination of a journey is often framed as a classification task, where the aim is to predict either the destination station or a cluster of adjacent stations.

For example, Dai et al. (2018) used a Random Forest with a station clustering algorithm that incorporated both geographic proximity and usage patterns to achieve an accuracy of 0.39 in New York using Citi Bike data from the full year 2017. They also integrated weather data, such as temperature and precipitation, to improve prediction accuracy. A different approach from Du et al. (2018) framed the problem as a binary classification task, estimating the probability of a trip between two stations under given conditions. Using LightGBM on Beijing Mobike data from two weeks in 2017, they achieved an accuracy of 0.454 in correctly predicting the first-choice destination and relied on user data to refine training and prediction accuracy.

[1] The full pipeline alongside a representative dataset is available here on GitHub.

2.2. *Trip destination prediction in free-floating networks*

Free-floating networks present a more complex challenge than station-based networks because destinations are not limited to predefined stations. This makes the problem a regression task by definition, where the goal is to predict accurate geographic coordinates. It is common practice to cluster destinations into regions in order to reduce the complexity of the classification.

For example, Liu et al. (2019) used XGBoost to predict destination coordinates using geo-hashing, achieving an accuracy of 0.55 in Beijing using Mobike data from 2017. This approach relied on user-specific data, including the frequency of departures and arrivals at specific locations for each user and the distance between pick-up and drop-off points. While effective, this reliance on user-specific data limits the approach's applicability to non-user-specific scenarios. Similarly, Jiang et al. (2019) combined LSTM networks and CNNs to capture spatial and temporal dependencies, achieving an F1 score of 0.42. Their approach used user data to identify likely destinations and incorporated meteorological characteristics to improve performance. The study focused on the Beijing Mobike stationless bike-sharing system. Another study by Li and Shuai (2020a) shifted the focus from destination forecasting to demand forecasting by clustering data into space–time cubes (STCs) using GBFS data. This approach aggregated trips into predefined regions on a minute-by-minute basis. A similar approach was taken by Liu et al. (2024), who trained an XGBoost model to predict the destination of a trip within a 1000 × 1000 metre area in Beijing, achieving an accuracy of 0.56 using data from two weeks in May 2017. Later incorporating user-specific features increased their performance to an accuracy of 0.76.

2.3. *General bikeshare feed specification*

The General Bikeshare Feed Specification (GBFS) is a standard for publishing open data from bikeshare and scooter systems. It defines a set of JSON-based feeds that provide snapshots of the system, including static metadata (e.g., system_information), station-level availability (station_status), and free-floating vehicle availability (vehicle_status). These feeds are designed to reflect the state of the system at a specific point in time, enabling applications such as trip planning, real-time availability displays, and system monitoring. MobilityData (2025)

In contrast to other standards like GTFS (General Transit Feed Specification), which are event-driven and can represent continuous trajectories, delays, or trip-level updates, GBFS is limited to availability and system snapshots. This design reflects its focus on transparency for end users rather than operational analytics. Notably, GBFS employs rotating identifiers for vehicles, which supports privacy protection but limits longitudinal tracking. MobilityData (2025)

2.4. *Characteristics of trips in micromobility*

Several studies have examined trip characteristics of shared micromobility systems, including e-scooters and bicycles, across different cities. Table 1 summarises key reported metrics such as trip duration, distance, and speed.

Across the reviewed literature micromobility trips are consistently short in both time and distance. Average trip durations typically range between 5 and 20 min, with slightly longer trips reported in Asian cities (Cao et al., 2021) compared to North America. Mean travel distances commonly fall between 0.7 and 2.5 km, reflecting the local character and accessibility of shared micromobility systems.

Average travel speeds vary by vehicle type and environment, generally ranging from 6 to 8 km/h for e-scooters and around 7 to 15 km/h for bicycles. Urban conditions, infrastructure quality, and vehicle type largely explain these differences, with higher speeds typically observed for bicycles and in less congested settings.

Overall, the reported values indicate that micromobility serves primarily short-distance, low-speed trips that complement rather than replace traditional modes of transport. While factors like topography, infrastructure, and vehicle type cause some variation, consistent patterns emerge across the studies.

2.5. *Research gaps*

Despite advances in destination prediction, several critical gaps remain, particularly for free-floating networks with rotating vehicle IDs that do not incorporate GPS trajectories:

1. **Dependence on user-specific data:** Many existing studies, such as Liu et al. (2019), Jiang et al. (2019), Du et al. (2018), rely on user-specific data to improve prediction accuracy. For instance, Liu et al. (2019) used features such as the frequency of departures and arrivals at specific locations, while Jiang et al. (2019) incorporated the travel history of the users to identify the likely destinations. However, such data is unavailable in GBFS-compliant datasets, which limits the applicability of these methods.
2. **Prediction of Station Clusters:** Several approaches abstract the problem by either clustering sharing stations (Dai et al., 2018), or by clustering free-floating destinations into large regions or grids (e.g. STCs) (Liu et al., 2024). Although this reduces computational complexity and achieves higher performance, the use of large regions limits granularity, rendering these methods unsuitable for applications requiring high spatial accuracy.

3. Methodology

This section presents the methodology used to retrospectively predict trip destinations in public mobility sharing systems. This process was applied to data collected from public mobility sharing operators of all cities.

3.1. *Architectural overview*

Fig. 1 illustrates our pipeline process: We collected data from GBFS feeds, to represent the network's current status, including the location of vehicles not currently in use in the free-floating area, updated every three minutes.

Information on vehicles currently on trips is unavailable in GBFS; it is only available after the vehicles are returned. The full pseudocode for the three pipeline steps that follow is available in Section 1 of Appendix.

Candidate Generation & Reduction: First, we extract all rentals and returns within the defined start and end times from the raw GBFS feed, especially the free_bike_status (changed to vehicle_status in GBFS version 3) endpoint of each feed, and adding them to separate lists. An excerpt of the feed looks like the following: `{‘‘last_updated’’: 1755627017, ‘‘ttl’’:0, ‘‘version’’:‘‘2.3’’, ‘‘data’’:{‘‘bikes’’: [{‘‘bike_id’’:‘‘3d523af2’’, ‘‘lat’’:49.454114, ‘‘lon’’:8.484718, ‘‘current_range_meters’’:23200, ‘‘battery_level’’: 0.90}, . . . ]}}`

If a vehicle/bike ID disappears from the feed (i.e. if it was present in the previous frame but not the current one), then the vehicle was rented. Conversely, if a new vehicle/bike ID appears (i.e. it was not present in the previous frame but is present in the current one), then a vehicle has been returned. Each rental and return has the following features: *timestamp*, *latitude*, *longitude*, *battery_level*, and *current_range_meters*. Using the rentals and returns, we create a set of destination candidates for each rental. The initial set of destination candidates contains all returns conducted after the rental. As this set may include hundreds or thousands of candidates, we reduce the number

Table 1
Overview of Average Reported Trip Characteristics in Micromobility Studies. Trips made using micromobility vehicles (bikes and e-scooters) tend to be short in distance, duration, and average speed.

Study	City (Country)	Duration [min]	Distance [m]	Speed [km/h]	Type
McKenzie (2020)	Washington, D.C. (USA)	4.8–14.6	874–2382	9.8–11[a]	E-Scooter/E-Bike
Bai et al. (2021)	Austin (USA)	6	1100	5	E-Scooter
Bai and Jiao (2020)	Minneapolis (USA)	19	1300	6	E-Scooter
Cao et al. (2021)	Singapore (Singapore)	21.5	2130	5.9[a]	E-Scooter
Eriksson et al. (2019)	Stockholm (Sweden)	–	–	15	Bike
Strauss and Miranda-Moreno (2017)	Montreal (Canada)	–	–	20	Bike
Lin et al. (2008)	Kunming (China)	–	–	14.8	E-Bike
Heumann et al. (2025)	Berlin (Germany)	17.6[a]	2202	7.5	Bike
	Berlin (Germany)	14.7[a]	1662	6.8	E-Scooter
Mavrogenidou and Polydoropoulou (2025)	Agia Paraskevi (Greece)	9.3	1150	7.5–8.5	E-Scooter
Reck et al. (2021)	Zurich (Switzerland)	–	730	–	E-Scooter
	Zurich (Switzerland)	–	1292	–	Bike
	Zurich (Switzerland)	–	1595	–	E-Bike

[a] Calculated from other available metrics.

