Discussion Paper

Rule-based transport mode classification in a smart travel and mobility survey

Jurgens Fourie
Jonas Klingwort
Yvonne Gootzen

August 2025

Abstract

Smartphones offer an innovative way to collect survey data on travel behavior through smart travel surveys. Smart travel apps potentially reduce the response burden, aiming to become fully automated and requiring no respondent interaction. To explore this, Statistics Netherlands developed a smart-travel app and collected data from a sample of the Dutch population. This paper focuses on classifying the mode of transport, a central smart functionality of the app, by developing a deterministic rule-based algorithm to classify seven transport modes: bike, bus, car, metro, train, tram, and walking. Three manual rule-based algorithms were developed that differed in the feature sets used. The final algorithm achieved an overall classification accuracy of 85% and a balanced accuracy of 70%. Public transport modes are the most challenging to classify. The results show that combining GPS and OSM data is important for achieving transport mode classifications of acceptable quality. The developed algorithm shows that a manual rule-based approach can be highly effective in classifying transport modes. Manual rule-based approaches offer several advantages over machine-learning approaches and should not be overlooked.

Keywords: smart-travel survey, transport mode, passive data collection, OpenStreetMap, GPS, location tracking, decision tree, algorithm

Contents

1 Introduction

Understanding travel behavior is important for national infrastructure planning and decision-making. Travel surveys provide the required demographic, socioeconomic, and track-related data about individuals and households (Murakami et al. 2000). The survey data supports transportation planners, modelers, and policymakers by offering insights into travel behavior, modal preferences, environmental impacts, and sustainable transport solutions (Sadeghian et al. 2022).

Traditional travel diary surveys, requiring manual recording, are prone to non-reporting, inaccuracies, and low response rates (Richardson 2000). Smart travel surveys address these limitations using smartphones with GPS to collect accurate, spatiotemporal travel data (Wu et al. 2016). GPS data enhances precision by providing coordinates and timestamps, filling gaps left by conventional methods (Gillis et al. 2024). Despite a potentially lower response burden and better data quality, smart surveys still face challenges, particularly during participant recruitment (Vij and Shankari 2015). However, smart survey participants tend to complete data collection and provide detailed track information, including rates, lengths, and modes (Gillis et al. 2024; Schouten et al. 2024).

Statistics Netherlands (CBS) is developing a smartphone app to collect travel statistics using GPS data. In combination with AI/ML (Artificial intelligence/Machine learning), location tracking may offer stop-track segmentation, travel mode prediction, and stop-purpose prediction. The first core smart functionality of the CBS app is the stop-track segmentation. The current app classifies GPS data into stops (stationary periods) and tracks (movements between stops), generating an automated digital travel diary. While users currently must manually label stop purposes and transport modes, future app versions aim to automate this process. Recent studies show that the stop/track segmentation achieves good results with a balanced accuracy between 0.91–0.95 (Klingwort et al. 2025c). However, it should be noted that long stop periods, which are usually easier to classify, may cause a positive upward bias in these findings. The second core smart functionality of the CBS app is stop-purpose prediction. The stop purposes can be predicted with a 70% overall balanced accuracy (Zahroh et al. 2025) with some classes, e.g., home, being easier to predict than, for example, doing a visit. The third core smart functionality of the CBS app is transport mode classification. Automating this process reduces the response burden and improves efficiency by replacing manual labeling with an automated and standardized approach. This paper contributes by developing a manual rule-based algorithm for transport mode classification in the CBS app. In particular, the algorithm is intended to classify a single mode per track. Multi-modal tracks cannot be classified using this algorithm.

The paper has five key contributions:

- How accurately can manual rule-based models determine transport mode?
- Which GPS features are relevant to classify the transport mode accurately?
- Which Open Street Map (OSM) data features, next to the GPS features, are relevant to classify the transport mode accurately?
- Does the developed algorithm generalize to different app versions?
- Does the developed algorithm generalize to data from a different country?

The paper is structured as follows. Section 2 contains the research background. The data used is described in Section 3. Section 4 describes the methods and algorithm development. Section 5 contains the results. The paper is discussed in Section 6 and concludes in Section 7. The Appendices A and B contain information on the features used in this study.

2 Research background

Traditional travel surveys, including face-to-face, mail, telephone, and web-based data collection, rely on respondent recall, leading to non-reporting, inaccurate reporting, and non-response (Richardson 2000). Non-reporting occurs when tracks are omitted due to forgetfulness, time constraints, or perceived irrelevance. Inaccurate reporting arises from memory lapses or misinterpreting survey questions, resulting in incorrectly reported details such as times or track purposes. Non-response stems from disinterest, privacy concerns, or survey fatigue. The integration of GPS technology to address these shortcomings began in the late 1990s and early 2000s (Wolf 2000; Schönfelder et al. 2002; Wolf et al. 2001).

GPS loggers were combined with paper-based, web, or phone surveys to enrich travel data and address traditional survey limitations (Schönfelder et al. 2002; Auld et al. 2009). However, standalone GPS loggers do not provide all required information for travel and mobility surveys (Bricka et al. 2009; Harding et al. 2021). As a result, smartphone-based smart-travel surveys have emerged as a more efficient alternative. Smartphone apps automatically record GPS-based track details, requiring participants to validate the data only. Smart-travel surveys address issues like recall bias, proxy bias, mode bias, and inaccuracies in track durations while allowing extended data collection, improving insights into travel behavior and intra-person variability (Harding et al. 2021; Wu et al. 2016; Safi et al. 2014). Despite these advantages, smart-travel surveys demand extensive data processing for analysis and model estimation. Current research focuses on GPS post-processing techniques to automate tasks like GPS-signal smoothing, determining transport mode, and stop purpose (Schuessler and Axhuasen 2009; Zahroh et al. 2025; Gootzen et al. 2025).

Next, research on transport mode classification using machine-learning and rule-based approaches is presented. Further, previous work by CBS is discussed.

2.1 Transport mode classification

The lack of a standard algorithm for transport mode determination is due to variations in data collection methods, application scenarios, and regional factors such as, for example, the cycling prevalence in the Netherlands compared to car dependency in the United States. Five review studies relating to transport mode classification based on GPS tracking data were found in the literature (Lei et al. 2014; Wu et al. 2016; Prelipcean et al. 2017; Sadeghian et al. 2021). Smeets et al. (2019) is not an explicit literature review paper but also contains a comprehensive literature review.

Early research on transportation mode determination predominantly relied on subjective methods such as fuzzy logic, rule-based heuristics, and decision trees, though the latter were less common (Prelipcean et al. 2017). In contrast, a shift toward machine-learning approaches has been observed in recent years. There is a growing preference for semi-supervised and unsupervised methods, driven by the increasing need to work with unlabeled data (Sadeghian et al. 2021). Statistical methods remain rare in this field (Moher et al. 2009; Sadeghian et al. 2021), and currently, rule-based and machine learning techniques dominate the field (Lei et al. 2014; Wu et al. 2016; Prelipcean et al. 2017; Sadeghian et al. 2021).

Despite the advancements, real-time transport mode classification using only GPS-derived features remains challenging due to their limited capacity to differentiate between similar transport modes (Prelipcean et al. 2017). This issue is particularly evident in high-density urban areas, where vehicles such as cars and buses exhibit overlapping movement patterns (Wu et al. 2016). This finding holds irrespective of whether rule-based, supervised, semi-supervised, or unsupervised methods are used. When using GPS features only, speed and speed-related features show the most predictive power. Therefore, using additional information, such as OpenStreetMap (OSM), to derive features about proximity to the infrastructure network improves the model performance considerably. In this study, demographic features about the respondents were purposely omitted, given that previous studies found these features have low predictive power for transport mode prediction.

Machine-learning for transport mode prediction

Machine-learning-based transport mode classification typically follows three approaches: supervised, semi-supervised, and unsupervised learning. Earlier studies primarily relied on supervised methods, which require labeled data but are time-intensive (Sadeghian et al. 2021). More recently, unsupervised methods have gained traction due to their ability to function without manual labeling. However, this shift does not necessarily indicate superior performance, as results vary across different methods, datasets, and study conditions. The Tables 2.1 – 2.3 give an overview of several studies in this field. In these tables, overall accuracy is used to evaluate the study performance. This metric was chosen because it was most often reported or the only metric reported. Overall accuracy can be an insufficient metric and give an over-optimistic picture of the model performance. As will be shown in this study, reporting other metrics, such as balanced accuracy or F1, is strongly advised to draw more realistic conclusions about the model performance.

Table 2.1 contains studies using supervised learning approaches. Supervised learning approaches generally achieved high accuracy, often exceeding 90%, with models such as Bayesian networks and random forests performing well. Studies incorporating OSM features, such as proximity to transit routes, improved classification performance. The dataset sizes varied considerably in terms of the number of users and GPS points. The best-performing studies, such as Feng and Timmermans (2016), achieved near-perfect accuracy, while others showed variability in mode-specific classification, particularly in distinguishing similar modes like buses and cars. The result of near-perfect accuracy is interesting and could indicate over-fitting. It is noticeable in this table that all studies perform better than the study by Smeets et al. (2019). The study by Smeets et al. (2019) is based on the first CBS app developed in 2018 and a random sample from the general population. In contrast, all other studies in this table are based on non-probability samples. These samples consisted of, for example, friends and relatives from the researchers, students, or participants recruited online. The participants were extensively trained in the study by Ansarilari and Golroo (2015). Furthermore, some of these studies predicted less transport modes or hard-to-predict public transport modes were not considered. These reasons might be among the key factors in explaining the difference in performance compared to Smeets et al. (2019).

Table 2.2 contains studies using semi-supervised learning approaches. Semi-supervised learning approaches used labeled and unlabeled data to achieve accuracy, typically around 90%. The use of convolutional autoencoders (CAE) and deep neural networks (DNN) in studies like Dabiri et al. (2019) and Yu (2020) demonstrated effective representation learning, enabling high classification accuracy even with limited labeled data. However, methods like label propagation had lower performance (80%) due to reliance on inferred labels rather than fully supervised learning. The

Table 2.1 Supervised Learning Approaches

Study	Model	GPS Features	OSM Features	Data	Accuracy
Xiao et al. (2015)	Bayesian Network	Avg. speed, 95th speed, avg. accel., low speed, distance, heading rate	None	202 users, 5 days, 1248 person-data streams	Overall: 92.7% Walk: 98.4%, Bike: 89.4%, E-bike: 80.8%, Car: 87.2%, Bus: 87.2%
Feng and Timmermans (2016)	Bayesian Network	Std. dev. speed, avg. speed, max speed, avg. accel., total distance	Distance to transit networks, satellite info, vehicle ownership	8 users, 6-8 weeks, 1554 tracks	Overall: 99.6% Train: 99.3%, Walk: 98.8%, Bike: 99.7%, Car: 99.4%, Bus: 97.7%, Motorbike: 99.9%, Running: 100%, Tram: 100%, Metro: 98.3%
Stenneth et al. (2011)	Random Forest	Avg. speed, accel., heading change rate	Distance to rail/bus lines, stops	6 users, 3 weeks	Overall: 93.4% Train: 93%, Walk: 92%, Bike: 93%, Car: 89%, Bus: 95%
Ansarilari and Golroo (2015)	Random Forest	Delta bearing, speed, accel., delta accel.	None	35 users, 2 weeks, 245,418 GPS points	Overall: 96% Bus: 86.1%, Car: 97.4%, Walk: 90.8%
Smeets et al. (2019)	Random Forest, Collapsed	Distance, time, speed, accel., heading	Proximity to transit routes	517 users, 7 days, 18,414 tracks (5641 labeled)	Overall: 85% Bike: 78.6%, Motorized: 94.9%, Public Transport: 61.3%, Walk: 80.8%
Smeets et al. (2019)	Random Forest, Uncollapsed	Distance, time, speed, accel., heading	Proximity to transit routes	517 users, 7 days, 18,414 tracks (5641 labeled)	Overall: 62% E-Bike: 19.7%, Bike: 75.4%, Car: 95.3%, Metro: 19.4%, Bus: 9.3%, Scooter: 8.9%, Train: 60.4%, Tram: 9.7%, Walk: 80.9%

Table 2.2 Semi-supervised Learning Approaches

Study	Model	GPS Features	OSM Features	Data	Accuracy
Dabiri et al. (2019)	CAE, CNN Classifier	Avg. speed, 95th speed, avg. accel., low speed, travel distance, heading rate	None	182 users (GeoLife), 69 with annotated data, 113 without, 17,621 trajectories	Overall: 92.7% Walk: 74%, Bike: 82%, Bus: 75%, Car: 86%, Train: 86%
Yu (2020)	DNN	Speed, accel., jerk, turn rate	None	182 users (GeoLife), 69 with annotated data, 113 without, 8,120 trajectories, 252,190 tracks	Overall: 91.5% Walk: 95.5%, Bike: 89.6%, Car: 84.6%, Bus: 92.2%, Train: 92%
Rezaie et al. (2017)	Label Propagation, KNN Kernel	Speed, track duration, track length	Distance to transit stops, origin-destination distance	44,000 contacted, participation unknown, 702 validated tracks	Overall: 80%

results indicate that semi-supervised methods balance data efficiency and accuracy, making them suitable for large-scale deployment where labeled data is scarce.

Table 2.3 contains studies using unsupervised learning approaches. Unsupervised learning approaches exhibited more variation in accuracy, with overall performance ranging from 80.5% to 91.8%. Neural network-based methods, such as deep neural networks (DNN) and convolutional neural networks (CNN), achieved the best results, with Ali et al. (2020) reaching an F1 score of over 90% for multiple modes. However, the lack of labeled data in unsupervised approaches made classification generally less accurate than supervised methods, particularly for complex transport mode differentiation. The effectiveness of these models depended largely on the quality of extracted features, as no external labels were available for training.

