



**HAL**  
open science

# **Android Application-based Automated Prediction for Green Transportation Mode Detection: Architecture, Challenges and Results**

Ilhem Gharbi, Denis Faure-Vincent, Hassen Fourati, Nicolas Vuillerme

## **► To cite this version:**

Ilhem Gharbi, Denis Faure-Vincent, Hassen Fourati, Nicolas Vuillerme. Android Application-based Automated Prediction for Green Transportation Mode Detection: Architecture, Challenges and Results. *Journal of Intelligent Transportation Systems: Technology, Planning, and Operations*, 2025, ?, <10.1080/15472450.2025.2576935>. <hal-05355698>

**HAL Id: hal-05355698**

**<https://hal.science/hal-05355698v1>**

Submitted on 9 Nov 2025

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons CC BY 4.0 - Attribution - International License

# Android Application-based Automated Prediction for Green Transportation Mode Detection: Architecture, Challenges and Results

Ilhem Gharbi <sup>a,c</sup>, Denis Faure-Vincent <sup>b</sup>, Hassen Fourati <sup>a</sup>, Nicolas Vuillerme <sup>c,d</sup>

<sup>a</sup>Univ. Grenoble Alpes, CNRS, Inria, Grenoble INP, GIPSA-Lab, Grenoble, 38000, France ,  
ilhem.gharbi@grenoble-inp.fr, hassen.fourati@grenoble-inp.fr

<sup>b</sup>Univ. Grenoble Alpes, GIPSA-Lab, Grenoble INP, 38000, France,  
denis.faure-vincent@grenoble-inp.fr

<sup>c</sup>Univ. Grenoble Alpes, AGEIS, Grenoble, 38000, France, ;  
nicolas.vuillerme@univ-grenoble-alpes.fr

<sup>d</sup>Institut Universitaire de France, Paris, France

## ARTICLE HISTORY

Compiled October 8, 2025

## **ABSTRACT**

Implementing an efficient Green Intelligent Transportation System (ITS) is essential for modern cities. An effective approach is to target users' Transportation Mode Detection (TMD) using smartphone sensors combined with Machine Learning (ML) models. We focus on green urban transportation modes, excluding cars and motorcycles, to emphasize eco-friendly alternatives, and to improve urban mobility planning. In this paper, we outline the development of an Android application, designed for two complementary purposes: 1) to collect data from smartphone sensors (accelerometer, gyroscope, magnetometer, and Global Positioning System (GPS)) and 2) to predict four green transportation modes (walk, bike, public transport and kick scooter) based on accelerometer and gyroscope data only. GPS data are omitted in this paper, to address privacy concerns and energy consumption. Unlike existing works that focused solely on modeling and classification in TMD, the main contribution of the paper goes beyond and provides moreover a detailed overview of the implementation of the mobile application, as well as the challenges encountered during data acquisition and practical solutions we applied and the entire automatic cycle from data acquisition to classification, prediction and display of results on the application. Developing this application and its automated pipeline with a reliable client-server communication presented many technical challenges, requiring reliable data transfer and robust design to ensure good performances. The results obtained show good predictions, providing the user with an efficient tool to evaluate their sustainable mobility habits. Encouraging such modes contributes to ease traffic congestion, and more sustainable urban lifestyles.

## **KEYWORDS**

Android application, data collection, machine learning, classification, green transportation mode detection, prediction, automated process.

## 1. Introduction

With the expansion of communication technologies and geolocation services, Intelligent Transportation Systems (ITS) have emerged as a key component of modern smart cities, offering solutions to resolve urban issues such as pollution, safety, congestion, etc. ITS provide several advantages, including the optimization of traffic flow and the reduction of travel time (Banani Ardecani et al., 2025; Lu et al., 2021) and fuel consumption through real-time route planning (Lu et al., 2025; Mahmoud et al., 2022; Golcarenenrenji et al., 2024). By collecting data from smartphone sensors, ITS enhance the overall efficiency and reliability of urban transportation networks (Zhuang and Mao, 2025; Soltani et al., 2025). Additionally, they facilitate better integration between many modes of transport, enhancing multimodal mobility and offering users more sustainable travel options.