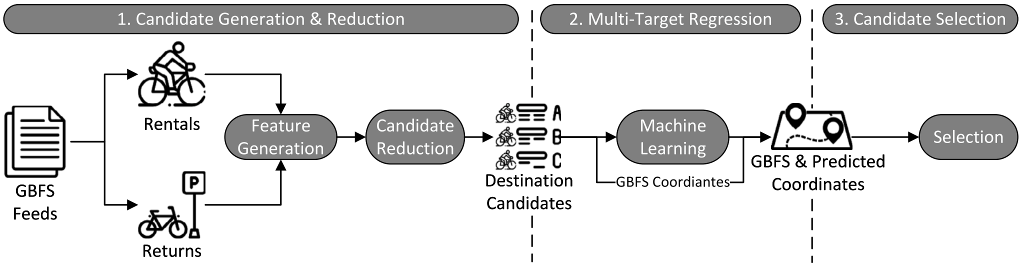

Fig. 1. Flow chart of the destination prediction pipeline. The process involves candidate generation and reduction using spatial–temporal filtering and multi-target regression via XGBoost to estimate destination coordinates. The final step is to select the closest matching candidate from the reduced set.

by applying a set of conditions. These conditions are categorised using statistics from over 850,000 past trips in the training sets of the cities of Mannheim, Ludwigshafen, Heidelberg, Karlsruhe, and Stuttgart, and are in line with the findings presented in Table 1. Later, they are transferred on the international cities Shanghai, London, New York, Montreal, and San Francisco. These include constraints such as a maximum travel speed of 20 km/h, a maximum travelled distance of 3000 metres, and a maximum trip length of 30 min. The reason why these three thresholds were set as constraints is explained in Section 3.2. It is possible that there may be no more candidates for a trip, resulting in removal. The final result of this step is a reduced set of assignable destination candidates for each rental.

Multi-Target Regression: In the second step, we use these sets of rental and return data to extract features for the subsequent machine learning model. For each rental and destination candidate in the set, we use features such as the time difference between rental and return, change in battery level, average speed and travel distance from the candidate reduction. We use the Open Source Routing Machine (OSRM) (Luxen and Vetter, 2011) to calculate travel distances. This enables us to plan routes using the city's bike network. By default, we use the shortest path, assuming that this is the route chosen by the user. However, to account for cases where the user does not choose the shortest path, it is possible to set an adjustable threshold to include longer routes in the distance calculation. If there is no bike geodata available for OSRM, we use the shortest route on the road network or, as a last resort, the air distance. We then use a multi-target regression model to predict the return coordinates for each destination candidate in the set. By the end of this step, we have a pair of predicted and extracted coordinates for each candidate.

Candidate Selection: In the final step, we use an selection algorithm to select one potential final destination for each rental.

To accomplish this, we use the pairs of coordinates from the previous step to select one candidate for each rental, as illustrated in Fig. 2. First, we calculate the distance between the GBFS-extracted and predicted coordinates for each candidate in the set of destination candidates. This distance is used as a metric of similarity to determine how feasible a trip is under the given constraints. Then, we select the candidate with the shortest distance as the final destination candidate for the rental. All others are disregarded. If two candidates have the same distance between their predicted and extracted coordinates, one is randomly selected. This did not happen for any of our over 2.7 million trips.

For performance reporting, we distinguish between true performance and the use of an additional tolerance radius, as in previous publications. True performance only considers correct matches between a rental and a return. If the selected candidate does not match the correct return, it is treated as a false prediction. This ensures that all other GBFS fields, in addition to the location, are correct. The tolerance radius considers the final candidate selection to be correct if the predicted location is within a defined threshold of the actual return position of the rental. Therefore, we calculate the airline distance between the predicted coordinates of the final destination candidate and the actual return coordinates. If this distance is within the threshold, the selection is considered correct.

3.2. Dataset

The main datasets, labelled $O1$ and $O2$ (Operator 1 and Operator 2), used in this study were collected from two public mobility sharing operators GBFS feeds in version 2.3 and version 3.0 in Germany and cover the cities of Mannheim, Ludwigshafen, Heidelberg, Karlsruhe and Stuttgart. We also used non-GBFS public international datasets from the cities of Shanghai (Heywhale, 2020), London (Au, 2019), New York (Citi, 2024), Montreal (Sigouin, 2017), and San Francisco (Hamner, 2019), which we have labelled $M1$ (Miscellaneous 1). Table 2 provides an overview of the operators and their data characteristics.

The datasets contain the features *vehicle_id*, *timestamp*, *latitude*, *longitude*, *battery_level* in percent, and *current_range_meters* in metres from the GBFS feeds. Since GBFS does not provide GPS trajectories, only the start and end locations of a trip can be used for later model training

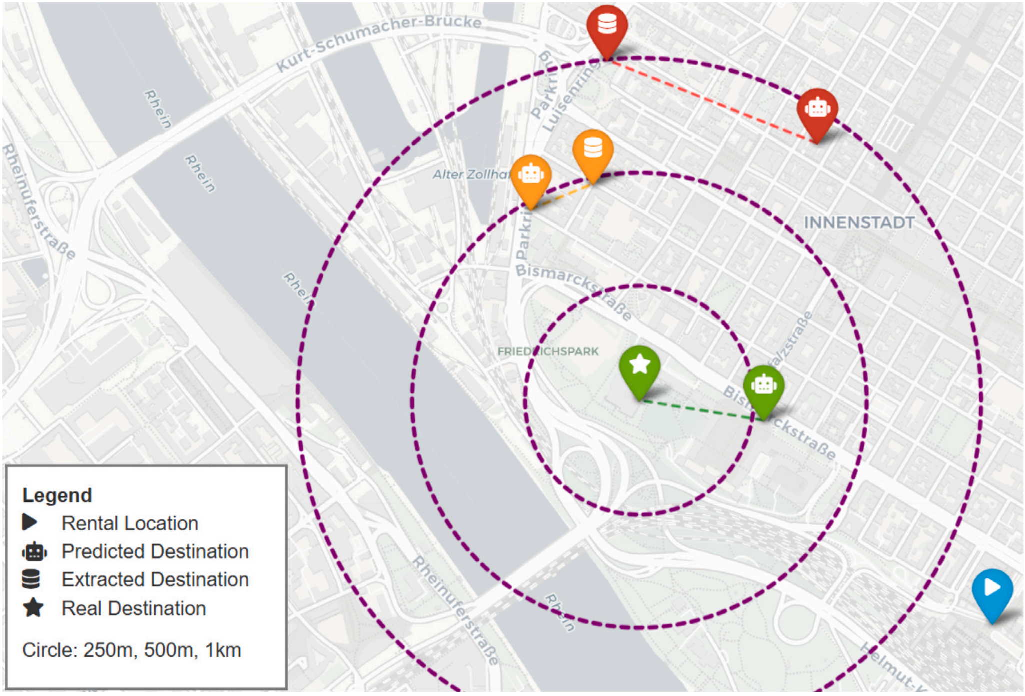

Fig. 2. Candidate selection process: In this example, the predicted candidate in yellow would be selected as the final destination for the starting location (blue), as the distance between the extracted GBFS and its predicted coordinates is the shortest compared to the other two pairs (the red pair has the longest distance, followed by the green pair). However, this would be a false prediction, as the correct candidate is the green pair containing the real extracted destination (green star). When the tolerance radius is incorporated at a resolution of 500 metres (the middle purple circle), the prediction is classified as correct because the distance to the real destination is within this later selected threshold.

Table 2
An overview of the public sharing operators used in the study. We used three different datasets, two of which were from German cities. The third dataset is publicly available to enable better performance comparisons and reproducibility.

Dataset	City	Missing features	Time frame
$O1$	Mannheim	–	31.08.2023–27.10.2024
	Ludwigshafen	–	31.08.2023–27.10.2024
	Heidelberg	–	31.08.2023–27.10.2024
	Karlsruhe	–	31.08.2023–27.10.2024
	Stuttgart	–	31.08.2023–27.10.2024
$O2$	Mannheim	*battery_level*	21.01.2025–01.05.2025
	Karlsruhe	*battery_level*	21.01.2025–01.05.2025
	Stuttgart	*battery_level*	21.01.2025–01.05.2025
$M1$	Shanghai	*battery_level*, *current_range_meters*	01.08.2016–31.08.2016
	London	*battery_level*, *current_range_meters*	01.08.2017–31.08.2017
	New York	*battery_level*, *current_range_meters*	01.01.2017–30.01.2017
	Montreal	*battery_level*, *current_range_meters*	01.05.2017–31.05.2017
	San Francisco	*battery_level*, *current_range_meters*	01.08.2013–30.11.2013

regarding geographical location. For datasets $O1$ and $O2$, we were able to reconstruct all trips using either a non-rotating *vehicle_id* or another field which does not rotate during rental and return from the GBFS feed. The dataset $M1$ was built using non-GBFS sources and also contains a static *vehicle_id*. The *vehicle_id* is solely used for analysis, not for destination matching. To enhance trip analysis, additional features were derived, including *time_diff* (rental-return time difference), *battery_diff* (battery level change), *range_diff* (range change), *distance* (road path via OpenStreetMap using OSRM routing), *mean_speed_distance* (average speed based on distance), and *mean_speed_range* (average speed based on range). As not all of these features are mandatory in the GBFS, the dataset $O2$ does not have a valid *battery_level*, and the dataset $M1$ additionally lacks *current_range_meters*. A performance decrease is to be expected.