Table 2.3 Unsupervised Learning Approaches

Study		Model GPS Features	OSM Features	Data	Accuracy
Dabiri and Heaslip (2018)	CNN	Speed, accel., jerk, bearing rate	None	69 users (GeoLife), 32,444 track segments	Overall: 84.8% Walk: 81.6%, Bike: 90.3%, Bus: 80.7%, Car: 86.6%, Train: 92.3%
Gonzalez et al. (2010)	Neural Network	Max/avg speed, accel., critical point ratio	None	114 tracks (smartphone app)	Overall: 91.3% Walk: 100%, Car: 92.1%, Bus: 81.6%
Li et al. (2020)	Unsupervised DNN	Speed, accel., jerk, bearing rate	None	178 users (GeoLife), 25,955 segments	Overall: 86.7%
Markos and Yu (2020)	Unsupervised DNN	Speed, accel., jerk	None	182 users (GeoLife, 5 years), 17,621 trajectories	Overall: 80.5%
Ali et al. (2020)	Concurrent NN	Speed, accel., jerk, distance to prev. point, bearing rate	None	MTL Trajec 2016, 38,845 tracks (walk, bike, transit, car)	Overall: 91.8% F1 Score: Car: 94.7%, Bike: 92.1%, Walk: 91.6%, Transit: 86.1%

Comparing these three approaches, supervised learning remains the most accurate and reliable, especially when large labeled datasets are available. Semi-supervised methods offer a good trade-off by incorporating labeled and unlabeled data, reducing annotation costs while maintaining high performance. Unsupervised learning, while generally less accurate, provides useful insights when labeled data is unavailable or costly to obtain. The choice of method depends on the availability of labeled data, computational resources, and the specific needs of the transportation mode classification task.

Rule-based methods for transport mode prediction

Rule-based methods rely on predefined rules derived from domain knowledge, offering high interpretability and efficiency but lacking adaptability and the ability to learn from unlabeled data. Their strength lies in clear, rule-driven classifications, making them particularly effective when transport modes exhibit distinct behavioral patterns. However, they struggle with complex or overlapping transport modes, such as distinguishing between cars and buses in heavy traffic. Despite these challenges, rule-based methods remain widely used due to their reliability and ease of implementation.

A key advantage of rule-based approaches is their ability to classify transport modes using well-defined conditions, typically identifying four to six transport modes depending on the geographical context. Factors such as the warm start problem (Gong et al. 2012), traffic

congestion, and GPS signal loss introduce classification difficulties, particularly for public transport.

The rule-based and fuzzy logic studies in Table 2.4 show that accuracy in transport mode detection varies across different studies. For example, Chen et al. (2010) reported an overall accuracy of 79.1%, with the highest accuracy for walking (91.5%) and the lowest for trains (28.6%). Gong et al. (2012) achieved an overall accuracy of 82.6%, with walking performing best (92.4%) and trains again showing the lowest accuracy (35.7%). Sauerländer-Biebl et al. (2017) reported an overall accuracy of 75%, which is lower than other studies but did not provide detailed mode-specific accuracy. In Rasmussen et al. (2009), with and without map matching, accuracies were higher—90.6% and 92.4%, respectively—highlighting improved performance when map matching was implemented. The highest accuracy was found in Filip et al. (2013), with an overall accuracy of 91.6%, including up to 10 transport modes. These studies reflect the effectiveness of rule-based and fuzzy logic approaches in transport mode detection, particularly for walking, cars, and bicycles. However, there are notable challenges in accurately detecting public transport modes such as buses and trains, with performance generally lower for these categories across multiple studies.

When compared to machine learning-based approaches, the results from these rule-based and fuzzy logic methods indicate a trade-off between interpretability and flexibility. As the previous discussion shows, machine learning models often show higher accuracy. While rule-based methods benefit from human-defined features and are simpler to implement, they struggle with more dynamic and ambiguous scenarios, such as distinguishing between similar transport modes in congested areas. With its ability to learn from data and adapt to new patterns, machine learning typically outperforms rule-based approaches in mode detection accuracy, particularly in complex, multimodal settings. Thus, while machine learning models offer greater potential for handling diverse and evolving transport data, the motivation in this paper is to develop a rule-based method because of the advantages above.

2.2 Previous work on transport mode prediction at CBS

Previous work by Smeets et al. (2019) tested transport mode classification using a random forest model. They experimented with three feature sets: GPS-only data, GPS + OSM data, and GPS + OSM + registry data. Accuracy improved from 52% with GPS alone to 60% with OSM features and further to 62% when registry data was added. Notably, collapsing transport modes into broader categories (e.g., motorized vs. non-motorized) significantly improved classification accuracy to 85%. However, the model struggled to distinguish between public transport modes (bus, tram, metro, and train) and similar modes like bike and e-bike.

The study also highlighted the dominance of speed-related features in mode classification. In their GPS-only model, nine of the ten most important features were speed-based, with the 95th percentile of speed being the most predictive. When OSM and registry data were included, the most relevant features shifted to proximity measures, such as distance to public transport routes, while speed remained critical.

Despite these advancements, Smeets et al. (2019) had limitations, most notably treating tracks as independent observations, which likely led to overfitting due to users appearing in both training and testing datasets. This study addresses these shortcomings by refining data partitioning methods and improving classification accuracy, particularly for public transport modes.

Table 2.4 Rule-based and fuzzy logic approaches

Study	GPS Features	OSM Features	Data	Accuracy
Chen et al. (2010)	Travel time, speed, GPS signal quality	Public transport network and bus stops and routes	Two datasets, one with 25 users and one with 24 users, collected using a GPS logger over 1 and 5 days, resulting in 311 track segments.	Overall: 79.1% Walk: 91.5% Subway: 68.2% Bus: 53.3% Car: 95.8% Train: 28.6%
Gong et al. (2012)	Average speed, 85th speed, duration, 95th acceleration	Distance to bus stops, train, and subway stations	N = 49 users over 1 weekday, using GPS loggers, totaling 281 track segments.	Overall: 82.6% Walk: 92.4% Subway: 65.6% Train: 35.7% Car: 84.1% Bus: 62.5%
Sauerländer-Biebl et al. (2017)	Mean and maximum speed, start acceleration, change of heading after stop, active time since prior stop, length of deceleration and acceleration times, mean of the differences of the acceleration (horizontal and vertical movement)	None	Analyzed 120 car tracks, with multimodal track analysis.	Overall: 75%
Rasmussen et al. (2009) (No Map Matching)	Speed, track duration, track length	Distance between origin and destination, distance to transit stops	N = 101 users over 3-5 days, using GPS devices, resulting in 741 track segments for analysis.	Overall: 90.6% Walk: 93.8% Bicycle: 85.1% Bus: 75.7% Car: 93.4% Train: 97.1%
Rasmussen et al. (2009) (Map Matching)	Speed, track duration, track length	Distance between origin and destination, distance to transit stops	N = 101 users over 3-5 days, using GPS devices, resulting in 427 track segments for analysis.	Overall: 92.4% Walk: 96.2% Bicycle: 88.5% Bus: 78.4% Car: 95.6% Train: 97.1%
Filip et al. (2013)	Speed, 95th percentile of speed, mean speed, mean moving speed	Railway, tramlines, roads, bus lines, metro, and water surfaces	Movement of 1000 users over 7 days, using a combination of GPS devices and mobile phone data, with a total of 16 million data points.	Overall: 91.6% (10 total modes)

3 Data

First, the development data is described. Second, the feature construction is explained. Third, the independent test data sets are introduced.

3.1 Data acquisition and pre-processing

CBS collected travel data through an app from November 2022 to February 2023 (Travel survey field test 2022/2023). First, a follow-up sample of former participants in the regular 2022 travel survey (sample size n=667) was used. Second, a new random sample of the Dutch population of 16 years and older (sample size n=2544) was drawn. The ultimate response dataset consists of 505 users. Three experimental conditions were randomized across the sample: reporting period, concurrent online questionnaire, and editing options. An experiment was introduced to assess the extent of editing respondents were allowed to perform. The sample was divided into two groups: one with full editing capabilities and the other with limited editing options. Full editing allowed respondents to add or delete stops and tracks, modify start and end times, and label travel modes and purposes. In contrast, limited editing was restricted to deleting stops and tracks and adding labels. It is important to note that editing stops and tracks was optional and not required by users. More information can be found in the paper by Schouten et al. (2024).

The CBS app processes smartphone GPS data by classifying it into two categories: stops and tracks. A stop is a phase where the user remains within a defined radius for a specified duration, indicating little to no movement. In contrast, a track represents a movement phase, such as when a user travels from the supermarket to their home. For all phases, users were required to provide the mode of transport and the reason for a stop, and they could add, remove, or connect tracks that were incorrectly presented to them by the app. Users could select the following modes: bus, bike, e-bike, car, tram, train, metro, walk, and other. The mode e-bike was merged with the mode bike into one bike category to address the overlap between these two modes and to account for the small number of e-bike tracks. The classification ‘other transport mode’ was excluded from further analysis as it did not represent a clearly defined transport mode.

The segmentation into stops and tracks and the transport mode classification depend highly on data resolution and gaps. It is important to know how detailed and frequent the measurements are (e.g., a GPS point every second vs. every 5 minutes). Higher resolution will likely allow better detection of when and where someone stopped or which transport mode was used. Missing data occurs often due to signal loss (e.g., in tunnels, urban canyons, or underground transport). Gaps can cause errors, for example, missing a stop or misclassifying a fast walk as a slow bike ride. This matters because if the data resolution is low (infrequent or coarse measurements) or there are gaps, the algorithm might miss short stops or falsely detect a stop. Transport modes might be misclassified (e.g., a bus ride mistaken for a car trip). This will reduce the overall accuracy and reliability of the travel survey results. For details on data quality and missing data, see Gootzen et al. (2025) and McCool et al. (2024).

After traveling with the app installed on their smartphones, users were presented with a spatial map of how they traveled during the day. The app user interface can be seen in Figure 3.1, where users can interact with the app and provide details on transport mode and purpose. Figure 3.1(b) shows what is presented to a user at the end of the day. The user is presented with stop and track segments and is asked to label the stop’s purpose and the track’s transport mode. This is where the app relies on the user’s ability to recall their travel behavior; one would expect that

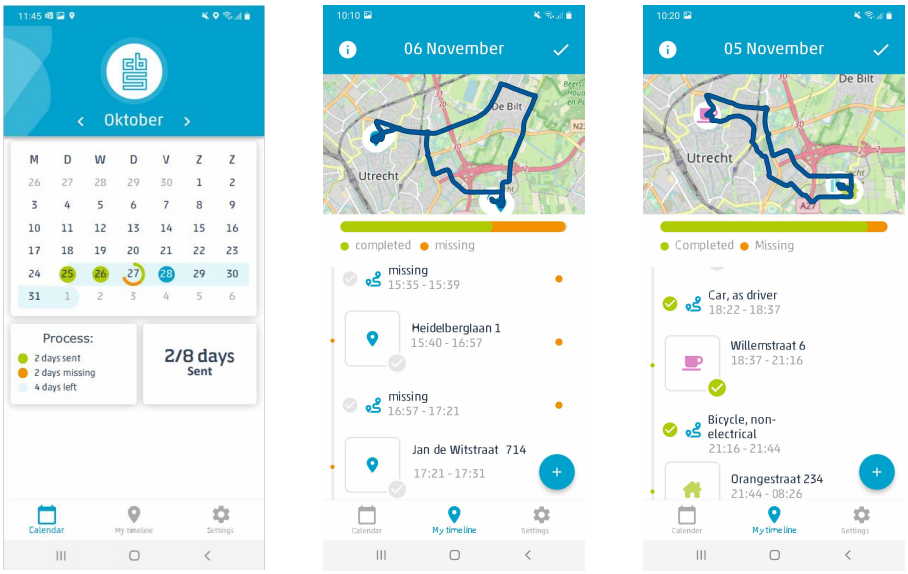
this is easier for the traveler than making their diary, but it could still result in some tracks and stops being wrongly labeled. Figure 3.1(c) shows the labeled stops and tracks. After the user has filled in the details of his daily travel behavior, the user can save the details and do this process daily. For this paper, the user-provided information is assumed to be correct. Information on labeling behavior is reported by Remmerswaal et al. (2025).

Figure 3.1 Travel app user interface.

(a) Homepage with diary overview

(b) Returned GPS track

(c) labelled GPS track



3.1.1 Data processing

The study by Gootzen et al. (2025) categorized respondents into 'sufficient' and 'insufficient' data quality profiles based on two criteria: the number of observed hours and geolocations. Users with fewer than 2000 geolocations or less than one hour of data were classified as 'insufficient' and excluded, resulting in 75% of users having sufficient data (Gootzen et al. 2025). After this step, 369 of the 505 initial respondents remain. In this study, users with 'sufficient' data quality are considered.

The raw GPS measurements, consisting of a timestamp, longitude, and latitude, are aggregated into events. The events are tracks or stops. The initial event data consists of 14026 stops and tracks with 6734 stop segments and 5943 track segments, with 1349 unlabelled events. The event data is based on the GPS data that contains 15,654,868 observations. Table 3.1 summarizes the number of observations for stops and tracks. For the development, 3,626,812 GPS observations, 257 users, and 4507 tracks can be used.