Transportation Mode Detection (TMD) is considered as a crucial component of ITS, due to its ability to inform urban planning and policymaking, improve traffic management, and promote sustainable mobility strategies. Several studies have focused on TMD using Global Positioning System (GPS) or inertial sensors (accelerometer, gyroscope, and magnetometer), a large choice of Machine Learning (ML) methods and public datasets, like Sussex-Huawei Locomotion (SHL) (Wang, Gjoreski, Ciliberto, Mekki, Valentin and Roggen, 2018; Merikhipour et al., 2025; Erdelić et al., 2025), Geolife (Zheng et al., 2010; Yang et al., 2024; Wang et al., 2025), United States transport mode detection (US-TMD) (Merikhipour et al., 2025; Carpineti et al., 2018; Fourez et al., 2025), etc. For instance, authors in (Wang et al., 2020) developed a deep learning framework to predict trip destination using bike Chattanooga trip data (*Bike Chattanooga Trip Data*, 2025). In Merikhipour et al. (2025), authors developed a TMD model called Transductive Long Short-Term Memory (TLSTM) that enhances LSTM performances. The model is able to capture subtle temporal patterns and prioritize data points similar to those in the test set. The model achieved an F1-score of 91% in detecting ten transportation modes (walk, still, run, motorcycle, bike, bus, car, train, subway, and high speed rail) using US-TMD (Carpineti et al., 2018) and SHL (Drosouli et al., 2021) datasets. In (Alam, Haque, Hassan, Huda, Hassan, Strickland and AlQahtani, 2023), authors presented a convolution neural network (CNN) model to identify users pattern and classify five modes of transportation (still, walk, car, bus, and train) using smartphone’s inertial sensor system. They yield an F1-score of up to 93%. In (Tang and Cheng, 2023), researchers proposed an approach called Feature Pyramid biLSTM (FPbiLSTM) using the SHL dataset (Gjoreski et al., 2018). They yield an accuracy of 95% and an F1-score of 94% for eight transportation modes. Authors in (Alaoui et al., 2022) explored the use of inertial measurement units (IMU) to classify seven transportation modes (still, walk, bike, bus, tram, stairs, and elevator). The used IMU is the Gaitup Physilog5<sup>1</sup> which includes a triaxial accelerometer, a triaxial gyroscope, and a barometer (Alaoui et al., 2021). Wang et al. (Wang and Roggen, 2019) and Demagh et al. Demagh et al. (2023) validated the efficiency of sound based methods in distinguishing between close transportation modes, like trains and subways. They obtained an accuracy of 86% using the SHL dataset (*SHL Dataset - Sussex-Huawei Locomotion Dataset*, n.d.). Image captured by wearable cameras has also been explored by Richoz et al. in (Richoz et al., 2019). They obtained an F1-score of 52%. Recent studies have investigated multimodal fusion strategies to enhance TMD accuracy. For example, Richoz et al. (Richoz et al.,

---

<sup>1</sup><https://physilog.com/>

2020) proposed a deep-learning framework based on sound, images, and IMU data. By combining the predictions from all these sensors, they obtained an F1-score of 95% using the SHL dataset to recognize eight transportation modes.