Before training, the data of $O1$ and $M1$ were cleaned to exclude data points which are assumed to belong to manual redistribution or repairs to keep the focus on the last-mile trips. To select suitable thresholds for filtering, we examined the distribution of the features *time_diff*, *mean_speed_distance*, *mean_speed_range*, and *range_diff*, shown in Fig. 3.

Trips were removed if the battery was charged during the journey, if the time difference exceeded 30 min, if the travel speed exceeded 20 km/h, or if the travel distance exceeded 3000 metres. The speed threshold is particularly significant, as e-scooters are only permitted to travel at a maximum speed of 20 km/h according to German law. All of the above mentioned thresholds are further above the averages of the reviewed literature in Table 1. It is important that this strict form of data cleaning is done only in the model training process. Similar reduction rules are applied in the subsequent testing with the data pipeline, but to destination candidates rather than preformed trips.

Rather than using the standard train-test split of e.g. 80/20, we introduce a more reliable, stratified split. We select one week from each month as the test set and use the remaining weeks for training. This ensures that weekly and seasonal influences are reflected in both the training and test sets in a similar way. As $O2$ and $M1$ are only used

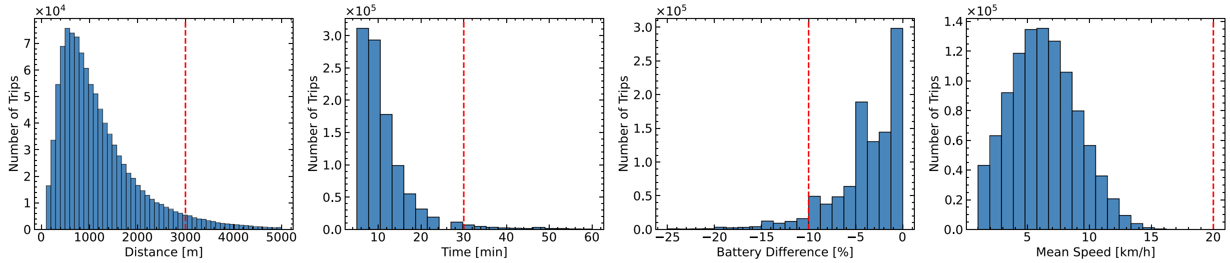

Fig. 3. Data distributions of the four features distance, time, battery difference, and mean speed of the trips in the datasets, with the selected threshold for the future reduction step of candidates. The data is taken from training sets for the cities of Mannheim, Ludwigshafen, Heidelberg, Karlsruhe and Stuttgart, ensuring that no data leakage into the test set is done.

Table 3
Training and test set trip distribution by city for $O1$, $O2$ and $M1$. Our datasets contain a total of approximately 2.7 million trips from ten different cities. Datasets $O1$ and $M1$ are split for training and testing purposes.

Dataset	Type	Cities				
		Mannheim	Ludwigshafen	Heidelberg	Karlsruhe	Stuttgart
$O1$	Train	307,771	63,588	137,193	63,789	287,376
	Test	99,872	21,236	44,610	18,910	91,831
$O2$		Mannheim	–	–	Karlsruhe	Stuttgart
	Test	59,560	–	–	288,769	340,115
$M1$		Shanghai	London	New York	Montreal	San Francisco
	Train	48,611	154,892	406,507	320,016	54,611
	Test	50,638	135,475	292,007	249,839	23,522

for testing, they are omitted from the rest of this section and will be referenced again in Section 4. No new model training is needed for $O2$ as the city models are trained on $O1$. Therefore, we are not doing a data split for $O2$, but we are doing one for $M1$. Model training is still required for $M1$, since no data on the cities is available in another dataset. This is done in the same way as for dataset $O1$. A detailed breakdown of the training and test trip sets by city can be found in Table 3.

3.3. Trip subset generation & candidate reduction

In the first step of our pipeline, we generate a subset of plausible destinations for each rental based on a set of reduction rules to address the challenge of forming trips with rotating vehicle IDs. The thresholds of the rules are based on findings from the literature and the characteristics of trips that were previously introduced in our dataset. This subset, in the following known as 'destination candidates', is created by iterating through all rentals and adding potential returns that satisfy the following conditions:

- $R1$: The return time is after the rental time.
- $R2$: The return time is within 30 min of the rental time.
- $R3$: The return battery level is less than the rental battery level.
- $R4$: The battery difference is less than 10%.
- $R5$: The mean speed is less than 20 km/h.
- $R6$: The distance on the bike-network (OpenStreetMap, 2025) is less than 3000 m.

In addition, if a rental and a return are identified as a perfect match, defined as a time difference of six minutes or less, a range difference of zero, a battery difference of zero, and a distance of less than 1000 metres, the corresponding return candidate is removed from all other pairings. This rule is denoted $R7$. The thresholds for rules $R2$, $R4$ and $R6$ were determined through data exploration to ensure the removal of implausible trips. We evaluated the effectiveness of each rule based on:

- The number of correct and wrong eliminated destination candidates.
- The number of still assignable trips (the correct return remains in the candidate set)
- The number of non-assignable trips (the correct return is removed from the candidate set).

As shown in Fig. 4, when comparing the relative number of correct and incorrect destination candidates before and after candidate reduction, the number of correct candidates remained above 80% for all cities except Ludwigshafen. Meanwhile, we were able to reduce the number of incorrect candidates to around 10% for each city. We eliminated any influence introduced by the reduction order by permuting the rule orders, which resulted in no difference. Therefore, the most effective reduction rules are $R1 - R3$, which filter twice for time, and battery level. This reduced up to 70% of all incorrect destination candidates for all cities except Karlsruhe, where the reduction was 30%. An important finding is that, since the battery level is one of the most important feature for candidate reduction, a performance decrease for $O2$ and $M1$ can be expected, since this feature is missing there.

The reduction process introduces an inherent error rate, meaning that some trips will have no correct candidate destinations after reduction and will therefore be unassignable. The error introduced by that is referred to as the minimal error rate and is shown in Table 4 for each city for $O1$, $O2$, and $M1$. For the $O1$ test set, all cities show that 82%–86% of all trips are still assignable. The $O2$ test set includes fewer cities, but the share of trip origins with assignable destination candidates is higher, between 87% and 91%, indicating better coverage or predictability in this dataset. Finally, the $M1$ test set has cities with a high percentage (87% to 95%) of trips with assignable destinations, while London and Montreal have lower percentages (69% and 78%).

3.4. Destination prediction model

Next, we introduce the machine learning models that will be used in the pipeline later on. Several types of machine learning regression model were trained for each city using the features stated in Table 5.

The target variables are a new coordinate pair consisting of a latitude and longitude value in the same format as the two input variables used by the model. If a model does not support multi-target regression,

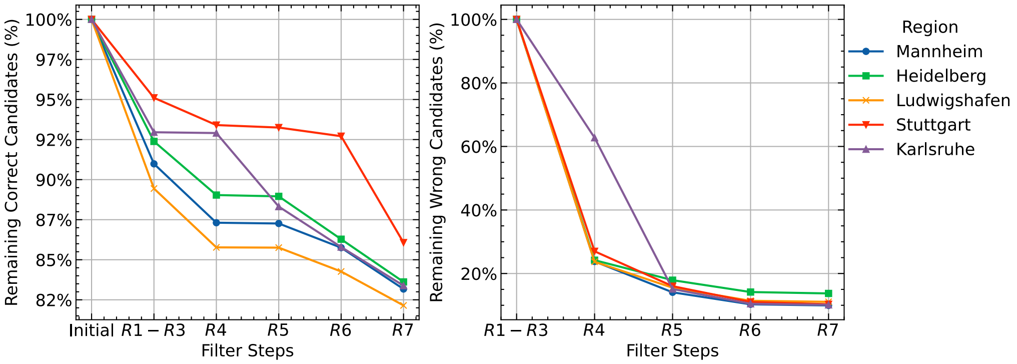

Fig. 4. The impact of different reduction rules on the relative number of correct (left) and incorrect (right) destination candidates in the $O1$ test set. During the candidate reduction process, we observed that the number of incorrect destination candidates decreased significantly, whereas the number of correct candidates decreased slightly.

Table 4
Assignable and non-assignable trips in the test sets. In most cities, around 80% of trips can be assigned to the correct destination, while the remaining 20% introduce minimal estimated error.

Set	Assignable	Cities				
		Mannheim	Ludwigshafen	Heidelberg	Karlsruhe	Stuttgart
$O1$	Yes	82, 894 (83%)	17, 414 (82%)	37, 027 (83%)	15, 696 (83%)	78, 975 (86%)
	No	16, 978 (17%)	3, 822 (18%)	7, 583 (17%)	3, 214 (17%)	12, 856 (14%)
		Mannheim	–	–	Karlsruhe	Stuttgart
$O2$	Yes	54, 432 (91%)	–	–	254, 116 (87%)	299, 301 (88%)
	No	5, 128 (9%)	–	–	34, 653 (13%)	40, 814 (12%)
		Shanghai	London	New York	Montreal	San Francisco
$M1$	Yes	46, 080 (87%)	93, 477 (69%)	254, 046 (87%)	194, 874 (78%)	19, 993 (95%)
	No	4, 558 (13%)	41, 998 (31%)	37, 961 (13%)	215, 186 (22%)	3, 529 (5%)

Table 5
Features used for the trip destination prediction model.