Table 3.1 Summary of GPS measurements for stop and track events

Label	Type	Number of Observations	Percentage
Stop	Labeled	6,842,589	44
Stop	Unlabeled	1,863,513	12
Track	Labeled	3,626,812	23
Track	Unlabeled	1,016,335	6
Missing	Missing	2,305,619	15

Data was collected using smartphones; users had different phones with different operating systems and sensors. Furthermore, a phone can use multiple sensors simultaneously to collect data, so there are duplicated observations when multiple sensors return timestamps and geolocation measurements. When checking for duplicates based on user ID, cluster ID, longitude, latitude, transport mode, and timestamps, there are 390,803 duplicates. These duplicates were removed, resulting in 3,236,009 observations, 257 users, and 4507 tracks.

Tracks exceeding 10 hours were excluded, as they may represent anomalies or malfunctions in recording, resulting in 60 tracks being removed. Tracks with fewer than 10 observations were removed because they lacked sufficient data points for meaningful analysis, resulting in 182 tracks being removed. Tracks labeled with 'Other' as the transport mode are also removed to focus on identifiable transport modes, resulting in 113 tracks being removed. After these steps, 2,973,586 observations, 251 users, and 4159 tracks remained.

3.2 Features

This section provides information on the GPS and OSM features created for transport mode classification.

3.2.1 GPS features

Here, the features based on the GPS measurements are described. These features were calculated on the timestamp level and aggregated to the event level. For all features created, several summary statistics were calculated: mean, standard deviation, median, minimum, maximum, interquartile range, skewness, kurtosis, and 85th and 95th percentiles. An overview is given in Appendix A.

The Haversine formula was used to determine the distance between consecutive GPS points and to calculate speed. Haversine was used because it accounts for the curvature of the Earth, providing a more accurate representation of the distance traveled. Speed was computed as the ratio of this distance to the time interval between consecutive GPS measurements.

Acceleration, jerk, and snap were calculated (jerk, the derivative of acceleration, and snap, the derivative of jerk). Each variable measures how motion changes over time, with acceleration being the most common, while jerk and snap are used in engineering and physics to describe smoother or more abrupt changes in motion, potentially valuable features to distinguish between transport modes. Acceleration is calculated as the rate of speed change over the interval between consecutive GPS points in a track. Jerk represents the rate of change in acceleration over time, while snap quantifies the jerk shift over time.

Low-speed intervals and their duration were calculated based on a criterion where the speed between two consecutive points is less than 3 km/h for less than 5 minutes. When points meet the conditions, the total number of low-speed periods and their combined duration are calculated for a given track. The frequency and duration of these low-speed intervals are hypothesized to provide insights into the characteristics of different transport modes. For example, public transport modes like buses are likelier to have frequent low-speed intervals than cars. Besides the low-speed intervals in a track, high-speed intervals were also calculated. High-speed intervals are defined as the proportion of time spent during a track at speeds of 120 km/h and above.

The sampling rate was calculated as 1/time_diff, where time_diff is the difference between consecutive GPS points in a track. This metric is also hypothesized to provide insights into specific transport modes. For example, a mode with weak GPS signals throughout a track may display irregular sampling frequencies. Variations in sampling frequency could reflect the travel characteristics of specific transport modes—for example, the stop-and-go patterns in public transport or continuous movement in cycling.

Smartphones provide information on bearing and accuracy. Bearing and changes in bearing were calculated using the longitude and latitude coordinates from the GPS data. Bearing refers to the direction of travel from one point to the next, given in degrees measured clockwise from true north (0). For example, a bearing of 90 indicates travel directly east. This feature is created to analyze directional changes in movement. Accuracy is a measure to represent the level of confidence in the GPS data's precision, indicating how closely the reported location aligns with the user's actual position. Additionally, a feature was created representing the proportion of GPS points that meet a predefined accuracy threshold of 5 meters.

The feature speed showed extreme outliers. This finding was despite the initial data cleaning applied by Gootzen et al. (2025). We define an extreme outlier or spike as a sudden and considerable/unreasonable increase or decrease in speed values. Median smoothing solved this issue. The smoothed speed was used as the basis for all other features based on speed (e.g., acceleration, jerk, or snap). Details and visualizations on the smoothing can be found in Fourie (2025).

3.2.2 OSM features

Here, the features based on OpenStreetMap (OSM) are described. OSM data is a comprehensive and widely used source of geospatial information, offering detailed geographic data about streets, buildings, and other infrastructure. The OSM data was downloaded from GeoFabrik. OSM data can be characterized by attributes describing every aspect of the platform's free tagging system. In OSM, a tag represents a combination of keys and values. There is an unofficial set of criteria for the most popular key-value tag pairs, which the community uses and can be found at OSMWiki. Tags can consist of many keys. For example, the railway tag includes values such as abandoned, construction, proposed, disused, funicular, light_rail, miniature, monorail, narrow_gauge, rail, subway, and tram, all representing different forms of railway infrastructure information. There are three types of elements: nodes (defining points in space), ways (defining linear features and area boundaries), and relations (defining how other elements interact).

Parking-related features were created using the OSM keys parking, fuel, parking_entrance, bicycle_parking, charging_station, and bicycle_rental. These features were derived by taking the first and last five points for a given track and calculating the distance from these points to the nearest parking_entrance, for example. These features were selected because proximity to parking facilities and charging stations can provide valuable insights into specific transport modes like cars or bikes.

Traffic-related features were created using infrastructure data such as stops, crossings, and street lamps. These features were calculated by drawing a 10-meter buffer around a track and each traffic-related key. It was checked whether the buffer intersected with any of these traffic-related features. If an intersection was detected, the number of objects passed during the track was counted. Alongside the counts, a normalized count was calculated by dividing the raw counts by the area of the entire track buffer. This normalization accounts for variations in the distance traveled by different users, enabling a fair comparison of counts across different tracks and transport modes.

Proximity to public transport stations was calculated to support distinguishing between public and private transport modes. These features were created by taking a track's first and last five measurements and calculating the average distance to the nearest train, tram, metro, and bus station.

The proximity of the 50th percentile point of the track to the nearest train, tram, metro, and bus route was calculated. The idea was that halfway through a track, we expect more divergence from different transport mode routes than at the beginning of a track.

The number of public transport routes overlapped during a track was calculated. For example, during a train track, we expect it to overlap with more train routes, whereas a tram track would overlap with more tram routes. This was done by creating a 10-meter buffer around the track and public transport routes. If the track intersects with a route, it is counted as an overlap. A normalized count was also calculated by dividing the number of routes intercepted by the area of the track's buffer to account for the length of a track.

The entire route's proximity to public transport routes was also calculated. This was done by creating a 10-meter buffer around the track and the route and checking if they overlapped. While a track may overlap with multiple routes, this approach considerably reduces computation costs compared to comparing each point in a route with all tracks. Summary statistics that represent the relative distance of all points in a track to the selected route were calculated. This was done for all public transport and bike routes as well.

In all cases, a buffer size of 10 meters was chosen. Different buffer sizes of 10, 20, 50, 75, and 100 meters were tested. No noticeable differences were observed. Therefore, it was decided to use a buffer size of 10 meters, as this increased the computational efficiency.

3.3 App data for external validation

The two datasets described above were exclusively reserved to test the developed algorithm. None of this data was used during development or in the train, and test splits are to be described in Section 4.1. These datasets will be used for external validation to be presented in Section 5.4. The app sensor configurations to collect this data differ from the sensor configuration used for the development dataset. For example, for the field test 2024, an optimized sensor configuration that collected GPS data on a considerably higher frequency was used. Furthermore, the 2024 data contains observations in two countries (The Netherlands and Germany).

Field test 2018

2018 CBS conducted the first field test using a travel app (Travel Survey Field Test 2018). About 2,000 individuals sampled from the Dutch population were invited to participate. 674 individuals downloaded the app, and 517 completed data collection for at least 7 days. In total, 75,649,744 GPS measurements were collected. The data contains 18,414 tracks, of which respondents labeled 5,641 with the mode of transport. Details can be found in Smeets et al. (2019) and McCool et al. (2021).

Data from 361 users with 4417 single-mode tracks (multi-modal tracks were excluded for this study) were used to test the developed algorithm. There are 2458 car tracks (56%), 956 bike tracks (22%), 773 walk tracks (18%), 131 train tracks (3%), 59 bus tracks (1%), 27 tram tracks (0.05%), and 13 metro tracks (0.3%). Here, the respondents could nod, add, or delete anything recorded by the app; they could only label the classified stops and tracks.

Field test 2024

In 2024, data was collected to obtain data with high-quality labels for the transport mode. This data was collected by a small group of CBS staff and staff from the University of Utrecht specifically trained and instructed for this data collection. This data set is limited in size (users, tracks, variation in transport modes) but has the advantage of error-free labels. It contains tracks in the Netherlands and Germany.

In total, data from five users with 137 tracks are available. A total of about 429,000 GPS observations were made. There are 78 walking tracks (61%), 27 tram tracks (21%), 10 bike tracks (8%), 9 train tracks (7%), 5 bus tracks (4%), 5 metro tracks (4%), and 3 ferry tracks (2%). The ferry tracks were removed, as the algorithm was trained and developed on data that does not contain this label. The transport mode car is not in the used modes. Tracks with less than 10 observations were removed. After pre-processing, about 402,000 raw GPS measurements and 134 tracks were available for analysis.

4 Methods

First, the train and test split is described and discussed. Second, the three developed algorithms are presented. Third, the metrics to evaluate the algorithms are explained.

4.1 Train and test split

The splitting was performed by dividing the users into subsets for training (70%) and testing (30%), ensuring that users do not appear in both the training and test sets (using the data described in Section 3.1). Stratification was applied based on each user's dominant transport mode to maintain an equal distribution of transport modes in both subsets. For example, consider a dataset where 60% of users predominantly travel by car, 30% use the metro, and 10% walk. Stratification would maintain these proportions in both the training and test sets. If there are 100 users, 70 would be allocated to the training set and 30 to the test set. Among the training set, approximately 42 users (60%) would be car users, 21 (30%) would be metro users, and 7 (10%) would walk. Similarly, the test set would include 18 car users, 9 metro users, and 3 walks, preserving the overall distribution of transport modes. This approach preserved the variability and distribution of transport modes in the training and test sets, ensuring that the test set accurately reflected the characteristics of the training data while preventing overlap between users in the two subsets. Tables 4.1 show all transport modes, except for bus and metro, exhibit a consistent distribution between the training and test sets. Although the share of each transport mode is preserved in the training and test sets, the class imbalance in the data remains present. This imbalance arises from the users participating in the study and their respective travel behavior. To maintain the structure of the data, over-sampling or under-sampling techniques were not used, ensuring that the analysis reflects the original dataset as closely as possible.

4.2 Rule-based algorithms

Three algorithms will be developed: ALG1, ALG2, and ALG3. The algorithms build upon each other. First, ALG1 was developed. This algorithm is purely based on GPS features only. The decision rules were developed based on summary statistics and visual feature inspections.

Table 4.1 Training and testing data

Training data

Mode	Count	Percentage
Car	1,288	44
Bus	52	2
Tram	22	1
Train	111	4
Metro	48	2
Walk	709	24
Bike	665	23
Total	2,895	100

Testing data

Mode	Count	Percentage
Car	570	45
Bus	43	3
Tram	11	1
Train	45	4
Metro	7	0
Walk	299	24
Bike	289	23
Total	1,264	100

Second, ALG2 was developed and built on ALG1. This algorithm is still based on GPS features only. A decision tree was fitted to analyze whether important features were missed in ALG1 and to evaluate whether the thresholds chosen in ALG1 could be refined. Third, ALG3 builds upon the previous developments but uses GPS and OSM features. Here, a decision tree was also fitted to analyze whether important features were missed in ALG1 or ALG2 and to see which OSM features are relevant. None of the presented algorithms here are based purely on a decision tree. The decision trees were only used to support the rules. This hybrid design was considered to improve robustness in cases where data is sparse or noisy. It would maintain interpretability for critical decisions (via rules) and allow adaptability in recognizing complex travel behaviors (via machine learning). The hybrid approach thus offers a pragmatic balance, particularly in real-world smart travel survey applications where data quality and context vary considerably. However, during development and implementation, many of the insights derived from the decision trees were ultimately not incorporated, as they did not contribute meaningfully to the final algorithms. Limitations of decision trees for this classification tasks are reported and discussed by Klingwort et al. (2025a) and Klingwort et al. (2025b).

All algorithms allow for multiple transport mode classifications, i.e., the algorithm's outcome may be a tram, metro, and bus. Such a result would reflect uncertainty in the classification because the observed data and features meet the conditions of all three transport modes. ALG1 and ALG2 allowed for several combinations of multiple classifications. ALG3 only allowed for the multiple classification of cars or buses. Minimizing multiple classifications was an essential factor when developing the algorithms. Furthermore, the algorithms also allowed the classification to be unknown, i.e., no single or multiple classification could be made, and none of the decision rules were met.

4.2.1 ALG1 - GPS

The development of ALG1 is based on descriptive statistics, visual analysis, and existing literature. It was explorative and built iteratively, starting with a single feature (e.g., mean speed) and setting threshold values based on visual analysis and descriptive statistics. Features were added incrementally, with inclusion based on a performance improvement of at least 0.05 in one of the class-specific evaluation metrics. This process ensured ALG1 remained effective and interpretable, avoiding overfitting while retaining essential features.