Although there are various TMD approaches developed until now with different sensors and classification methods and they have made solid progress by using public datasets and modern methods, they often concentrated solely on model performance, with a limited focus on practical deployment challenges. In order to fill this gap, our contribution in this paper focuses on real-world usage by developing a lightweight Mobidou Android application, well-suited for widespread adoption on mobile platforms. This application uses smartphone sensors such as accelerometer, gyroscope, magnetometer, and GPS. The paper details the full development pipeline from the design of the application and data acquisition to preprocessing, modeling, and experimental evaluation while addressing the challenges of real-world development and deployment. The aim of our work is to predict four green transportation modes—walking, biking, public transport, and kick scooter—using a minimal number of sensors. By "green" we refer to environmentally friendly modes of transportation that involve low or no fuel consumption and minimal environmental pollution. The study was conducted in Grenoble, France, a city already committed to environmental initiatives and actively promoting the use of sustainable transportation modes. Walking, cycling, and using a kick scooter are clearly low-impact modes. Their use is encouraged, as they help reduce reliance on private cars, thereby decreasing traffic congestion and lowering carbon emissions. In the same spirit, public transport (tram and bus) operates primarily on electricity and produces less direct pollution. More broadly, promoting public transport helps reduce dependence on private vehicles and motorcycles, contributing to lower congestion and environmental impact. This paper focuses on urban transportation modes and does not currently include subways, however this mode could be incorporated in future work as part of the public transport category. Cars and motorcycles are excluded in order to promote healthier lifestyles (by encouraging physical activity), reduce carbon emissions, and support more sustainable urban mobility. To achieve this, we used only accelerometer and gyroscope sensors to classify the four targeted modes. The magnetometer was excluded due to its sensitivity to magnetic disturbances commonly found in urban environments and its limited added value in distinguishing between these transportation modes. GPS data was also omitted for two main reasons: 1) To preserve users' privacy. By not collecting location information, we ensure the anonymity of the data. All participants in the dataset and experiments strictly requested not to record their GPS data. 2) To better preserve smartphone battery life. It is well known that the GPS sensor consumes significantly more energy, as highlighted in the literature (Lo'ai et al., 2016; Choi, 2013). Hence, the proposed classification approach is GPS-free, and the results were very promising. However, our Android application is designed to optionally allow GPS data collection if needed. Several studies have successfully used GPS for various purposes (Gao et al., 2025; Sutanto et al., 2020), such as estimating the distance traveled for each mode during a trip, as shown in (Huang et al., 2021; Sadeghian et al., 2021).

Following the introduction section, the paper is structured into six sections. Section 2 gives an overview of the existing scientific smartphone applications in the literature. Section 3 introduces the developed smartphone application and the challenges encountered during recording. Section 4 describes the dataset, data filtering, segmentation, feature extraction and classification. Section 5 discusses the main experimental results.

Finally, Section 6 resumes this work and proposes some perspectives.

## 2. State of the art

Initial mobile applications designed for mobility purposes were developed for navigation and location-based services. Recent applications focused on integrating traffic data collection, travel information, and mobility behavior studies (Gharbi et al., 2024; Petraki et al., 2020; Ghahramani et al., 2017; Deilami et al., 2020; Wang, Luo, Zhao, Qin, Zhao and Chen, 2018; Dutzik et al., 2013). In this section, we focus on smartphone-based applications specifically designed for TMD tasks. Numerous works used smartphone applications for data collection and transportation mode classification. For instance, authors in (Sharma et al., 2020) proposed a TMD model to classify seven transportation modes (still, walk, bike, bus, car, train, and subway) using partially observed sequences obtained from smartphone sensor data via an Android application. Similarly, in (Lorintiu and Vassilev, 2016), authors proposed a system based on Random Forest (RF) and Discrete Hidden Markov Model (DHMM) to classify eight modes of transportation (still, walk, run, bike, road, rail, plane, and others) using a combination of accelerometer, magnetometer, and GPS data gathered by an Android application. In (Hasan et al., 2022), data from both smartwatches and smartphones were collected to predict four transportation modes (bike, auto, bus, and walk). Other methods have explored the combination of additional data sources. In (Bjerre-Nielsen et al., 2020), authors proposed the use of Bluetooth data and GPS collected by an Android application. They obtained an overall accuracy of 89% and an average F1-score of 83% in detecting four transportation modes (car, bus, metro, and train). In (Molina-Campoverde et al., 2024), a mobile application was developed to collect multi-modal sensor data (GPS, altitude and speed). Authors yielded an accuracy of 94.9% in detecting six transportation modes (walk, bike, tram, bus, taxi, and vehicle) using a Decision Tree (DT) algorithm. In (Coroamă et al., 2019), the authors presented an Android application that collects gyroscope, accelerometer, gravity, GPS, and Bluetooth data from users. The collected data is used to classify seven transportation modes (tram, bus, walk, train, car, bike, and wait) using the RF model. In (Hedemalm et al., 2021), the authors designed interactive applications to influence transportation choices, and others as (Zhou et al., 2022) focused on trip validation using web interfaces.