Feature	Type	Description
lat_lend	Float	Start coordinate's latitude of the vehicle.
lng_lend	Float	Start coordinate's longitude of the vehicle.
time_diff	Integer	Time difference between rent and return.
battery_diff	Integer	Battery difference between rent and return.
range_diff	Integer	Range difference between rent and return.
distance	Float	Route distance between rent and return.
mean_speed_distance_based	Float	Mean Speed of the trip based on distance.
mean_speed_range_based	Float	Mean speed of the trip based on range.

we utilise MultiOutputRegressor from scikit-learn (Pedregosa et al., 2011).

Since GBFS does not provide GPS trajectories (MobilityData, 2025), we cannot adapt deep learning techniques that frequently require these features to our exact problem statement. These include approaches, using hybrid deep learning approaches, such as graph convolutional neural networks (GCNs), used by Zhang et al. (2025) for traffic flow prediction, or a combination of recurrent neural networks (RNNs) and convolutional neural networks (CNNs), such as proposed by Zhao et al. (2021) for taxi or Li and Shuai (2020b), and Miao et al. (2022) for bicycle destination prediction.

Unlike other related works involving CNNs from other modes of transport (Wu et al., 2025; Noursalehi et al., 2021), an origin–destination (O–D) matrix was not adopted here. While O–D matrices capture aggregate traffic demand between areas and work well for origin–destination demand prediction, using them to predict the exact location of a returning trip is insufficient, as they ignore contextual factors such as battery state, weather, and temporal dynamics, and even probabilistic approaches would yield identical predictions for each timestamp within a region. Machine learning models are therefore required to capture these nuances and generate accurate, context-sensitive destination predictions.

In the performance evaluation, several common metrics from the field of machine learning are utilised. The evaluation of internal models is conducted through the utilisation of the mean square error (MSE) (Eq. (1)), the mean absolute error (MAE) (Eq. (2)), and the R^2 (Eq. (3)) coefficient of determination. We use their respective implementations from scikit-learn (Pedregosa et al., 2011), which are defined as follows:

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

Table 6
Performance comparison for Mannheim on O1. We tested multiple models using the Mannheim dataset and found that the XGBoost model produced the best results.

Model type	Multi-Target	MSE	R^2	MAE
XGBoost	Yes	4.01×10^{-5}	0.92	0.004
LightGBM	No	4.63×10^{-5}	0.91	0.004
Catboost	Yes	4.7×10^{-5}	0.91	0.004
Linear Regression	No	7.01×10^{-5}	0.869	0.006
Ridge Regression	No	7.01×10^{-5}	0.869	0.006
Random Forest	Yes	5.3×10^{-4}	0.84	0.015
Support Vector Machine	No	0.001	−2.12	0.034

$$\mathrm{MAE}(y, \hat{y}) = \frac{1}{n} \sum_{i=1}^{n} \left| y_i - \hat{y}_i \right| \tag{2}$$

$$R^2(y, \hat{y}) = 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2} \tag{3}$$

Further, in the subsequent chapter we introduce the metrics root mean squared error (RMSE) (Eq. (4)), precision (Eq. (5)), and accuracy (Eq. (6)), also with their implementations from scikit-learn (Pedregosa et al., 2011), which are defined as follows, where TP are true positives, TN are true negatives, FP area false positives, and FN are false negatives.

$$\mathrm{RMSE}(y, \hat{y}) = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2} \tag{4}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{5}$$

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{6}$$

As shown in Table 6, the XGBoost model, trained as a multi-target regressor to predict destination latitude and longitude, outperformed all other models. It achieved a MSE of $4.01e^{-5}$ and a R^2 of 0.92. We also tested it with additional time-related features, such as day of the week and time of day, in both single-feature and one-hot encoded forms, but found no improvement in performance. All features except *lat_lend* and *lng_lend* were normalised using scikit-learn's StandardScaler (Pedregosa et al., 2011). *lat_lend* and *lng_lend* were not normalised to preserve their absolute geographic values, which are essential for accurately representing spatial relationships. All models, LightGBM, CatBoost, Linear Regression, Ridge Regression, and Random Forest performed well, whereas the Support Vector Machine underperformed, likely due to the unscaled spatial features and targets hindering the model's ability to capture underlying patterns. Despite further hyperparameter tuning, no significant performance enhancement was achieved.

We decided against incorporating deep learning models, such as deep neural networks (DNNs), into our research. Our XGBoost model already achieved a high level of performance. Given these results, the potential performance gains from deploying a more computationally expensive deep learning approach would likely be marginal and not justify the considerable training and deployment costs. Instead, we focused on traditional machine learning methods, which offered a more efficient balance between accuracy, interpretability, and resource consumption.

All models were validated using tenfold cross-validation to ensure stable performance. Given these results, further training and optimisation were performed using XGBoost, as it had consistently outperformed the other models in all cities (see Table 7).

Extended training and optimisation were performed for each city using the aforementioned features. Hyperparameter tuning for *n_estimators*, *max_depth*, *learning_rate*, *subsample*, and *colsample_bytree* in the XGBoost model was performed using Optuna (Akiba et al., 2019) to minimise the mean squared error (MSE). The spaces were adjusted several times during training when upper or lower bounds were reached. The random seed was set to 42 for all cities to ensure reproducibility.

The optimisation process was performed over 500 trials, with tenfold cross-validation performed on each trial to evaluate performance. Fig. 5 shows the convergence of MSE and R^2, which stabilised after fewer than 500 trials, indicating that near-optimal hyperparameter values were achieved. For the $M1$ hyperparameter, see Table A.12.

Taking into account the importance of the features, the geographic features, specifically *lat_lend* and *lng_lend*, consistently emerge as the most important predictors of destinations in all cities, dominating both weight and gain metrics as shown in Fig. 6. Weight measures how frequently a feature is used in the model's splits, while gain quantifies the improvement in prediction accuracy attributed to that feature (Pedregosa et al., 2011). In Mannheim, these features contribute 26% and 23% in weight and 36% and 43% in gain. Similarly, in Karlsruhe, *lat_lend* and *lng_lend* dominate the gain-based importance, contributing over 65% together. The feature *distance* is another key factor, ranking just below the geographical coordinates in importance, with a weight contribution of approximately 15–18% between cities. Features such as *battery_diff*, *speed_distance*, and *time_diff* are of moderate importance, while *range_diff* and *speed_range* are consistently among the least influential features, contributing minimally to the model's predictive power.

4. Results and discussion

This section presents an evaluation of destination prediction. We analyse the training performance of the multi-target regression model, both on its own and in combination with candidate reduction and final candidate selection. Additionally, we compare the results with those of state-of-the-art methods, discuss an inter-city transfer learning approach, and possible security implications for the GBFS.

4.1. Multi-target regression model training performance

Table 8 shows the performance of the trip destination prediction models trained on the $O1$ and $M1$ datasets. No new models were trained for $O2$ as the models from $O1$ were used later.

Most cities in the $O1$ dataset — including Mannheim, Ludwigshafen, Heidelberg and Karlsruhe — demonstrate high predictive accuracy, with low RMSE, MSE and MAE values, and high R^2 scores ranging from 0.75 to 0.92. However, Stuttgart stands out as an outlier, with higher errors (an RMSE of 1.029 and an MSE of 1.05989) and a lower R^2 of 0.21, indicating poorer model performance there.

For the $M1$ dataset, performance varies more widely between cities. San Francisco and Shanghai achieved very low RMSE values (0.001 and 0.011, respectively) and high R^2 values (0.99 and 0.95, respectively), indicating accurate predictions. New York, Montreal and London have moderate results; however, London has the lowest R^2 (0.76) and higher error values among them, indicating weaker performance. Distance error metrics also follow this trend, with San Francisco showing minimal median and mean distance errors, while other cities have higher values. Overall, the $M1$ dataset results reveal greater variability due to the different time frame sizes and the absence of the *battery_level* and *current_range_meters* attributes.

4.2. Performance with candidate reduction and destination selection

In addition to the previously introduced model performance, the effect of using the entire pipeline on the dataset is examined. For performance evaluation, we rely solely on the accuracy metric, as there are only two possibilities: a correct or false selection of the destination candidate. This does not allow us to report other metrics, such as recall, precision or the F1 score. We also differentiate between various levels of resolution, ranging from 11 metres (true performance), where a prediction is only considered correct if the correct destination candidate is selected from the set, to 1000 metres, where the selected destination candidate must be within this tolerance radius to be considered correct.

Table 7
Optimal hyperparameters for XGBoost models by city for $O1$. We used Optima to optimise the most important XGBoost model hyperparameter to improve performance.