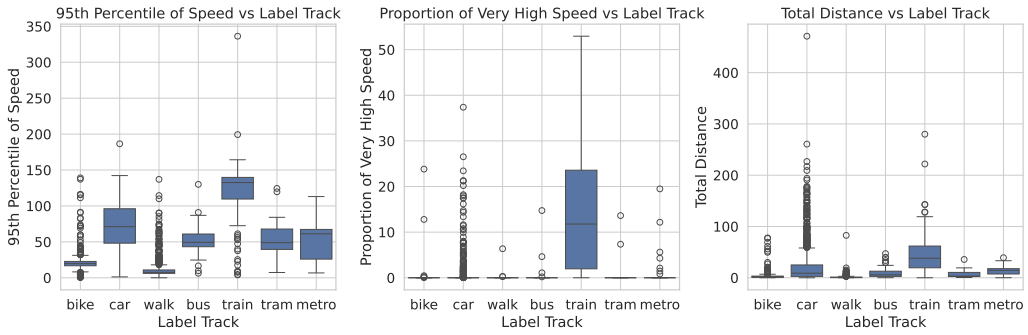
The 95th percentile of speed was the most effective variable for splitting among all the speed-related features tested. Walking and biking showed distinct speed ranges that differed from motorized modes. However, overlap still exists with other modes such as tram, metro, bus, and car. This supported the findings from previous research that OSM data is necessary to distinguish between these modes. In addition to speed-related features, total distance traveled

seemed promising in differentiating motorized and non-motorized modes. For instance, 86% of bike tracks and 96% of walk tracks were shorter than 5km. Furthermore, the proportion at very high speed effectively separated trains from other motorized modes, addressing a key challenge initially faced in identifying trains. Visual inspection and descriptive statistics revealed that the other GPS features lack clear distinctions between transport modes. While minor differences exist, considerable overlap complicates differentiation. Relying on such features results in overly specific rules, leading to multiple classifications for some features while leaving others unclassified.

To set the decision thresholds, rounding was found to be the nearest whole number or multiples of 5 or 10. A case where rounding decisions come into play is, for instance, when the 95th percentile of speed for walking ranges from 0 to 11 and for bicycles from 0 to 25. If 90% of bicycle speeds are greater than 13 and 80% of walking is less than 13, then 13 would be chosen as the starting threshold to differentiate walking from cycling. By avoiding overly specific thresholds, this approach provided a more robust set of rules that accommodated potential variability between the training and testing datasets and general differences in travel behavior.

The final set of selected GPS features for ALG1 is presented in Figure 4.1. The Python code for ALG1 is shown in Listing 1 in Appendix C. ALG1 is described in words below.

Figure 4.1 GPS features used in ALG1



- **Algorithm description:** The algorithm classifies transportation modes based on the 95th percentile of speed, total travel distance, and the proportion of very high speeds. Multiple transport modes can be identified per trip depending on overlapping criteria.
- **Steps:**
  - **Initialize** an empty list `modes` to store classified transportation modes.
  - **Classify based on speed and distance:**
    - If `speed_p95` $<$ 13 km/h $\Rightarrow$ classify as **walk**.
    - If 13 km/h $<$ `speed_p95` $<$ 30 km/h $\Rightarrow$ classify as **bike**.
    - If 30 km/h $<$ `speed_p95` $<$ 140 km/h $\Rightarrow$ classify as **car**.
    - If 40 km/h $<$ `speed_p95` $<$ 80 km/h and `total_distance` $<$ 5 km $\Rightarrow$ classify as **bus**.
    - If `speed_p95` $>$ 65 km/h and `prop_very_high_speed` $>$ 10% $\Rightarrow$ classify as **train**.
    - If 35 km/h $<$ `speed_p95` $<$ 70 km/h and `total_distance` $<$ 5 km $\Rightarrow$ classify as **metro**.
    - If 30 km/h $<$ `speed_p95` $<$ 60 km/h and `total_distance` $<$ 5 km $\Rightarrow$ classify as **tram**.
  - If no mode is detected, assign **unknown**.
- **Final Output:** The classified transportation modes are stored in a new column `modes` in the data frame.
- **Special Note:** If multiple modes are detected, those are not included in the final output, but moved to a separate output that contains the multiple classification.

4.2.2 ALG2 - GPS and decision tree

ALG2 is developed based on ALG1 and supported by a decision tree. Deterministic rule-based algorithms classify data using predefined logical rules rather than learned patterns. Unlike machine learning models, which infer relationships from training data, these algorithms apply explicit if-else conditions to reach decisions. Decision trees are supervised machine learning models that partition data based on feature values, constructing a tree-like structure where internal nodes represent decisions, branches denote possible outcomes, and leaf nodes indicate predicted categories. These two methods differ in decision-making. Deterministic algorithms rely on expert-defined rules, while decision trees autonomously learn patterns from data, enabling them to model complex interactions. Deterministic rule-based approaches can accommodate multiple classifications, allowing for uncertainty, whereas decision trees typically yield a single prediction. Additionally, decision trees may incorporate irrelevant features if they improve predictive performance, while deterministic models explicitly include only logically relevant features.

The decision tree used a 10-fold cross-validation to ensure generalization. Note that the cross-validation still only used the 70% training set. A grid search was conducted to optimize hyperparameters, such as tree depth, to prevent overfitting. The decision tree's feature selection and threshold values were compared to those used in ALG1 to identify overlooked features or refine thresholds. Newly identified features were incorporated into ALG2 to improve decision performance, particularly for challenging transport modes. To be explicit, ALG2 is still a deterministic rule-based algorithm, but its development was supported by a decision tree to optimize thresholds or to identify previously overlooked features. Information and the output of the decision tree are not shown in this paper and can be found in Fourie (2025). The Python code for ALG2 is shown in Listing 2 in Appendix C. ALG2 is described in words below.

- **Algorithm description:** The algorithm classifies transportation modes based on the 85th and 95th percentiles of speed, total travel distance, and the proportion of very high speeds. Multiple transport modes can be identified per trip depending on overlapping criteria.
- **Steps:**
  - **Initialize** an empty list `modes` to store classified transportation modes.
  - **Classify based on speed and distance:**
    - If `speed_p85` $<$ 12 km/h $\Rightarrow$ classify as **walk**.
    - If 12 km/h $<$ `speed_p85` $<$ 25 km/h $\Rightarrow$ classify as **bike**.
    - If `prop_very_high_speed` $>$ 10% or `speed_p95` $>$ 130 km/h $\Rightarrow$ classify as **train**.
    - If 30 km/h $<$ `speed_p95` $<$ 125 km/h $\Rightarrow$ classify as **car**.
    - If 40 km/h $<$ `speed_p95` $<$ 80 km/h and `total_distance` $<$ 5 km $\Rightarrow$ classify as **bus**.
    - If 30 km/h $<$ `speed_p95` $<$ 70 km/h and `total_distance` $<$ 5 km $\Rightarrow$ classify as **metro**.
    - If 30 km/h $<$ `speed_p95` $<$ 70 km/h and `total_distance` $<$ 5 km $\Rightarrow$ classify as **tram**.
  - If no mode is detected, assign **unknown**.
- **Final Output:** The classified transportation modes are stored in a new column `modes` in the data frame.
- **Special Note:** If multiple modes are detected, those are not included in the final output, but moved to a separate output that contains the multiple classification.

4.2.3 ALG3 - GPS, OSM, decision tree

ALG1 and ALG2 only used GPS features. ALG3 also incorporates OSM features to address the potential limitations of the GPS features. For ALG3, a combined approach was employed as well. ALG3 is based on the previously developed ALG1 and ALG2 and results from a decision tree. The decision tree also used GPS and OSM data. ALG3 was developed iteratively, and the decision tree

supported the development of the rule-based algorithm. Each iteration involved tuning thresholds, adding/removing features, and addressing miss-classifications to improve model performance.

However, the decision tree failed to predict public transport modes effectively. That was also previously the case with ALG2. Additionally, some splitting variables used were indirectly related to the transport mode. For example, the number of train stations was used to identify metro and train modes. Though these two are somewhat related, using features directly related to the mode classified is more reasonable and preferred in this study. Thus, ALG3 will focus on using features directly related to the specific transport mode. As a result, the decision tree was not used as a reference. Instead, features specifically created to identify trains will be utilized for modes such as trains. For public transport modes, the features used in ALG3 include relative proximity measures to specific routes, ensuring a targeted and mode-specific approach to decisions. Public transport modes will be classified using relative distance metrics to particular routes, such as minimum distance, standard deviation of distance, and mean distance to the route associated with a given track. Information and the output of the decision tree are not shown in this paper and can be found in Fourie (2025).

Table 4.2 shows that transport modes associated with a specific route typically have corresponding proximity statistics. For instance, 98% of metro tracks have associated routes and summary statistics, whereas only 40% of train tracks include proximity statistics for metro routes. This further reinforces the idea of using proximity measures directly related to a specific transport mode to predict that particular mode. Doing so minimized the overlap between transport modes and reduced instances of multiple classifications.

Table 4.2 Percentage of tracks that have proximity values

Proximity to Metro Route

Mode	Percentage
Car	5
Bus	25
Tram	45
Train	38
Metro	98
Walk	6
Bike	8

Proximity to Train Route

Mode	Percentage
Car	48
Bus	58
Tram	59
Train	92
Metro	79
Walk	9
Bike	24

Table 4.2 provides additional insights into the potential for constructing a nested structure. The decision to classify ‘walk’ first, followed by ‘bike’, was based on their performance in ALG1 and ALG2. The order of ‘metro’ and ‘tram’ had minimal impact on performance. However, the larger disparity in metro values compared to train values suggests that classifying metro before train could reduce conflicts and simplify classification. These considerations informed the development of ALG3. ‘Bus’ and ‘car’ were intentionally placed at the end of the algorithm due to their similarities, with 89% of car tracks having proximity measures. This made defining clear thresholds difficult, leading to an approach that allows multiple classifications to balance miss-classification risks between these modes. Since public transport classification relies on three proximity measures without predefined thresholds, a grid search within nested loops was used to estimate initial thresholds. These were then refined by adjusting thresholds based on descriptive statistics and their effect on precision-recall trade-offs. Achieving a balance between precision and recall was particularly challenging for specific modes, especially metro and tram.

Effective classification often required combinations of the three proximity metrics, using either independent 'or' statements or conjunctions with 'and' for more specific conditions. For example, in Listing 3, tram classification was most effective with a mixed condition: (x and y) or z. In contrast, train classification performed well using only a 'or' condition: x or y or z. The Python code for ALG3 is shown in Listing 3 in Appendix C. ALG3 is described in words below.

- **Algorithm description:** The algorithm classifies transportation modes based on statistical speed metrics and total travel distance to determine the most likely mode of transport.
- **Steps:**
  - **Initialize** an empty list `modes` to store classified transportation modes.
  - **Classify based on speed:**
    - If `speed_p95` $<$ 13 km/h $\Rightarrow$ classify as **walk**.
    - If 13 km/h $<$ `speed_p95` $<$ 30 km/h $\Rightarrow$ classify as **bike**.
  - **Classify based on proximity to transport infrastructure:**
    - If close to tram stations (`min_distance_tram` $<$ 0.5 m, `std_distance_tram` $<$ 250 m, or `mean_distance_tram` $<$ 100 m) $\Rightarrow$ classify as **tram**.
    - If close to metro stations (`min_distance_metro` $<$ 1.5 m, `std_distance_metro` $<$ 450 m, or `mean_distance_metro` $<$ 100 m) $\Rightarrow$ classify as **metro**.
    - If close to train stations (`min_distance_train` $<$ 0.05 m, `std_distance_train` $<$ 25 m, or `mean_distance_train` $<$ 100 m) $\Rightarrow$ classify as **train**.
    - If close to bus stops (`std_distance_bus` $<$ 120 m, `mean_distance_bus` $<$ 40 m, or `min_distance_bus` $<$ 0.015 m) $\Rightarrow$ classify as **bus**.
  - **Classify as car:** If 30 km/h $<$ `speed_p95` $<$ 140 km/h $\Rightarrow$ classify as **car**.
  - If no mode is detected, assign **unknown**.
- **Final Output:** The classified transportation modes are stored in a new column `modes` in the data frame.
- **Note:** If the modes car and bus are detected, those are not included in the final output, but moved to a separate output that contained the multiple classification.

Table C.1 in Appendix C shows a comparison of ALG1, ALG2, and ALG3.

4.3 Evaluation metrics

The algorithm's performance will be assessed using precision, recall, F1-score, accuracy, and balanced accuracy, metrics commonly used in transport mode prediction. Precision evaluates the model's ability to minimize false positives, while recall measures its ability to capture true positives. The F1-score combines both metrics to provide a balanced evaluation, which is particularly useful for imbalanced datasets. Accuracy represents the overall correctness of predictions but can be misleading in imbalanced data, whereas balanced accuracy offers an evaluation by averaging recall across all classes.

Key definitions include true positive (TP) when the model correctly predicts the actual class (e.g., predicting 'walking' when correct); false positive (FP), where an incorrect class is predicted (e.g., predicting 'car' instead of 'bike'), false negative (FN), when the correct class is missed, and true negative (TN), when incorrect classes are correctly excluded.

The formulas for each metric are:

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

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

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

$$\text{Balanced Accuracy} = \frac{1}{2}\left(\frac{\text{TP}}{\text{TP} + \text{FN}} + \frac{\text{TN}}{\text{TN} + \text{FP}}\right)$$

5 Results

This section presents the results of the five key contributions of this paper. The first question, 'How accurately can manual rule-based models determine transport mode?' is answered by the entirety of Section 5. Section 5.1 and Section 5.2 answer the question, 'Which GPS features are relevant to classify the transport mode accurately?'. Section 5.3 addresses 'Which Open Street Map (OSM) data features, next to the GPS features, are relevant to classify the transport mode accurately?'. In Section 5.4, the questions 'Does the developed algorithm generalize to different app versions?' and 'Does the developed algorithm generalize to data from a different country?' are answered. The presentation of the results in Sections 5.1 – 5.3 follows the same structure: confusion matrix training set, confusion matrix test set, classification report training set, classification report test set, and evaluation of multiple classifications. In Section 5.4, the confusion matrix, classification report, and evaluation are presented separately for the field test of 2018 and 2024.

5.1 ALG1 - GPS

The confusion matrix 5.1 summarizes the performance of ALG1 on the training set. The algorithm performs well in classifying bikes, walks, and cars but exhibits notable miss-classifications, particularly between bikes and walking. Bike is misclassified as walking in about 10% of cases, while walking is misclassified as bike in 12%. Further analysis suggests potential labeling errors, as misclassified bike tracks had an average speed of 3.23 km/h—an implausibly low cycling speed. Similarly, misclassified walk tracks had a mean speed of 8.71 km/h, higher than the typical walking speed of 4.4 km/h (Fitzpatrick et al. 2006). Additionally, walk tracks correctly classified had a speed standard deviation of 2.1 km/h. In comparison, those

misclassified as bikes had a higher standard deviation of 6.35 km/h, indicating data quality inconsistencies. Car is misclassified as walk in about 2% of the cases, with misclassified tracks averaging a speed of 3.72 km/h, again suggesting possible labeling errors. Public transport is frequently misclassified as car, which aligns with ALG1's reliance on GPS-based features that do not clearly distinguish between these modes.

Table 5.1 ALG1 confusion matrix: train set

	Predicted						
Observed	**Bike**	**Bus**	**Car**	**Metro**	**Train**	**Tram**	**Walk**
Bike	560	0	22	0	0	0	62
Bus	3	0	31	0	0	0	2
Car	70	0	812	0	2	0	30
Metro	5	0	30	0	0	0	8
Train	3	0	38	0	20	0	5
Tram	0	0	8	0	0	0	2
Walk	81	0	15	0	0	0	573

The confusion matrix in Table 5.2 for the test set shows consistent performance with the training set, particularly for the majority classes—bike, car, and walk—indicating good generalization. Bike is misclassified as walking in about 6% of the cases, while walking is misclassified as bike in about 11% (same trend as in the training set). Similarly, car miss-classifications such as bike or walk likely stem from user labeling errors rather than algorithmic shortcomings, having minimal impact on overall performance. Further analysis reveals that bike tracks misclassified as walking have an average speed of 4.86 km/h, suggesting labeling errors, as cycling at such low speeds is unlikely. Walks misclassified as bikes have a mean speed of 9.54 km/h—nearly double the average walking speed (4.4 km/h). Additionally, the standard deviation of speed for correctly classified Walk tracks is 2.04 km/h, compared to 6.08 km/h for misclassified ones, indicating potential data quality issues or labeling inconsistencies. Car is misclassified as walking in about 3% of cases, with these tracks averaging 3.22 km/h, again suggesting labeling errors. Misclassified car tracks as bike tracks occur in 7%, with an average speed of 13.4 km/h, likely due to low-quality data or user mislabeling.

Table 5.2 ALG1 confusion matrix: test set

	Predicted						
Observed	**Bike**	**Bus**	**Car**	**Metro**	**Train**	**Tram**	**Walk**
Bike	243	0	7	0	0	0	17
Bus	2	0	36	0	0	0	0
Car	25	0	315	0	0	0	12
Metro	0	0	3	0	0	0	0
Train	1	0	13	0	11	0	1
Tram	0	0	8	0	0	0	1
Walk	30	0	7	0	1	0	247

The classification report (Table 5.3) shows an overall accuracy of 82% in the training set. However, this metric does not account for class imbalance. The balanced accuracy is considerably lower at 41%, indicating weaker performance for minority classes despite strong results for majority classes. Bike, car, and walk demonstrate high precision (0.77–0.85), recall (0.86–0.89), and F1-score (0.82–0.87), suggesting effective classification with minimal false positives and negatives. These results indicate that ALG1 successfully differentiates these modes based on GPS data and the predefined rules. Train exhibits high precision (90%) but low recall (28%), leading to a low F1-score (43%). This suggests that while the model accurately classifies

train when predicted, it fails to identify many actual train instances, likely due to the rule-based thresholds. As expected, bus, metro, and tram have 0% across all metrics since these modes were indistinguishable from others using GPS features alone. Of 2,895 tracks, the algorithm classified 2,382 (82%). Subsequent models aim to improve classification coverage, i.e., reducing the number of multiple or unknown classifications.

Table 5.3 ALG1 performance: train set

Class	Precision	Recall	F1-Score	Support
Bike	0.78	0.87	0.82	644
Bus	0.00	0.00	0.00	36
Car	0.85	0.89	0.87	914
Metro	0.00	0.00	0.00	43
Train	0.91	0.28	0.43	66
Tram	0.00	0.00	0.00	10
Walk	0.84	0.86	0.85	669
Accuracy		0.82		2,382
Balanced Accuracy		0.41		2,382

The performance of ALG1 on the test set closely mirrors that of the training set (see Table 5.4). It correctly classified 980 out of 1,264 tracks, yielding an accuracy of 78%, a slight decrease from the training set. Bike, car, and walk maintain strong performance, with high precision (0.81–0.89), recall (0.87–0.91), and F1-score (0.85–0.88), demonstrating effective classification and generalization to unseen data. These results confirm ALG1's ability to differentiate these transport modes. Minor precision, recall, and F1-score improvements for these classes may stem from fewer mislabeled tracks in the test set. Train continues to exhibit high precision (92%) but low recall (42%), consistent with training results. The stable test set performance provides a solid foundation for refining the classification of modes that ALG1 could not identify, such as bus, metro, and tram.

Table 5.4 ALG1 performance: test set

Class	Precision	Recall	F1-Score	Support
Bike	0.81	0.91	0.86	267
Bus	0.00	0.00	0.00	38
Car	0.81	0.89	0.85	352
Metro	0.00	0.00	0.00	3
Train	0.92	0.42	0.58	26
Tram	0.00	0.00	0.00	9
Walk	0.88	0.87	0.88	285
Accuracy		0.83		980
Balanced Accuracy		0.44		980

The multiple classifications in the test set are shown in Table 5.5. They reveal no bus instances are included when predicting car and bus. Similarly, only cars and metros are classified in the car, bus, and metro grouping, indicating that buses should be excluded from this category. Likewise, in the car, bus, metro, and tram group, tram and metro appear misclassified, suggesting required threshold adjustments in future models. The car, metro, and tram group includes five of the seven classes, with the bike and walk also appearing, likely due to mislabeled tracks or inconsistencies in the GPS data. The car-train grouping remains the most reliable, with 41% of instances classified as car and 53% as train, indicating that the algorithm effectively differentiates these modes within this combination.

Table 5.5 Multiple classifications ALG1: test set

	Predicted					
Observed	(car, bus)	(car, bus, metro)	(car, bus, metro, tram)	(car, metro, tram)	(car, train)	unknown
Bike	0	0	3	19	0	0
Bus	0	0	5	0	0	0
Car	7	21	127	50	13	0
Metro	0	3	0	0	1	0
Train	0	0	1	1	17	0
Tram	0	0	0	0	0	0
Walk	1	0	6	6	1	1

ALG1 fails to classify public transport modes. Thus, ALG2 and ALG3 will focus on improving the classification of public transport modes.

5.2 ALG2 - GPS and decision tree

The decision tree used (results not shown) had the same difficulties in classifying public transport modes (bus, metro, tram) using only GPS data. However, the tree offered valuable insights for improving the train classification. Adjusting the 95th percentile speed threshold to 130 km/h improved train detection accuracy. Despite this improvement, issues from ALG1 persisted in ALG2, particularly the bidirectional misclassifications between bike and walk. The confusion matrices in Tables 5.6 and 5.7 show ALG2 accurately classified walk, bike, and car but struggled with public transport modes, often misclassifying bus and tram as car or walk. Train classification improved, supporting the idea that speed thresholds help differentiate specific modes. Notably, bike misclassifications as walk increases from 6% to 10% in the test set. While minor fluctuations in misclassification rates occur between ALG1 and ALG2, no major differences are observed, underscoring the persistent challenge of distinguishing public transport modes with GPS data alone.

Table 5.6 ALG2 confusion matrix: train set

	Predicted						
Observed	Bike	Bus	Car	Metro	Train	Tram	Walk
Bike	529	0	19	0	2	0	86
Bus	4	0	31	0	1	0	2
Car	59	0	797	0	20	0	49
Metro	0	0	19	0	1	0	24
Train	4	0	24	0	69	0	5
Tram	2	0	7	0	1	0	2
Walk	57	0	11	0	1	0	613

Table 5.7 ALG2 confusion matrix: test set

	Predicted						
Observed	**Bike**	**Bus**	**Car**	**Metro**	**Train**	**Tram**	**Walk**
Bike	221	0	7	0	0	0	24
Bus	0	0	36	0	0	0	1
Car	23	0	311	0	14	0	15
Metro	0	0	3	0	1	0	0
Train	2	0	6	0	33	0	1
Tram	0	0	8		0	0	1
Walk	13	0	3	0	1	0	268

In the classification reports (Tables 5.8 and 5.9), ALG2 achieves 82% accuracy on the training set and 84% on the test set. However, the balanced accuracy is much lower (48% and 49%, respectively), highlighting the issues with minority classes despite the strong performance of majority classes.

Table 5.8 ALG2 classification report: train set

Class	Precision	Recall	F1-Score	Support
Bike	0.81	0.83	0.82	636
Bus	0.00	0.00	0.00	38
Car	0.88	0.86	0.87	925
Metro	0.00	0.00	0.00	34
Train	0.73	0.68	0.70	102
Tram	0.00	0.00	0.00	12
Walk	0.78	0.80	0.84	682
Accuracy		0.82		2,439
Balanced Accuracy		0.48		2,439

Table 5.9 ALG2 classification report: test set

Class	Precision	Recall	F1-Score	Support
Bike	0.85	0.88	0.86	252
Bus	0.00	0.00	0.00	37
Car	0.83	0.86	0.84	363
Metro	0.00	0.00	0.00	4
Train	0.67	0.79	0.73	42
Tram	0.00	0.00	0.00	9
Walk	0.84	0.94	0.90	285
Accuracy		0.84		992
Balanced Accuracy		0.49		992

The algorithm performs well for Bike, Car, and Walk, with F1-scores of 0.86, 0.84, and 0.9, respectively – results similar to ALG1. ALG1 classifies 15 more Bike tracks, while ALG2 correctly classifies 11 more Car tracks. Walk performs slightly better in ALG2, possibly due to higher-quality test set data with 153 more observations on average. Public transport modes like buses, metro, and trams remain poorly classified, with zero precision, recall, and F1 scores, underlining the need for OSM data. Train classification improves considerably, with F1-scores of 0.7 (train) and 0.74 (test), reflecting the benefit of refining speed thresholds to isolate Train tracks. Although ALG2 shows minimal improvement over ALG1 for Walk, Bike, and Car, it notably enhances Train classification. Of 2,895 training cases, 2,439 are classified (84%). On the test set, 992 out of 1,264 tracks are correct (78%).

Table 5.10 shows the results of multiple classifications using ALG2 on the test set. Similar to ALG1, the multiple classifications often misgroup transport modes. For instance, the (car, bus) classification contains no bus tracks. However, ALG2 assigned 35 tracks to the 'Unknown' group (up from 1 in Algorithm ALG1), reducing the likelihood of forcing incorrect groupings.

Table 5.10 Multiple classifications ALG2: test set

	Predicted			
Observed	(car, bus)	(car, bus, metro, tram)	(car, metro, tram)	unknown
Bike	0	3	16	18
Bus	0	5	0	1
Car	7	146	44	10
Metro	0	3	0	0
Train	0	1	0	2
Tram	0	0	2	0
Walk	1	6	3	4

ALG2 did not show considerable improvements over ALG1, but it performs better in handling multiple classifications and identifying trains.

5.3 ALG3 - GPS, OSM, and decision tree

The confusion matrix in Table 5.11 shows that ALG3 improves public transport classification compared to ALG1 and ALG2. On the training set, the train achieved 79% accuracy, metro 61%, and tram 59%, highlighting the OSM features' value. The bus remains unclassified, with instances appearing in multiple classifications (shown in Table 5.15). Misclassifications between bike, walk, and car persisted but have been previously discussed. Public transport misclassification as the car dropped: Metro decreased from 73% (ALG1) and 43% (ALG2) to 9%, train from 57% and 24% to 11%, and bus from 86% and 82% to 46%. Similar improvements are found in the test set (Table 5.12), confirming that ALG3 made a step forward in accurately distinguishing public transport modes.

Table 5.11 ALG3 confusion matrix: train set

	Predicted						
Observed	Bike	Bus	Car	Metro	Train	Tram	Walk
Bike	560	0	28	0	6	0	62
Bus	3	0	6	0	2	0	2
Car	70	1	825	2	8	3	30
Metro	5	0	4	29	1	0	8
Train	3	0	11	2	80	0	5
Tram	0	0	7	0	0	13	2
Walk	81	0	31	2	3	2	573

Table 5.12 ALG3 confusion matrix: test set

Observed	Predicted						
	Bike	**Bus**	**Car**	**Metro**	**Train**	**Tram**	**Walk**
Bike	243	0	24	0	0	0	17
Bus	2	0	8	0	0	0	0
Car	25	0	360	2	2	2	12
Metro	0	0	1	5	1	0	0
Train	1	0	8	0	32	1	1
Tram	0	0	1	0	0	9	1
Walk	30	1	11	0	1	1	247

ALG3 achieves 84% accuracy on the training set and 85% on the test set, as shown in the classification reports in Tables 5.13 and 5.14. However, balanced accuracy is lower (65% and 70%), mainly due to challenges in classifying buses. Still, this is an improvement over ALG1 and ALG2, with balanced accuracy rising from 0.41 to 0.65 for training and from 0.48 to 0.7 for testing. Adding relative distance to routes as a feature increased the predictive performance for public transport. On the training set, the train achieved a precision/recall/F1 of 0.8/0.79/0.8, metro scores 0.83/0.62/0.71, and tram reaches 0.72/0.59/0.65. On the test set, the metro stabilizes at 0.71 for all metrics, the train's precision increases to 0.89 while recall drops to 0.74, and the tram's recall increases to 0.82, causing an F1 score of 0.75. Bike, walk, and car show consistent performance across sets. Car performance improves over previous algorithms, as public transport modes are no longer misclassified as cars. This algorithm increases test set precision, recall, and F1 by 0.04–0.05 over its predecessor. Walk and bike performance remains stable.