Parameter	Mannheim	Ludwigshafen	Heidelberg	Karlsruhe	Stuttgart	Range
n_estimators	1148	843	435	682	126	50–1, 500
max_depth	16	15	15	14	4	3–20
learning_rate	0.01	0.01	0.02	0.01	0.06	0.01–0.1
subsample	0.96	0.97	0.90	0.94	0.69	0.5–1.0
colsample_bytree	0.89	0.76	0.99	0.80	0.81	0.5–1.0
random_seed	42	42	42	42	42	42

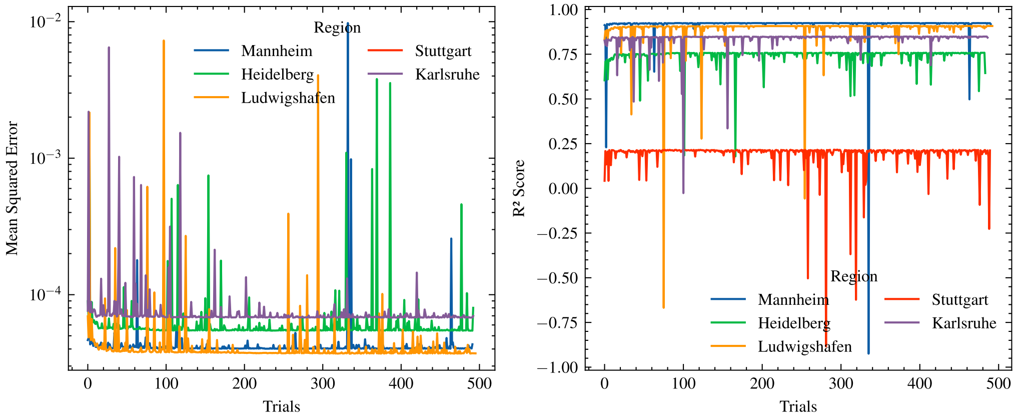

Fig. 5. The figure shows the convergence of MSE (left) and R^2 (right) during Optuna hyperparameter tuning for $O1$. The MSE and R^2 converge after fewer than 500 trials, indicating that near-optimal hyperparameter values were achieved.

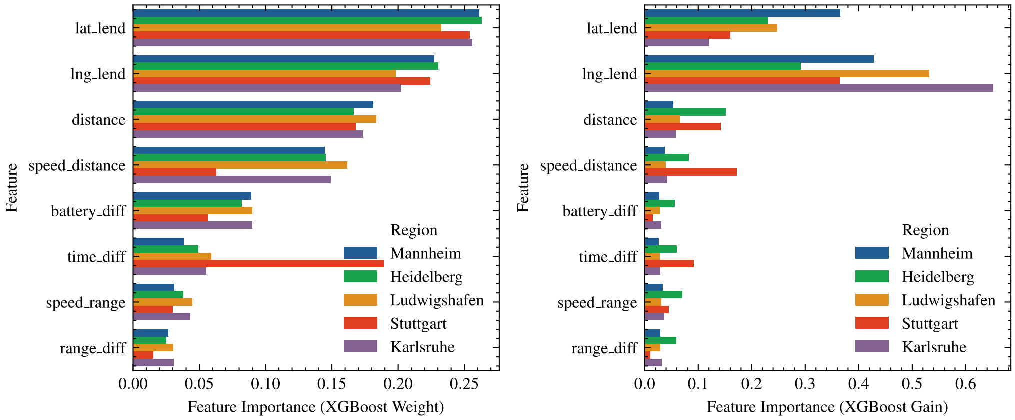

Fig. 6. Feature Importance for Destination Prediction by XGBoost Weight (left) and Gain (right) for $O1$. *lat_lend* and *lng_lend*, consistently emerge as the most important predictors of destinations in all cities.

Table 8
Trip Destination Prediction Performance on the Test Set of $O1$ and $M1$. Most cities demonstrate high predictive accuracy, with low RMSE, MSE, and MAE values and high R^2 scores.

Set	Metric	Cities				
		Mannheim	Ludwigshafen	Heidelberg	Karlsruhe	Stuttgart
$O1$	RMSE	0.006	0.006	0.007	0.008	1.029
	MSE	$4.01e^{-5}$	$3.69e^{-5}$	$5.43e^{-5}$	$6.78e^{-5}$	1.05989
	MAE	0.004	0.003	0.004	0.005	0.021
	R^2	0.92	0.90	0.75	0.84	0.21
	$\tilde{d}$	446 m	367 m	493 m	601 m	893 m
	$\bar{d}$	582 m	521 m	664 m	744 m	3, 472 m
		Shanghai	London	New York	Montreal	San Francisco
$M1$	RMSE	0.011	0.015	0.007	0.009	0.001
	MSE	0.0001	0.002	$6.23e^{-5}$	$9.40e^{-5}$	$1.08e^{-5}$
	MAE	0.008	0.009	0.004	0.005	0.0003
	R^2	0.95	0.76	0.84	0.77	0.99
	$\tilde{d}$	1045 m	902 m	300 m	344 m	26 m
	$\bar{d}$	1336 m	1259 m	649 m	763 m	60 m

Table 9
Accuracy by distance threshold and region on the test sets. Performance increases as resolution decreases, ranging from 0.49 to 0.77 for true performance in $O1$, with a slight decrease in $O2$ and high variation in $M1$ due to missing features.

Set	Resolution	Cities				
		Mannheim	Heidelberg	Ludwigshafen	Karlsruhe	Stuttgart
$O1$	@11 m	0.61	0.56	0.77	0.60	0.49
	@250 m	0.70	0.66	0.81	0.66	0.56
	@500 m	0.81	0.79	0.86	0.76	0.69
	@1000 m	0.94	0.94	0.94	0.92	0.88
		Mannheim	–	–	Karlsruhe	Stuttgart
$O2$	@11 m	0.42	–	–	0.18	0.23
	@250 m	0.59	–	–	0.62	0.44
	@500 m	0.75	–	–	0.89	0.67
	@1000 m	0.89	–	–	0.99	0.92
		Shanghai	London	New York	Montreal	San Francisco
$M1$	@11 m	0.34	0.11	0.17	0.15	0.70
	@250 m	0.42	0.37	0.53	0.47	0.79
	@500 m	0.58	0.66	0.81	0.76	0.89
	@1000 m	0.85	0.90	0.95	0.93	0.97

We determined that a region of 500 metres offered the best trade-off between high accuracy and reasonable resolution, in line with state-of-the-art performance. Further testing of other resolutions indicates that accuracy converges after 1000 metres, as shown in Fig. A.7.

Table 9 shows the detailed accuracy metrics for the $O1$, $O2$ and $M1$ test sets. Generally, performance improves as resolution decreases for all test sets. The mean accuracy for the $O1$ test set is 0.60 for true performance and 0.78 at 500 metres. For $O2$, true performance decreases by an average of 0.29 and remains steady at 500 metres. This decrease in performance was to be expected, as the battery consumption feature is unavailable in the dataset, resulting in larger sets of potential destinations with over 100 candidates. For the $M1$ test set, the accuracy values are lower for true performance, with London exhibiting the lowest resolution at 0.11. However, accuracy improves as the distance threshold increases, reaching values of up to 0.89 for San Francisco at 500 metres. The lower overall true performance on $M1$ is related to the missing features *battery_level* and *current_range_meters*, which impacted the candidate reduction step of the solution.

4.3. Comparison with state of the art

We compared our models with several state-of-the-art approaches, including Liu et al. (2024), Liu et al. (2019), Jiang et al. (2019), Du et al. (2018), Jiang et al. (2021), and Dai et al. (2018). As far as we are aware, there are no other works that provide a pipeline or baseline tailored to GBFS. Therefore, we compare our pipeline with models in the field of trip destination prediction in public sharing systems. Most of the works we compared relied on the 2017 Beijing dataset. However, this dataset is no longer publicly accessible and could not be obtained upon request. To ensure reproducibility and comparability across diverse urban contexts, we additionally evaluated our models on five international, publicly available datasets ($M1$). Notably, this set includes the 2017 New York data also used by Dai et al. (2018), enabling at least partial direct comparison. As mentioned earlier, we are not comparing our results with those of approaches that use GPS trajectory data as input features, such as the works from e.g.Zhang et al. (2025), Zhao et al. (2021), Li and Shuai (2020b), and Miao et al. (2022). These approaches focus on a different aspect of the problem and are therefore not comparable.

The results, presented in Table 10, show that our models consistently achieve higher accuracy across multiple cities and configurations, even under more demanding prediction settings and without relying on user-specific data.

Dai et al. (2018)'s approach predicts station clusters in New York, achieving a maximum accuracy of 0.39. When they adapt this approach to our setting by treating each station as its own class, the performance drops to 0.14. In contrast, our own pipeline for New York, trained on a comparable subset of the same 2017 dataset, achieves an accuracy of up to 0.66, reflecting a substantial improvement of 52%.

Compared to other station-based methods, such as those described in Liu et al. (2019) and Du et al. (2018), our approach achieves higher accuracy in all other cities for which data is available. Notably, the aforementioned methods rely on user identifiers or profile features, whereas our models operate without such information. As we do not have access to their data, a direct comparison is not possible. However, our mean result of 76% for all tested cities suggests that our performance exceeds that of the their methods.