Table 5.13 ALG3 classification report: train set

Class	Precision	Recall	F1-Score	Support
Bike	0.78	0.85	0.81	656
Bus	0.00	0.00	0.00	13
Car	0.90	0.88	0.89	939
Metro	0.83	0.62	0.71	47
Train	0.80	0.79	0.80	101
Tram	0.72	0.59	0.65	22
Walk	0.84	0.83	0.84	692
Accuracy		0.84		2,470
Balanced Accuracy		0.65		2,470

Table 5.14 ALG3 classification report: test set

Class	Precision	Recall	F1-Score	Support
Bike	0.81	0.86	0.83	284
Bus	0.00	0.00	0.00	10
Car	0.87	0.89	0.88	403
Metro	0.71	0.71	0.71	7
Train	0.89	0.74	0.81	43
Tram	0.69	0.82	0.75	11
Walk	0.89	0.85	0.87	291
Accuracy		0.85		1,049
Balanced Accuracy		0.70		1,049

Table 5.15 shows the multiple classifications in the test set. There are 43 bus tracks, with 10 misclassified by the algorithm and 33 being multiple classifications. It is important to note that although ALG3 does not allow multiple classifications for bike, metro, train, tram, and walk

modes, it allows multiple classifications for buses and cars. This was done since, even with the inclusion of OSM data, it was challenging to distinguish between these two modes, as even with the OSM data, there was a lot of overlap.

Table 5.15 Multiple classifications ALG3: test set

	Predicted	
Observed	(car, bus)	unknown
Bike	5	0
Bus	33	0
Car	167	0
Metro	0	0
Train	2	0
Tram	0	0
Walk	7	1

5.4 Performance of ALG3

ALG3 was tested on the two external datasets described in Section 3.3, which were exclusively reserved for model evaluation, and none of their data was used during development. First, we report the results on the data collected during the first field test in 2018.

The confusion matrix in Table 5.16 provides insight into the classification performance of ALG3 when applied to the 2018 field test data. ALG3 performs well in identifying specific modes, particularly cars, which were correctly classified in 1,629 cases, and bikes, with 837 correct classifications. Walking is also relatively well recognized, with 449 correct classifications. However, there are notable misclassifications. One of the most considerable errors is frequently misclassifying walking as biking, occurring in 231 instances. Similarly, 61 bike tracks were mistakenly labeled as car tracks, suggesting some overlap in the distinguishing features of these modes. Additionally, 46 car tracks were incorrectly identified as bike tracks, further reinforcing the challenge of differentiating between these categories. Public transportation modes present additional challenges. Bus tracks, for instance, were rarely classified correctly, with most being labeled as car tracks, likely due to similarities in movement patterns. Metro and tram tracks have relatively low sample sizes, making it difficult to evaluate their classification performance. Overall, the results suggest that ALG3 struggled particularly with distinguishing between biking and walking, as well as between biking and driving.

Table 5.16 ALG3 confusion matrix: field test 2018

	Predicted						
Observed	Bike	Bus	Car	Metro	Train	Tram	Walk
Bike	837	4	61	4	1	2	19
Bus	2	0	14	0	0	1	0
Car	46	7	1629	7	13	7	30
Metro	0	0	0	11	0	0	2
Train	7	0	10	1	107	1	2
Tram	1	0	1	1	0	22	0
Walk	231	1	49	2	2	5	449

The classification report for ALG3 is shown in Table 5.17.

Table 5.17 ALG3 classification report: field test 2018

Class	Precision	Recall	F1-Score	Support
Bike	0.74	0.90	0.82	928
Bus	0.00	0.00	0.00	17
Car	0.92	0.94	0.93	1,739
Metro	0.42	0.85	0.56	13
Train	0.87	0.84	0.85	128
Tram	0.58	0.88	0.70	25
Walk	0.89	0.61	0.72	739
Accuracy		0.85		3,589
Balanced Accuracy		0.72		3,589

Overall, the model achieves an accuracy of 85%, indicating strong general performance. However, the balanced accuracy is lower at 72%, suggesting that some classes are underrepresented or misclassified more frequently than others. Cars are the best-classified category, with a precision of 0.92, a recall of 0.94, and an F1-score of 0.93, highlighting ALG3's reliability in identifying car tracks correctly. Bikes also perform well, with an F1-score of 0.82, benefiting from a high recall of 0.90, meaning most bike tracks are correctly identified, though some are misclassified. Walking, however, shows a weaker recall (0.61) despite a high precision of 0.89, indicating that while most predictions labeled as "Walk" are correct, many actual walking tracks are misclassified as other modes, particularly biking. Public transportation modes show mixed results. Train classification is relatively strong, with an F1-score of 0.85. Still, the metro and tram have lower performance, with the metro achieving only 42% precision. Still, a high recall of 85% means that most metro tracks are correctly identified, but many non-metro tracks are mistakenly labeled as metro. Tram classification follows a similar pattern, with a precision of 0.58 but a recall of 88%, suggesting frequent misclassification of other modes as tram. The most concerning result is for buses with an F1-score of 0.00, indicating that the model fails to classify any bus tracks correctly. This is likely due to their small representation in the dataset (only 17 samples) and potentially overlapping characteristics with other transport modes, particularly cars. In summary, while the model performs well for standard transport modes such as cars, bikes, and trains, it struggles with buses and some forms of public transport, likely due to data imbalance and feature similarities.

Table 5.18 shows the multiple classifications in the 2018 test set.

Table 5.18 Multiple classifications ALG3: field test 2018

	Predicted	
Observed	(car, bus)	unknown
Bike	22	6
Bus	42	0
Car	690	29
Metro	0	0
Train	1	2
Tram	2	0
Walk	29	5

The most frequent misclassification occurs with the (bus, car) category, where 690 car tracks were classified as a mix of bus and car, reinforcing the previously observed confusion between these two modes. 42 bus tracks were misclassified as bus and car, further highlighting the model's difficulty distinguishing between them. For walking, 29 instances were assigned both

bus and car labels, which suggests that some walking tracks might have shared features with short vehicle-based tracks. Biking also has 22 cases of this misclassification pattern. The 'unknown' category, where ALG3 failed to classify a track confidently, appears in a few cases (primarily for car tracks, with 29 instances). This finding suggests that while the model is generally confident in its predictions, specific movements still create uncertainty. The results indicate that bus and car classification remains challenging, as the model frequently assigns tracks to both categories simultaneously.

Second, the results on the data collected during the small-scale field test in 2024 are reported. The confusion matrix in Table 5.19 provides insight into the classification performance of ALG3 when applied to the 2024 field test data. In terms of performance, walking (58 correct classifications) and trams (20 correct classifications) show relatively strong classification accuracy, while other transport modes, such as bus (4 correct classifications) and metro (3 correct classifications), have weaker performance. A difference compared to the test on the 2018 field test data is the absence of the car category, which was previously the most frequent transport mode. Biking and walking still exhibit considerable confusion, with 14 walking tracks misclassified as bike tracks. However, compared to the 2018 field test data, when walking was frequently confused with biking and car tracks, there were fewer misclassifications across different transport modes, indicating improved differentiation. While some transport modes like trams and walking perform well, buses and metro remain problematic categories, and the absence of cars marks a substantial shift from the previous evaluation.

Table 5.19 ALG3 confusion matrix: field test 2024

				Predicted		
Observed	**Bike**	**Bus**	**Metro**	**Train**	**Tram**	**Walk**
Bike	9	0	0	0	0	0
Bus	1	4	0	0	0	0
Metro	0	0	3	0	1	1
Train	0	0	0	8	1	0
Tram	1	0	0	0	20	1
Walk	14	0	0	1	4	58

The classification report for ALG3 is shown in Table 5.20. The test on the 2024 field test data shows an overall accuracy of 80%, with a balanced accuracy of 83%, suggesting that the model is performing well across different transport modes. However, the number of observations per class is relatively low, making generalization more challenging. Among the individual transport modes, bike classification has a recall of 1.00 but a low precision of 0.36, meaning that while all actual bike tracks were correctly identified, many non-bike tracks were mistakenly labeled as bikes. Bus and metro classifications show strong precision (both at 1.00), but their recall values (0.80 and 0.60, respectively) indicate that some actual bus and metro tracks were missed. Train classification is balanced, with precision and recall at 0.89, suggesting consistent performance. Tram classification is also strong, with an F1 score of 0.83. Walking, the largest class, has a high precision (0.97) but a recall of only 0.75, meaning that while most predicted walking tracks are correct, some actual walking tracks are misclassified.

Table 5.20 ALG3 classification report: field test 2024

Class	Precision	Recall	F1-Score	Support
Bike	0.36	1.00	0.53	9
Bus	1.00	0.80	0.89	5
Metro	1.00	0.60	0.75	5
Train	0.89	0.89	0.89	9
Tram	0.77	0.91	0.83	22
Walk	0.97	0.75	0.85	77
Accuracy		0.80		127
Balanced Accuracy		0.83		127

Compared to the 2018 classification report, the overall accuracy (85% in 2018 vs. 80% in 2024) has slightly decreased, but balanced accuracy has improved (72% in 2018 vs. 83% in 2024). This finding suggests that while the overall classification performance has dropped slightly, the model treats different classes more evenly, reducing bias toward dominant categories. One crucial difference is the absence of car tracks in the 2024 data. In 2018, car classification was the strongest, with a high F1 score of 0.93. This change alters the classification dynamics, potentially explaining the shift in performance. Another key difference is in bike classification, where recall has improved from 0.90 (2018) to 1.00 (2024), but precision has dropped considerably from 0.74 to 0.36, leading to an overall lower F1 score. Walking classification has also changed, with a recall drop from 0.61 to 0.75, but precision has improved from 0.89 to 0.97, making its predictions more reliable. Overall, the test on the 2024 data results indicates an improvement in balanced accuracy, but at the cost of higher misclassification rates for some transport modes, particularly biking. Additionally, the absence of car tracks changes the classification task, making direct comparisons with 2018 somewhat difficult.

Table 5.21 shows the multiple classifications in the test set from 2024. Here, no multiple classifications were observed because of the absence of car tracks in the data. The 'unknown' category, where ALG3 failed to classify a track confidently, only occurred very few times. Thus, 127 out of the 134 tracks were classified.

Table 5.21 Multiple classifications ALG3: field test 2024

Observed	Predicted unknown
Bike	1
Bus	0
Metro	0
Train	0
Tram	5
Walk	1

6 Discussion

This paper describes the development of a deterministic rule-based algorithm to predict transport modes in a smart-travel survey using GPS and OSM data. The focus was on five key contributions:

- How accurately can manual rule-based models determine transport mode?
- Which GPS features are relevant to classify the transport mode accurately?
- Which Open Street Map (OSM) data features, next to the GPS features, are relevant to classify the transport mode accurately?
- Does the developed algorithm generalize to different app versions?
- Does the developed algorithm generalize to data from a different country?

Considering the first question, the developed algorithm (ALG3) achieved an overall balanced accuracy of 70% and an overall accuracy of 85% based on the test set taken from the development dataset. Considering the second and third questions, 183 features were considered (107 GPS features and 76 OSM features). Ultimately, only 13 features were included in ALG3. Speed-related features were the most important GPS features for distinguishing transport modes. The relative proximity to transport routes was the most useful OSM feature for transport mode classification. These results are consistent with the existing literature. Overall, the results of the developed ALG3 are considered promising, given the algorithm's simplicity. Regarding the fourth question, 'Does the developed algorithm generalize to different app versions?' it was found that when tested on external datasets exclusively reserved for model evaluation, the developed algorithm ALG3 generalizes well to these datasets. On the test set from the development dataset, balanced accuracy was 70%, and overall accuracy was 85%. Comparable results were found when tested on the 2018 dataset (72% balanced accuracy and 85% accuracy). From this, we conclude that ALG3 generalizes well to this data and, accordingly, to different app versions since the 2018 app was differently configured. When tested on the 2024 test set, 83% balanced accuracy and 80% accuracy were obtained. ALG3 generalizes this data well and to different app versions since the 2024 app was configured differently than the 2018 and 2022/2023 apps. Furthermore, considering the last question, these results show that ALG3, which was developed to classify transport modes in the Netherlands, generalizes to a different country (Germany). Again, these results are considered promising, given the algorithm's simplicity. However, whether a loss or gain in performance may be due to method effects (different app and sensor configurations) needs to be studied in the future.

This work builds on and improves on Smeets et al. (2019), enhancing public transport mode prediction without collapsing transport modes. ALG3 also achieves an overall accuracy of 85% as reported by Smeets et al. (2019). Smeets et al. (2019) did not report balanced accuracy. We speculate that the 85% reported by Smeets et al. (2019) are inflated because the train/test split did not consider the user, i.e., a user can be in both sets with different tracks. Compared to Gong et al. (2012) and Chen et al. (2010), who predicted five modes, the developed algorithm predicts seven modes with higher accuracy, especially for metro and train.

Unlike earlier studies using GPS loggers, this study relies on GPS data from smartphone sensors, reducing participant burden by automating transport mode detection. As noted by Smeets et al. (2019), many studies tested prediction feasibility but did not address automation – a gap this study fills with the CBS smart-travel app.

This study has some limitations. First, while OSM data was assumed reliable, its open-source nature makes it prone to human error, inconsistencies, and update delays, which affect feature quality.

Second, the CBS stop-track segmentation algorithm was assumed to work perfectly, but clusters exceeding 10 hours suggested room for improvement. These outliers were removed, but segmentation errors may persist.

Third, labeled tracks were treated as ground truth, though some users likely mislabeled tracks. Mislabeling may have impacted results, especially for tracks with characteristics that deviate from their labels.

Fourth, public transport tracks were underrepresented, potentially affecting threshold values in ALG3. More data on minority classes like metro, tram, bus, and train would allow for better feature tuning and threshold adjustments.

Fifth, the algorithm predicts seven standard transport modes but excludes the 'other' category, as it overlaps with primary modes and reduces classification accuracy. A simple user prompt could help manage this challenge, though the category adds limited value for transport planning.

Sixth, the algorithm only can classify single-mode tracks. In practice, multi-modal tracks will be partially misclassified with this algorithm. However, new training data, including such labels, is required to develop a multi-modal algorithm.

Seventh, unlike previous studies focusing solely on transport mode prediction, the CBS app collected general travel data from untrained users with minimal screening. While this limits direct comparisons to earlier research, the app's thorough data processing documentation is a strength.

Eight, a limitation is the partial subjectivity of the algorithms' construction. Accordingly, whether the algorithms are robust and generalize to future data must be evaluated. While it is conceivable that other researchers would construct a different version of the algorithm, the overall structure would likely remain broadly similar. Typically, classification begins with identifying walking segments. This is a common first step in the literature. The distinct position of cycling between motorized and non-motorized modes made it the next logical classification task. The classification order was less critical for the railway-related modes (tram, metro, train). Although this may change with more granular data in the future, it did not impact the current implementation. The final grouping of car and bus modes reflected the practical challenge of distinguishing between them; assigning multiple possible labels, in this case, seemed justified and consistent with standard practice. While some elements, such as the ordering of rail modes or specific thresholds, may be subjective, the overall classification structure appears robust and unlikely to differ substantially across independent implementations, given the same mode set. Future extensions involving a broader range of transport modes may introduce more structural variation, but under the current scope, such differences are expected to remain limited.

Finally, although some evidence of cross-country generalizability was demonstrated, it must be evaluated whether differences in traffic laws and regulations must be considered. For example, speed limits are relevant for GPS apps because speed features are very important for classification. When comparing the Netherlands, Germany, and Belgium, already considerable differences are observed for motorways: Netherlands: 100 km/h (6:00–19:00), 120/130 after; Germany: No general limit, but advisory 130 km/h; Belgium: 120 km/h (some 130 km/h zones).

The developed manual rule-based approach is efficient and easy to implement, making it a candidate for the CBS smart-travel app's goal of collecting, processing, and predicting transport modes and purposes. Its simplicity reduces costs, lowers computational requirements, and speeds up processing. While machine learning might improve predictions for some modes, the added computational costs and complexity may outweigh the benefits. Additionally, user prompts – which enhance prediction accuracy – are easier to integrate into a rule-based system, whereas machine learning models would likely require retraining to adapt to user preferences.

Future work

The future work section is organized using the MoSCoW prioritization method (Must have, Should have, Could have, Won't have yet but would like in the future):

Must Have: These are critical improvements that directly affect the performance and validity of the transport mode prediction model.

- Improve data coverage and quality
  - Since the OSM data may not be entirely reliable (coverage issues), future work should explore alternative sources like `hotosm.org`, `flosm.de`, or proprietary options (e.g., Google). Better coverage could strengthen mode-infrastructure relationships and improve prediction accuracy.
- Expand public transport training data
  - Limited public transport data made it difficult to fine-tune thresholds. Future data collection strategies must aim to gather more data on these transport modes. For these modes, more training data is urgently required.
- Refine stop-track segmentation
  - The CBS stop-track segmentation may need refinement, as some tracks contain long gaps or missing data. Future research should investigate how segmentation and GPS errors impact predictions.

Should Have: These enhancements are not essential for core functionality but would considerably improve the performance and validity of the transport mode prediction model.

- Redefine ambiguous mode categories:
  - Grouping unrelated modes under 'Other' reduces accuracy. For this reason, this category was excluded in this study. Future data collection strategies should aim to define more precise categories rather than combining transport modes in vague categories.
- Improve feature set with contextual data
  - Adding registry features or asking introductory questions (e.g., car ownership or city of residence) could help improve prediction quality. For example, if a user does not own a car, the algorithm could exclude the car as a potential mode. Furthermore, owning subscriptions for public transportation, the smartphone being logged in to WiFi from public transport, or using services such as Apple Carplay or Android Auto or other smartphone background services will likely improve the algorithm's performance. More informative features that distinguish clearly between the transport modes are required. No machine learning algorithm alone will solve these problems.
- Enable iterative prediction processes
  - The smart functionalities (segmentation, transport mode prediction, and stop purpose) are not single sequential steps. They should be part of an iterative procedure. Currently, these functionalities have been analyzed independently. Thus, if these functionalities have to be deployed in the field, more research is required on how to set up the iterative process.

Could Have: These are optional but beneficial additions that could further enhance prediction performance or user experience.

- Integrate OpenTripPlanner for enhanced prediction
  - Integrating OpenTripPlanner into a GPS tracking app would open up various use cases that enhance user experience and operational efficiency. It could enhance the prediction of public transport performance (particularly for buses) by integrating both scheduled and real-time data to generate dynamic, data-informed routing and timing estimates. By analyzing delays, vehicle locations, and service frequency patterns, OpenTripPlanner could allow for more accurate predictions of bus arrival times and potential disruptions. This is especially valuable for buses, which are more susceptible to variable traffic conditions than rail-based modes.
- Leverage short-term behavioral routines
  - Users often have consistent weekly routines (e.g., commuting). Learning these patterns through short-term data collection (e.g., 3 days) could improve classification accuracy. Rule-based systems could incorporate simple rules, e.g., assuming the same mode if the user travels the same route simultaneously.
- Add accelerometer data
  - Adding accelerometer data could help distinguish modes with similar speeds but different acceleration profiles (e.g., bus vs. car). GPS-derived acceleration features were uniform, suggesting accelerometer data could fill this gap.

Won't Have (Yet): These ideas are interesting for future exploration but are currently beyond the project scope or dependent on prior improvements. This is currently not applicable.

7 Conclusion

This paper developed a manual rule-based transport mode classification algorithm for the CBS smart-travel app, laying the groundwork for future automation. It demonstrated that a deterministic rule-based algorithm, using both GPS and OSM features, can achieve high prediction quality – reinforcing the value of rule-based approaches. The study showed that carefully designed rules can rival or surpass machine learning in prediction quality, offering greater efficiency and interpretability. However, respondents will always have the option to check and adjust the classifications.

By establishing this manual rule-based algorithm as a benchmark, the research provides a practical starting point for refining future models. The algorithm's simplicity and speed make it a viable option for real-world implementation, potentially alongside user prompts, to reduce respondent burden without compromising accuracy.

Acknowledgments

The views expressed in this paper are those of the author(s) and do not necessarily reflect the policies of Statistics Netherlands.

We thank Barry Schouten for a thoughtful review of an earlier version of this manuscript.

References

Ali, Y., P. Zachary, and F. Bilal (2020). "Ensemble convolutional neural networks for mode inference in smartphone travel survey". In: *IEEE Transactions on Intelligent Transportation Systems* 21.6, pp. 2232–2239. DOI: 10.1109/TITS.2019.2918923.

Ansarilari, Z. and A. Golroo (2015). "Automated transportation mode detection using smart phone application via machine learning: Case study mega city of Tehran". In: 94th Annual Meeting of the Transportation Research Board (TRB 2015), Washington, DC, USA, January 11-15, 2015. DOI: https://doi.org/10.3929/ethz-b-000722728.

Auld, J., C. Williams, A. Mohammadian, and P. Nelson (2009). "An automated GPS-based prompted recall survey with learning algorithms". In: *Transportation Letter* 1, pp. 59–79. DOI: doi.org/10.3328/TL.2009.01.01.59-79.

Bricka, S., J. Zmud, J. Wolf, and J. Freedman (2009). "Household travel survey with GPS". In: *Transportation Research Record Journal of the Transportation Research Board* 2105 (1). DOI: http://dx.doi.org/10.3141/2105-07.

Chen, C., H. Gong, C. Lawson, and E. Bialostozky (2010). "Evaluating the feasibility of a passive travel survey collection in a complex urban environment: Lessons learned from the New York City case study". In: *Transportation Research Part A: Policy and Practice* 44.10, pp. 830–840. DOI: https://doi.org/10.1016/j.tra.2010.08.004.

Dabiri, S. and K. Heaslip (2018). "Inferring transportation modes from GPS trajectories using a convolutional neural network". In: *Transportation Research Part C: Emerging Technologies* 86, pp. 360–371. DOI: https://doi.org/10.1016/j.trc.2017.11.021.

Dabiri, S., C.-T. Lu, K. Heaslip, and C. K. Reddy (2019). "Semi-supervised deep learning approach for transportation mode identification using GPS trajectory data". In: *IEEE Transactions on Knowledge and Data Engineering* 32.5, pp. 1010–1023. DOI: 10.1109/TKDE.2019.2896985.

Feng, T. and H. J. P. Timmermans (2016). "Comparison of advanced imputation algorithms for detection of transportation mode and activity episode using GPS data". In: *Transportation Planning and Technology* 39 (2). DOI: 10.1080/03081060.2015.1127540.

Filip, B., L. Hugo, and v. O. Peter (2013). "Transportation mode-based segmentation and classification of movement trajectories". In: *International Journal of Geographical Information Science* 27.2, pp. 385–407. DOI: 10.1080/13658816.2012.692791.

Fitzpatrick, K., M. A. Brewer, and S. Turner (2006). "Another look at pedestrian walking speed". In: *Transportation Research Record* 1982.1, pp. 21–29. DOI: 10.1177/0361198106198200104.

Fourie, J. J. (2025). *Rules for transport mode determination in smart travel surveys*. Master Thesis – Statistics and Data Science University Leiden.

Gillis, D., A. J. Lopez, and S. Guatama (2024). "An evaluation of smartphone tracking for travel behaviour studies". In: *ISPRS International Journal of Geo-Information* 12.8, p. 335. DOI: 10.3390/ijgi12080335.

Gong, H., C. Chen, E. Bialostozky, and C. T. Lawson (2012). "A GPS/GIS method for travel mode detection in New York City". In: *Computers Environment and Urban Systems* 36.2, pp. 131–139. DOI: 10.1016/j.compenvurbsys.2011.05.003.

Gonzalez, P., J. Weinstein, S. Barbeau, M. Labrador, P. Winters, N. Georggi, and R. Perez (2010). "Automating mode detection for travel behaviour analysis by using global positioning systems-enabled mobile phones and neural networks". In: *IET Intelligent Transport Systems* 4 (1), pp. 37–49.

Gootzen, Y., J. Klingwort, and B. Schouten (2025). *Data quality aspects for location-tracking in smart travel and mobility surveys*. Discussion paper, Statistics Netherlands. DOI: http://dx.doi.org/10.13140/RG.2.2.33690.76480.

Harding, C., A. F. Imani, S. Sikukenthiran, E. J. Miller, and K. N. Habib (2021). “Are we there yet? Assigning smartphone apps as full-fledged tools for activity-travel surveys”. In: *Transportation* 48, pp. 2433–2460. DOI: https://doi.org/10.1007/s11116-020-10135-7.

Klingwort, J., Y. Gootzen, and J. Fourie (2025a). *Development and performance of a transport mode classification algorithm for smart surveys*. Technical report. Smart Survey Implementation (SSI). Report number: WP3: Developing Smart Data Microservices. DOI: 10.13140/RG.2.2.30160.83203.

Klingwort, J., Y. Gootzen, M. Kompier, and V. Toepoel (2025b). *How smart are smart travel surveys? Evaluating trip segmentation, travel motive, and travel mode predictions*. Conference paper. Mobile Apps and Sensors in Surveys (MASS).

Klingwort, J., Y. Gootzen, D. Remmerswaal, and B. Schouten (2025c). “Algorithms versus survey response: Comparing a smart travel and mobility survey with a web diary”. In: *Transportation Research Interdisciplinary Perspectives* 31, p. 101436. DOI: https://doi.org/10.1016/j.trip.2025.101436.

Lei, G., M. Takayuki, Y. Toshiyuki, and S. Hitomi (2014). “Deriving personal trip data from GPS data: A literature review on the existing methodologies”. In: *Procedia - Social and Behavioral Sciences* 138. The 9th International Conference on Traffic and Transportation Studies (ICTTS 2014), pp. 557–565. DOI: https://doi.org/10.1016/j.sbspro.2014.07.239.

Li, L., J. Zhu, H. Zhang, H. Tan, B. Du, and B. Ran (2020). “Coupled application of generative adversarial networks and conventional neural networks for travel mode detection using GPS data”. In: *Transportation Research Part A: Policy and Practice* 136, pp. 282–292. DOI: doi.org/10.1016/j.tra.2020.04.005.

Markos, C. and J. J. Yu (2020). “Unsupervised deep learning for GPS-based transportation mode identification”. In: *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1–6. DOI: 10.1109/ITSC45102.2020.9294673.