The LSTM-CNN model proposed by Jiang et al. (2019) does not report accuracy, but shows weaker performance on other metrics (precision of 0.54). This suggests that, in certain configurations, our XGBoost-based pipeline may outperform certain deep learning architectures with access to user-specific data. Similar results can be derived for the approach proposed by Jiang et al. (2021), which also only states resolution within a 500-metre radius for their LightGBM model.

We also compare our models with the current state of the art, as defined by Liu et al. (2024). Across nearly all cities — except Stuttgart, London and Shanghai — our models outperform theirs in terms of 500-metre resolution, even when they are using user-specific data. When considering Liu et al. (2024)'s reported non-user-specific models, our approach yields gains of between 2% and 30%. Lower gains are mostly due to a lack of key features such as battery level or current range in the international datasets. Furthermore, our mean accuracy of 76% across all cities is comparable to their approach when using user-specific data. In contrast, we achieve a mean performance increase of 20% when compared to the non-user-specific data approach.

These results suggest that our method is a promising alternative to existing approaches. Although direct dataset-level comparisons are limited by data availability, our consistent performance across ten different urban environments, combined with the generalisability of our feature set, indicates that our approach is robust and scalable, even in the absence of user-specific information.

4.4. Possibility of inter-city transfer learning

A major challenge in applying machine learning for trip destination prediction is the reliance on ground truth data. Data is often unavailable because many modern datasets only provide information where an algorithmic reconstruction of trips is impossible. Inter-City transfer learning presents a potential solution: a model trained on one city, or a combination of cities, could be adapted to predict trips in a city without ground truth data.

In our investigation of this approach, we encountered several limitations that ultimately rendered inter-city transfer learning infeasible for

Table 10
Performance comparison with state-of-the-art solutions. Our models achieve superior accuracy without relying on user-specific data. They outperform all previously reported methods across multiple cities and operators, even under more demanding resolution levels on several datasets from different years.

Paper	Model type	City	Accuracy	Precision
Own ($M1$)	XGBoost	San Francisco	0.89[e]	–
Own ($O1$)	XGBoost	Ludwigshafen	0.86[e]	–
Own ($O1, O2$)	XGBoost	Mannheim	0.81[e], 0.75[e]	–
Own ($M1$)	XGBoost	New York	0.81[e]	–
Own ($O1$)	XGBoost	Heidelberg	0.79[e]	–
Own ($O1, O2$)	XGBoost	Karlsruhe	0.76[e], 0.89[e]	–
Own ($M1$)	XGBoost	Montreal	0.76[e]	–
Liu et al. (2024)	XGBoost	Beijing	0.76[c,d], 0.56[c]	0.75[c,d], 0.55[c]
Own ($O1, O2$)	XGBoost	Stuttgart	0.69[e], 0.67[e]	–
Jiang et al. (2021)	LightGBM	Fuyang	–	0.68[d,e]
Own ($M1$)	XGBoost	London	0.66[e]	–
Own ($M1$)	XGBoost	Shanghai	0.58[e]	–
Liu et al. (2019)	XGBoost	Beijing	0.57[a,d]	–
Jiang et al. (2019)	LSTM-CNN	Beijing	–	0.54[d]
Du et al. (2018)	LightGBM	Beijing	0.45[a,d]	–
Dai et al. (2018)	Random Forest	New York	0.39[a,d], 0.14[a,d]	–

[a]Predicts stations; [b]Predicts station clusters; [c]Result within 1000 × 1000 m; [d]Uses user-specific data; [e]Result within 500 m.

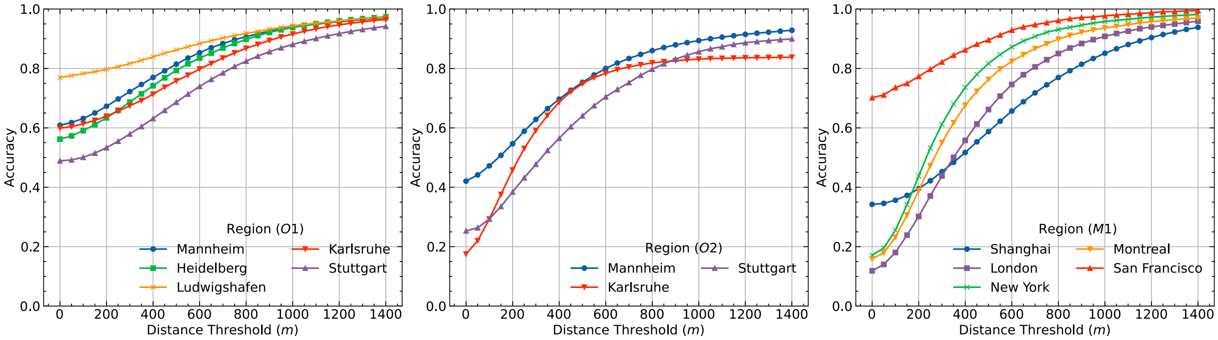

Fig. A.7. Fine granular accuracy by distance threshold and region on the test sets. Further testing of other resolutions indicated that the accuracy converges after 1000 metres.

Table 11
Accuracy by distance threshold and city using an inter-city transfer learning approach. The model trained on Mannheim data was applied to trips in other cities.

Resolution	Cities			
	Heidelberg	Ludwigshafen	Karlsruhe	Stuttgart
@11 m	0.21	0.23	0.20	0.37
@250 m	0.21	0.23	0.20	0.37
@500 m	0.21	0.23	0.20	0.37
@1000 m	0.21	0.23	0.20	0.37

our purposes with XGBoost: As shown in Fig. 6 before, spatial features such as *lat_lend* and *lng_lend* are some of the most important predictors for destination. These coordinates are inherently city-specific, reflecting the local distribution of stations, demand centres, and typical travel patterns. Additionally, local factors such as topology, infrastructure layout, and zoning regulations strongly influence mobility dynamics. Consequently, the relationships learned from one city's spatial structure most likely cannot be generalised to another city. We nevertheless attempted transfer learning by using the model trained on Mannheim data to predict destinations for trips in other German cities. The results are summarised in Table 11.

Overall, performance is consistently lower than that of models trained specifically for each city (see Table 10 for comparison). Notably, all correct predictions occurred when the filtering step reduced the set of destination candidates to a single option, making the machine learning step irrelevant. This outcome was expected. Since the model has never encountered coordinates outside of Mannheim, it predicts destinations relative to Mannheim's coordinate system. This leads to a 100% error rate and unusable results in other cities.

4.5. Security implications for the GBFS

Although the GBFS is designed to provide open access to shared mobility data while protecting personal information, retrospectively predicting destinations raises potential privacy concerns. Our approach does not directly reveal who was using a specific vehicle at a given time since GBFS datasets do not contain user identifiers. However, predicted destinations, when combined with temporal and spatial patterns, can be used to infer habitual trips, particularly in small networks or areas with few users.

For instance, in smaller networks or low-density environments, repeated patterns could enable observers to link trips to particular individuals based on their departure and destination locations and times. Similarly, in settings where trips are sparse, even anonymised vehicle IDs could indirectly reveal daily routines or behavioural patterns. Conversely, in large urban networks with high vehicle turnover and many users, such identification is significantly more challenging due to the sheer volume of trips and the absence of user-specific identifiers.

These observations suggest that, while GBFS data remains broadly anonymised, enhanced predictive methods can partially reconstruct travel behaviours that may previously have been considered private. Therefore, operators and regulators should be aware of the potential for indirect inference of user habits.

Table A.12
Optimal hyperparameters for XGBoost models by city for $M1$. We used Optima to optimise the most important XGBoost model hyperparameter to improve performance.

Parameter	Shanghai	London	New York	Montreal	San Francisco	Range
n_estimators	478	791	476	447	461	$50-1,500$
max_depth	7	14	14	15	15	$3-20$
learning_rate	0.02	0.02	0.09	0.08	0.09	$0.01-0.1$
subsample	0.69	0.68	0.99	0.98	0.83	$0.5-1.0$
colsample_bytree	0.63	0.89	0.89	0.97	0.78	$0.5-1.0$
random_seed	42	42	42	42	42	42

5. Conclusion

This study demonstrates the feasibility of retrospectively predicting trip destinations using datasets from public sharing systems with rotating vehicle IDs. By generating a subset of plausible destinations and applying a machine learning model, accurate predictions were achieved even with rotating vehicle IDs.

Models trained for five German cities achieved an average accuracy of 77% within a 500-metre radius, outperforming existing methods by 21% (when they do not use user features) and performing similarly (when they do use user features). Comparable results were obtained on international datasets from Shanghai, London, New York, Montreal and San Francisco, with minor losses due to missing features, achieving an average accuracy of 74%.

Beyond methodological contributions, our approach offers practical value for operators and city planners. For operators, destination predictions enable proactive fleet rebalancing, targeted pricing or incentive schemes, and efficient planning of charging or maintenance operations. For city planners, predicted destinations reveal demand hotspots and mobility gaps, support infrastructure investment such as bike lanes and parking zones, and highlight opportunities for integration with public transport. Moreover, this solution can be combined with other machine learning approaches — such as demand prediction, traffic simulation, or vehicle rebalancing models — to create more comprehensive decision-support systems. Together, these applications demonstrate that destination prediction can directly support both efficient system management and sustainable urban mobility planning.

Future work could explore performance variations across cities, refine the candidate reduction process to minimise error and investigate likelihood models to handle cases where there are no correct destination candidates. Moreover, future work could address how to adopt a similar pipeline in cities where ground truth data is unavailable and how the approach would need to be adapted to accommodate different spatial characteristics, such as points of interest, using the first results of our inter-city transfer learning approach. Furthermore, we plan to use explainable AI methods such as SHAP to investigate differences in traffic flow and demand hotspots between original and predicted trips at the feature level. Finally, we intend to integrate this solution into our future real-time data processing pipeline to provide real-time analytics across multiple operators and cities.

CRediT authorship contribution statement

Daniel Kerger: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Resources, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Heiner Stuckenschmidt:** Validation, Supervision, Project administration.

Declaration of competing interest

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

Appendix. Additional insights

See Algorithm.

Algorithm 1 Trip Destination Prediction Pipeline

```
procedure Phase 1: Candidate Generation and Reduction
    Load trip data from GBFS in parallel
    for each trip record do
        Handle missing or invalid data
        Extract rentals and returns with GBFS features
    end for
    for each rental do
        for each return in time frame do
            Calculate temporal and spatial features: time_diff, battery_diff, distance, speeds
            Apply time, battery, speed and distance filters
            if non-moving trip detected then
                Mark as definitive match
            else
                Add to candidate set
            end if
        end for
    end for
    Output: Destination candidate mapping
end procedure

procedure Phase 2: Multi-Target Prediction
    Load trained ML model and scaler
    for each rental with candidates do
        for each candidate return do
            Prepare feature vector: coordinates, time, battery, range, distance, speeds
            Scale features using trained scaler
            Predict return coordinates using ML model
            Calculate distance from predicted coordinates to actual return location
        end for
        Store all predictions with distances for candidate selection
    end for
end procedure

procedure Phase 3: Candidate Selection
    for each rental do
        Select candidate with minimum predicted distance as final prediction
    end for
    Evaluate predictions (confusion matrix & evaluation metrics)
end procedure
```

Data availability

A subset of the data and the used source code is available on GitHub. Further data is available on request.

References

Akiba, T., Sano, S., Yanase, T., Ohta, T., Koyama, M., 2019. Optuna: A next-generation hyperparameter optimization framework. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. KDD '19, Association for Computing Machinery, New York, NY, USA, pp. 2623–2631. http://dx.doi.org/10.1145/3292500.3330701.

Au, E., 2019. London bike sharing system data. URL: https://www.kaggle.com/datasets/edenau/london-bike-sharing-system-data, Online.

Bai, S., Jiao, J., 2020. Dockless E-scooter usage patterns and urban built environments: A comparison study of austin, TX, and minneapolis, mn. Travel behaviour and society 20, 264–272.

Bai, S., Jiao, J., Chen, Y., Guo, J., 2021. The relationship between E-scooter travels and daily leisure activities in Austin, Texas. Transp. Res. Part D: Transp. Environ. 95, 102844.

Bissel, M., Becker, S., 2024. Can cargo bikes compete with cars? Cargo bike sharing users rate cargo bikes superior on most motives–especially if they reduced car ownership. Transp. Res. Part F: Traffic Psychol. Behav. 101, 218–235.

Cao, Z., Zhang, X., Chua, K., Yu, H., Zhao, J., 2021. E-scooter sharing to serve short-distance transit trips: A Singapore case. Transp. Res. Part A: Policy Pr. 147, 177–196.

Chahine, R., Duarte, J., Gkritza, K., 2025. Effect of protected bike lanes on bike-sharing ridership: A new york city case study. J. Transp. Geogr. 123, 104147.

Chen, L., Lv, M., Chen, G., 2010-12-01. A system for destination and future route prediction based on trajectory mining. Special Issue PerCom 2010, Pervasive Mob. Comput. Special Issue PerCom 2010, vol. 6 (6), 657–676.http://dx.doi.org/10.1016/j.pmcj.2010.08.004,

Cheng, Z., Trépanier, M., Sun, L., 2022. Real-time forecasting of metro origin-destination matrices with high-order weighted dynamic mode decomposition. Transp. Sci. 56 (4), 904–918.

Citi, 2024. Citi bike trip data. URL: https://s3.amazonaws.com/tripdata/index.html, Online.

Dai, P., Song, C., Lin, H., Jia, P., Xu, Z., 2018. Cluster-based destination prediction in bike sharing system. In: Proceedings of the 2018 Artificial Intelligence and Cloud Computing Conference. AICCC '18, Association for Computing Machinery, New York, NY, USA, pp. 1–8. http://dx.doi.org/10.1145/3299819.3299826.

De Chardon, C.M., Caruso, G., Thomas, I., 2016. Bike-share rebalancing strategies, patterns, and purpose. J. Transp. Geogr. 55, 22–39.

Du, Y., Xiao, B., Xu, W., Cui, D., Xu, Q., Yan, L., 2018. Destination prediction for sharing-bikes' trips. In: 2018 International Conference on Network Infrastructure and Digital Content. IC-NIDC, pp. 198–202. http://dx.doi.org/10.1109/ICNIDC.2018.8525600.

Eriksson, J., Forsman, Å., Niska, A., Gustafsson, S., Sörensen, G., 2019. An analysis of cyclists' speed at combined pedestrian and cycle paths. Traffic Inj. Prev. 20 (sup3), 56–61.

Faghih-Imani, A., Eluru, N., 2015. Analysing bicycle-sharing system user destination choice preferences: Chicago's Divvy system. J. Transp. Geogr. 44, 53–64.

Faghih-Imani, A., Eluru, N., 2020-04-01. A finite mixture modeling approach to examine New York City bicycle sharing system (CitiBike) users' destination preferences. Transportation 47 (2), 529–553. http://dx.doi.org/10.1007/s11116-018-9896-1.

Félix, R., Cambra, P., Moura, F., 2020. Build it and give 'em bikes, and they will come: The effects of cycling infrastructure and bike-sharing system in Lisbon. Case Stud. Transp. Policy 8 (2), 672–682.

García-Palomares, J.C., Gutiérrez, J., Latorre, M., 2012. Optimizing the location of stations in bike-sharing programs: A GIS approach. Appl. Geogr. 35 (1–2), 235–246.

González, F., Melo-Riquelme, C., 2016-05-01. A combined destination and route choice model for a bicycle sharing system. Transportation 43 (3), 407–423. http://dx.doi.org/10.1007/s11116-015-9581-6.

Griffin, G.P., Sener, I.N., 2016. Planning for bike share connectivity to rail transit. J. Public Transp. 19 (2), 1–22.

Hamner, B., 2019. SF bay area bike share dataset. URL: https://www.kaggle.com/datasets/benhamner/sf-bay-area-bike-share, Online.

Heumann, M., Kraschewski, T., Otto, P., Tilch, L., Brauner, T., Breitner, M.H., 2025. Factors influencing the usage of shared micromobility: Implications from Berlin. J. Cycl. Micromobility Res. 4, 100063.

Heywhale, 2020. Public bicycle sharing dataset. URL: https://www.heywhale.com/mw/dataset/5eb6787e366f4d002d77c331/file, Online.

Huang, G., Xu, D., 2023. The last mile matters: Impact of dockless bike-sharing services on traffic congestion. Transp. Res. Part D: Transp. Environ. 121, 103836.

Jaber, A., Abu Baker, L., Csonka, B., 2022. The influence of public transportation stops on bike-sharing destination trips: spatial analysis of Budapest City. Futur. Transp. 2 (3).

Jiang, M., Li, C., Li, K., Liu, H., 2021. Destination prediction based on virtual POI docks in dockless bike-sharing system. IEEE Trans. Intell. Transp. Syst. 23 (3), 2457–2470.

Jiang, J., Lin, F., Fan, J., Lv, H., Wu, J., Li, J., 2019. A destination prediction network based on spatiotemporal data for bike-sharing. Complex. 2019, http://dx.doi.org/10.1155/2019/7643905.

Jiang, W., Ma, Z., Koutsopoulos, H.N., 2022. Deep learning for short-term origin–destination passenger flow prediction under partial observability in urban railway systems. Neural Comput. Appl. 1–18.

Li, Y., Shuai, B., 2020a. Origin and destination forecasting on dockless shared bicycle in a hybrid deep-learning algorithms. Multimedia Tools Appl. 79 (7–8), 5269–5280. http://dx.doi.org/10.1007/s11042-018-6374-x.

Li, Y., Shuai, B., 2020b. Origin and destination forecasting on dockless shared bicycle in a hybrid deep-learning algorithms. Multimedia Tools Appl. 79 (7), 5269–5280.