McCool, D., P. Lugtig, O. Mussman, and B. Schouten (2021). “An app-assisted travel survey in official statistics: Possibilities and challenges”. In: *Journal of Official Statistics*, pp. 149–170. DOI: 10.2478/jos-2021-0007.

McCool, D., P. Lugtig, and B. Schouten (2024). “Maximum interpolable gap length in missing smartphone-based GPS mobility data”. In: *Transportation* 51.1, pp. 297–327. DOI: 10.1007/s11116-022-10328-2.

Moher, D., A. Liberati, J. Tetzlaff, and D. G. Altman (2009). “Preferred reporting items for systematic reviews and meta-analyses: The PRISMA Statement”. In: *Annals of Internal Medicine* 151.4, pp. 264–269. DOI: 10.7326/0003-4819-151-4-200908180-00135.

Murakami, E., R. Griffiths, A. J. Richardson, and M. E. H. Lee-Gosselin (2000). *Travel survey*. URL: https://onlinepubs.trb.org/Onlinepubs/millennium/00135.pdf.

Prelipcean, A. C., G. Gidófalvi, and Y. O. Susilo (2017). “Transportation mode detection – an in-depth review of applicability and reliability”. In: *Transport Reviews* 37.4, pp. 442–464. DOI: 10.1080/01441647.2016.1246489.

Rasmussen, T. K., J. B. Ingvardson., K. Halldórsdóttir, and O. A. Nielsen (2009). “Improved methods to deduct trip legs and mode from travel surveys using wearable GPS devices: A case study from the Greater Copenhagen area”. In: *Computers, Enviroment and Urban Systems* 54. DOI: doi.org/10.1016/j.compenvurbsys.2015.04.001.

Remmerswaal, D., B. Schouten, J. Bakker, J. Heuvel, and J. Klingwort (2025). *A smart travel survey: What is the role of the respondent?* Discussion paper, Statistics Netherlands. DOI: 10.13140/RG.2.2.16084.31364.

Rezaie, M., Z. Patterson, J. Y. Yu, and A. Yazdizadeh (2017). “Semi-supervised travel mode detection from smartphone data”. In: pp. 1–8. DOI: 10.1109/ISC2.2017.8090800.

Richardson, J. A. (2000). *Behavioural mechanisms of non-response in mailback surveys*. DOI: 10.3141/1855-24.

Sadeghian, P., J. Hakansson, and X. Zhao (2021). “Review and evaluation of methods in transport mode detection based on GPS tracking data”. In: *Journal of Traffic and Transportation Engineering* 8.4, pp. 467–482.

Sadeghian, P., X. Zhao, A. Golshan, and J. Håkansson (2022). “A stepwise methodology for transport mode detection in GPS tracking data”. In: *Travel Behaviour and Society* 26, pp. 159–167.

Safi, H., M. Mesbah, and L. Ferreira (2014). “Smartphone-assisted travel surveys: A smart way for transport planning”. In: *32nd Conference of Australian Institutes of Transport Research*.

Sauerländer-Biebl, A., E. Brockfeld, D. Suske, and E. Melde (2017). “Evaluation of a transport mode detection using fuzzy rules”. In: *Transportation Research Procedia*. Vol. 25, pp. 591–602. DOI: 10.1016/j.trpro.2017.05.444.

Schouten, B., D. Remmerswaal, A. Elevelt, J. de Groot, J. Klingwort, T. Schijvenaars, M. Schulte, and M. Vollebregt (2024). *A smart travel survey: Results of a push-to-smart field experiment in the Netherlands*. Discussion paper, Statistics Netherlands. DOI: http://dx.doi.org/10.13140/RG.2.2.30248.38404.

Schuessler, N. and K. W. Axhuasen (2009). “Processing raw data from global positioning systems without additional information”. In: *Transportation Research Record Journal of the Transportation Research Board* 2105 (1), pp. 28–36. DOI: 10.3141/2105-04.

Schönfelder, S., K. W. Axhausen, N. Antille, and M. Bierlaire (2002). “Exploring the potentials of automatically collected GPS data for travel behaviour analysis”. In: *Arbeitsberichte Verkehrs- und Raumplanung 124* 13.13, pp. 155–179. DOI: 10.3929/ethz-a-004403386.

Smeets, L., P. Lugtig, and B. Schouten (2019). *Automatic travel mode prediction in a national travel survey*. Discussion paper, Statistics Netherlands.

Stenneth, L., O. Wolfson, P. S. Yu, and B. Xu (2011). “Transportation mode detection using mobile phones and GIS information”. In: GIS ’11. Chicago, Illinois: Association for Computing Machinery, pp. 54–63. ISBN: 9781450310314. DOI: 10.1145/2093973.2093982.

Vij, A. and K. Shankari (2015). “When is big data big enough? Implications of using GPS-based surveys for travel demand analysis”. In: *Transportation Research Part C, Emerging Technologies* 56, pp. 446–462. DOI: 10.1016/j.trc.2015.04.025.

Wolf, J., M. Loechl, J. Myers, and C. Arce (2001). *Trip rate analysis in GPS-enhanced personal travel surveys*. URL: https://web.archive.org/web/20200709184639id_/http://www.isctsc.cl/archivos/2001/Wolf.pdf.

Wolf, J. (May 2000). “Using GPS data loggers to replace travel diaries in the collection of travel data”. Georgia Institute of Technology. PhD thesis.

Wu, L., B. Yang, and P. Jing (2016). “Travel mode detection based on GPS raw data collected by smartphones: A systematic review of the existing methodologies”. In: *Information* 7.4, p. 67. DOI: 10.3390/info7040067.

Xiao, G., Z. Juan, and C. Zhang (2015). “Travel mode detection based on GPS track data and Bayesian networks”. In: *Computers Environment and Urban Systems* 54. DOI: doi.org/10.1016/j.compenvurbsys.2015.05.005.

Yu, J. J. Q. (2020). “Semi-supervised deep ensemble learning for travel mode identification”. In: *Transportation Research Part C: Emerging Technologies* 112, pp. 120–135. DOI: doi.org/10.1016/j.trc.2020.01.003.

Zahroh, S., P. Lugtig, Y. Gootzen, J. Klingwort, and B. Schouten (2025). *Predicting trip purpose in a smartphone-based travel survey*. Discussion paper, Statistics Netherlands. DOI: http://dx.doi.org/10.13140/RG.2.2.26114.80329.

Appendices

A GPS features

- **Speed**
  - Mean: Average speed over data points.
  - Standard Deviation: Variability of speed values.
  - Median: Middle speed value.
  - Minimum / Maximum: Lowest / highest speed observed.
  - Inter-quartile Range (IQR): Range between 25th and 75th percentile speeds.
  - Skewness / Kurtosis: Measure of asymmetry / extreme deviations in speed distribution.
  - 95th / 85th Percentile: Speed threshold below which 95% / 85% of observations fall.
  - Proportion at Various Speed Levels: Proportion of time spend during the track in different speed ranges (0-5, 5-15, 15-30, 30-50, 50-80, 80-120, 120+ km/h).
- **Acceleration, Jerk, Snap**
  - Mean, Median, Standard Deviation, Minimum, Maximum.
  - Inter-quartile Range (IQR).
  - Skewness, Kurtosis.
  - 95th / 85th Percentile thresholds.
- **Low Speed Intervals**
  - Total duration, mean, median, standard deviation, minimum, maximum, and IQR of low-speed intervals.
  - Number of low-speed intervals, valid intervals, and long intervals (above 60 sec).
  - Percentage of total time spent in low-speed intervals.
- **Distance**
  - Mean, Median, Standard Deviation, Minimum, Maximum, and IQR.
  - Total Distance Covered.
  - Average Distance per Low Speed Interval.
- **Bearing**
  - Mean, Median, Standard Deviation, Minimum, Maximum, and IQR.
  - Skewness, Kurtosis, Mode, Count.
  - Change in Bearing statistics: Mean, Median, Standard Deviation, IQR, Skewness, Kurtosis, Count, Minimum, Maximum.
- **Bearing Change Rate**
  - Mean, Standard Deviation, Minimum, Maximum.
- **Accuracy**
  - Mean, Median, Standard Deviation, Minimum, Maximum.
  - Percentage of Accurate Points (within 5 meters).
- **Time**
  - Total Travel Time (hours).
  - Trip Length (seconds, minutes, hours).
- **Sampling**
  - Total Data Points.
  - Average, Maximum, Minimum Sampling Frequency, and Variance.
  - Duration of Sampling Period (hours).
  - Number of Significant Gaps.
  - Average Time Between Samples.

B OSM features

- **Parking**
  - Distance to parking, charger, bike parking, bike rental, and parking entrance at the start and end of a track.
- **Traffic**
  - Number of traffic circles, crossings, junctions, street lamps, roundabouts, traffic signals, speed cameras, and stops (both absolute and normalized by track size).
- **Proximity**
  - Minimum, maximum, mean, and standard deviation of distance to bike, bus, metro, tram, and train routes.
  - Distance to nearest train, bus, metro, and tram stations at the start and end of a track.
  - Distance from the 50th percentile point of a track to the nearest bus, metro, train, and tram routes.
- **Counts**
  - Number of bus, metro, train, tram, and bike routes (both absolute and normalized by track size).
  - Number of bus, metro, train, and tram stations (both absolute and normalized by track size).

C Python code for algorithms

ALG1

Listing 1 Python code for ALG1

```python
def ALG1(df):
    def apply_classification(row):

        modes = []
        # Walking
        if (row['speed_p95'] < 13):
            modes.append('walk')
        # Bike
        if (13 < row['speed_p95'] < 30):
            modes.append('bike')
        # Car
        if (30 < row['speed_p95'] < 140):
           modes.append('car')
        # Bus
        if (40 < row['speed_p95'] < 80 and 5>row['total_distance']):
            modes.append('bus')
        # Train
        if (row['speed_p95'] > 65 and row['prop_very_high_speed']>10):
            modes.append('train')
        # Metro
        if (35 < row['speed_p95'] < 70 and 5>row['total_distance']):
            modes.append('metro')
        # Tram
        if (30 < row['speed_p95'] < 60 and 5>row['total_distance']):
           modes.append('tram')
        return modes if modes else ['unknown']
    df['modes'] = df.apply(apply_classification, axis=1)
    return df
```

ALG2

Listing 2 Python code for ALG2

```
def ALG2(df):
    def apply_classification(row):
        modes = []
        # Walking
        if (row['speed_p85'] < 12):
            modes.append('walk')
        else:
            # Bike
            if (12 < row['speed_p85'] < 25):
                modes.append('bike')
            else:
                # Train
                if (row['prop_very_high_speed']>10 or row['speed_p95']>130):
                            modes.append('train')
                else:
                        #Car
                        if (30<row['speed_p95'] < 125):
                            modes.append('car')
                        # Bus
                        if (40 < row['speed_p95'] < 80  and 5>row['total_distance']):
                            modes.append('bus')
                        # Metro
                        if (30 < row['speed_p95'] < 70 and 5>row['total_distance']):
                            modes.append('metro')
                        # Tram
                        if (30 < row['speed_p95'] < 70 and 5>row['total_distance']):
                            modes.append('tram')
        return modes if modes else ['unknown']
    df['modes'] = df.apply(apply_classification, axis=1)
    return df
```

ALG3

Listing 3 Python code for ALG3

```python
def ALG3(df):
    def apply_classification(row):
        modes = []
        # Walking
        if (row['speed_p95'] < 13):
            modes.append('walk')
        else:
            # Bike
            if (13 < row['speed_p95'] < 30):
                modes.append('bike')
            else:
                # Tram
                if(row['min_distance_tram']<0.5 and row['std_distance_tram']<250 or row['mean_distance_tram']<100):
                    modes.append('tram')
                else:
                    # Metro
                    if(row['min_distance_metro']<1.5 and row['std_distance_metro']<450 or row['mean_distance_metro']<100):
                        modes.append('metro')
                    else:
                        # Train
                        if (row['min_distance_train']<0.05 or row['std_distance_train']<25 or row['mean_distance_train']<100):
                            modes.append('train')
                        else:
                            # Bus
                            if(row['std_distance_bus']<120 or row['mean_distance_bus']<40 or row['min_distance_bus']<0.015):
                                modes.append('bus')
                            # Car
                            if (30 < row['speed_p95'] < 140):
                                modes.append('car')
        return modes if modes else ['unknown']
    df['modes'] = df.apply(apply_classification, axis=1)
    return df
```

Table C.1 Comparison of transportation mode classification algorithms

Aspect	ALG1	ALG2	ALG3
Input features	– speed_p95 – total_distance – prop_very_high_speed	– speed_p85, speed_p95 – total_distance – prop_very_high_speed	– speed_p95 – min, stdv, mean distance to stations
Modes detected	walk, bike, car, bus, train, metro, tram, unknown	Same as ALG1	Same as ALG1
Classification logic	Thresholds based on speed and distance only	More refined thresholds using two speed percentiles and high-speed proportion	Uses proximity to known infrastructure (stations/stops) in addition to speed
Handling of multiple modes	removed multiple classifications from final output; stored separately	Same as ALG1	multiple classifications allowed only for bus and car; stored separately
Unique criteria	Simple logic with clear speed cutoffs	Adds `speed_p85` and high-speed check for train	Uses geospatial proximity metrics
Final output	`modes` column in dataframe + optional separate list for multiple classifications	Same as ALG1	Same as ALG1

Colophon

Publisher
Statistics Netherlands
Henri Faasdreef 312, 2492 JP The Hague
www.cbs.nl

Prepress
Statistics Netherlands, Grafimedia

Design
Edenspiekermann

Information
Telephone +31 88 570 70 70, fax +31 70 337 59 94
Via contact form: www.cbs.nl/information