Liao, C., Chen, C., Xiang, C., Huang, H., Xie, H., Guo, S., 2022-05. Taxi-Passenger's destination prediction via GPS embedding and attention-based BiLSTM model. IEEE Trans. Intell. Transp. Syst. 23 (5), 4460–4473. http://dx.doi.org/10.1109/TITS.2020.3044943.

Lin, L., He, Z., Peeta, S., 2018. Predicting station-level hourly demand in a large-scale bike-sharing network: A graph convolutional neural network approach. Transp. Res. Part C: Emerg. Technol. 97, 258–276.

Lin, S., He, M., Tan, Y., He, M., 2008. Comparison study on operating speeds of electric bicycles and bicycles: experience from field investigation in Kunming, China. Transp. Res. Rec. 2048 (1), 52–59.

Liu, T., Chang, X., Yin, H., Wu, J., Sun, H., 2024. Incorporating user travel behavior into bike-sharing destination prediction: A data-informed macro-micro integration approach. http://dx.doi.org/10.2139/ssrn.4976179, Available at SSRN 4976179, arXiv:4976179.

Liu, C., Gao, X., Wang, X., 2022a. Data adaptive functional outlier detection: Analysis of the Paris bike sharing system data. Inform. Sci. 602, 13–42.

Liu, Y., Jia, R., Xie, X., Liu, Z., 2019. A two-stage destination prediction framework of shared bicycles based on geographical position recommendation. IEEE Intell. Transp. Syst. Mag. 11 (1), 42–47. http://dx.doi.org/10.1109/MITS.2018.2884517.

Liu, L., Zhu, Y., Li, G., Wu, Z., Bai, L., Lin, L., 2022b. Online metro origin-destination prediction via heterogeneous information aggregation. IEEE Trans. Pattern Anal. Mach. Intell. 45 (3), 3574–3589.

Lu, M., Hsu, S.-C., Chen, P.-C., Lee, W.-Y., 2018. Improving the sustainability of integrated transportation system with bike-sharing: A spatial agent-based approach. Sustain. Cities Soc. 41, 44–51.

Luxen, D., Vetter, C., 2011. Real-time routing with OpenStreetMap data. In: Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems. GIS '11, ACM, New York, NY, USA, pp. 513–516. http://dx.doi.org/10.1145/2093973.2094062, URL: http://doi.acm.org/10.1145/2093973.2094062.

Mavrogenidou, P., Polydoropoulou, A., 2025. Personas-based e-scooter usage patterns analysis at a greek research campus. J. Cycl. Micromobility Res. 4, 100062.

McKenzie, G., 2020. Urban mobility in the sharing economy: A spatiotemporal comparison of shared mobility services. Comput. Environ. Urban Syst. 79, 101418.

Miao, H., Fei, Y., Wang, S., Wang, F., Wen, D., 2022. Deep learning based origin-destination prediction via contextual information fusion. Multimedia Tools Appl. 81 (9), 12029–12045.

MobilityData, 2025. General bikeshare feed specification. URL: https://github.com/MobilityData/gbfs, Online.

Noursalehi, P., Koutsopoulos, H.N., Zhao, J., 2021. Dynamic origin-destination prediction in urban rail systems: A multi-resolution spatio-temporal deep learning approach. IEEE Trans. Intell. Transp. Syst. 23 (6), 5106–5115.

OpenStreetMap, 2025. OpenStreetMap. URL: https://www.openstreetmap.org, Online.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E., 2011. Scikit-learn: Machine learning in python. J. Mach. Learn. Res. 12, 2825–2830.

Reck, D.J., Haitao, H., Guidon, S., Axhausen, K.W., 2021. Explaining shared micromobility usage, competition and mode choice by modelling empirical data from Zurich, Switzerland. Transp. Res. Part C: Emerg. Technol. 124, 102947.

Roos, J., Bonnevay, S., Gavin, G., 2016. Short-term urban rail passenger flow forecasting: A dynamic bayesian network approach. In: 2016 15th IEEE International Conference on Machine Learning and Applications. ICMLA, IEEE, pp. 1034–1039.

Sathishkumar, V.E., Park, J., Cho, Y., 2020. Using data mining techniques for bike sharing demand prediction in metropolitan city. Comput. Commun. 153, 353–366.

Si, H., Shi, J.-g., Tang, D., Wu, G., Lan, J., 2020. Understanding intention and behavior toward sustainable usage of bike sharing by extending the theory of planned behavior. Resour. Conserv. Recycl. 152, 104513.

Sigouin, A., 2017. BIXI montreal station data (2017). URL: https://www.kaggle.com/datasets/aubertsigouin/biximtl?select=Stations_2017.csv, Online.

Song, Z., Wu, K., Shao, J., 2020-09. Destination prediction using deep echo state network. Neurocomputing 406, 343–353. http://dx.doi.org/10.1016/j.neucom.2019.09.115.

Strauss, J., Miranda-Moreno, L.F., 2017. Speed, travel time and delay for intersections and road segments in the montreal network using cyclist smartphone GPS data. Transp. Res. Part D: Transp. Environ. 57, 155–171.

Tanaka, K., Kishino, Y., Terada, T., Nishio, S., 2009. A destination prediction method using driving contexts and trajectory for car navigation systems. In: Proceedings of the 2009 ACM Symposium on Applied Computing. SAC '09, Association for Computing Machinery, New York, NY, USA, pp. 190–195. http://dx.doi.org/10.1145/1529282.1529323.

Terada, T., Miyamae, M., Kishino, Y., Tanaka, K., Nishio, S., Nakagawa, T., Yamaguchi, Y., 2006. Design of a car navigation system that predicts user destination. In: 7th International Conference on Mobile Data Management. MDM'06, IEEE, 145–145.

Todd, J., O'Brien, O., Cheshire, J., 2021. A global comparison of bicycle sharing systems. J. Transp. Geogr. 94, 103119.

Tran, T.D., Ovtracht, N., d'Arcier, B.F., 2015. Modeling bike sharing system using built environment factors. Procedia Cirp 30, 293–298.

Vega-Gonzalo, M., Gomez, J., Christidis, P., Vassallo, J.M., 2024. The role of shared mobility in reducing perceived private car dependency. Transp. Res. Part D: Transp. Environ. 126, 104023.

Wang, M., Zhou, X., 2017. Bike-sharing systems and congestion: Evidence from US cities. J. Transp. Geogr. 65, 147–154.

Wielinski, G., Trépanier, M., Morency, C., 2017. Carsharing versus bikesharing: Comparing mobility behaviors. Transp. Res. Rec. 2650 (1), 112–122.

Wu, F., Zheng, C., Du, M., Ma, W., Ma, J., 2025. Heterogeneous multi-view graph gated neural networks for real-time origin-destination matrix prediction in metro systems. Transp. B: Transp. Dyn. 13 (1), 2449483.

Xu, C., Ji, J., Liu, P., 2018. The station-free sharing bike demand forecasting with a deep learning approach and large-scale datasets. Transp. Res. Part C: Emerg. Technol. 95, 47–60.

Xu, Y., Yan, X., Sisiopiku, V.P., Merlin, L.A., Xing, F., Zhao, X., 2022. Micromobility trip origin and destination inference using general bikeshare feed specification data. Transp. Res. Rec. 2676 (11), 223–238.

Yang, Z., Sun, H., Huang, J., Sun, Z., Xiong, H., Qiao, S., Guan, Z., Jia, X., 2020-02. An efficient destination prediction approach based on future trajectory prediction and transition matrix optimization. IEEE Trans. Knowl. Data Eng. 32 (2), 203–217. http://dx.doi.org/10.1109/TKDE.2018.2883938.

Zhang, L., Hu, T., Min, Y., Wu, G., Zhang, J., Feng, P., Gong, P., Ye, J., 2017. A taxi order dispatch model based on combinatorial optimization. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD '17, Association for Computing Machinery, New York, NY, USA, pp. 2151–2159. http://dx.doi.org/10.1145/3097983.3098138.

Zhang, Y., Mi, Z., 2018. Environmental benefits of bike sharing: A big data-based analysis. Appl. Energy 220, 296–301.

Zhang, J., Yang, Y., Wu, X., Li, S., 2025. Spatio-temporal transformer and graph convolutional networks based traffic flow prediction. Sci. Rep. 15 (1), 24299.

Zhang, J., Yu, P.S., 2016-11. Trip route planning for bicycle-sharing systems. In: 2016 IEEE 2nd International Conference on Collaboration and Internet Computing (CIC). pp. 381–390. http://dx.doi.org/10.1109/CIC.2016.057.

Zhao, J., Zhang, L., Ye, J., Xu, C., 2021. MDLF: A multi-view-based deep learning framework for individual trip destination prediction in public transportation systems. IEEE Trans. Intell. Transp. Syst. 23 (8), 13316–13329.

Zhou, X., 2015. Understanding spatiotemporal patterns of biking behavior by analyzing massive bike sharing data in chicago. PloS One 10 (10).