While many applications exist, they are often designed with the primary goal of data collection rather than prediction and results display. Our work introduces a practical and energy-efficient solution to predict four green transportation modes on smartphones, including walking, biking, kick scooter, and public transport (bus/tram) to support sustainable mobility in urban areas. We propose an efficient Android application for TMD that goes beyond model performance. We present a complete pipeline encompassing system design, data acquisition, preprocessing, and predictions calculated on the server and displayed on the smartphone. The application is designed to address a wide range of user needs, with a practical focus on transportation mode classification. For example, individuals aiming to maintain a regular walking routine can monitor the duration and proportion of their walking activity. Similarly, users of bike-sharing systems or public transportation can assess whether they are optimizing their subscriptions. It is important to note that the Android application does not perform classification in real time. Instead, sensor data are recorded during users' trips, and predictions are processed offline. By selecting a specific time interval (start and end

times), users receive a summary indicating the duration and proportion of time spent in each transportation mode. In other words, the system conducts post-hoc transportation mode classification rather than real-time detection. Consequently, real-time detection parameters are not considered in this study.

### 3. Methodology

In this section, we will discuss the following issues: the mobile application architecture, the challenges encountered during data acquisition and the practical solutions we proposed and applied.

#### 3.1. *Android application architecture*

We have developed an Android application that records sensor data (accelerometer, gyroscope, magnetometer, and GPS) from the user’s smartphone and transfers them to a web server, where they are analyzed by an ML model to predict the user’s transportation modes. The so-called Mobidou application was developed in Java using an Integrated Development Environment (IDE), Android Studio (AS). The smartphone used for development and testing is a Samsung Galaxy A73 5G. The server runs a Debian environment with an Apache web server and a MariaDB relational database.

##### 3.1.1. *Automatic acquisition cycle*

The purpose of this section is to describe the automatic data acquisition cycle, from triggering of the recording to saving and sending the data, and detailing each stage of the process. At first connecting to the Mobidou server, on the application’s homepage, the user authorizes their data to be saved on the server. A random ID is automatically generated by the server, but no personal data (such as smartphone number) is stored on the server and all communications are made with this ID. After this operation, the cycle includes the following five phases:

- **Phase 1:** Users of the smartphone application can select the sensors they wish to use for data acquisition, as well as configure their respective sampling frequencies. A description of the input screens is provided below. Once the acquisition is completed, the recorded data must be transferred to the Mobidou server for analysis, resampling, and prediction as shown in the Fig. 1.
- **Phase 2:** When the data is transferred to the server, the raw sensor data is saved in a dedicated directory, while the metadata (sensor type, requested frequency, timestamps, etc.) is stored in a MariaDB relational database. A background program runs every minute to analyze the actual sampling frequencies recorded by the smartphone sensors (see Section 3.2.1 on the frequency capture anomaly). The acquisition duration is also calculated. If anomalies are detected, the program excludes the corresponding file from further processing. The validation or rejection status of each file is then recorded in the database.
- **Phase 3:** This phase involves resampling the data to ensure that they match the sampling frequency initially requested on the smartphone. For the Mobidou application, the default frequency is set to 20 Hz. The resampled data is saved in a new file, processing is performed only once, and this resampled file is used for all subsequent analyses.

- **Phase 4:** A process running every minute reads the newly resampled files to perform the prediction. The data is analyzed by an ML model (Sections 4.4 and 4.5 discuss the selected ML model and the features used for classification), which generates predictions based on fixed time windows. Each prediction result is then stored in the database, preventing redundant computations in future processing.
- **Phase 5:** On the homepage of the Mobidou smartphone application, a program automatically downloads and displays the predicted transportation modes, expressed in total time each one and generated by the classification algorithm embedded on the server.

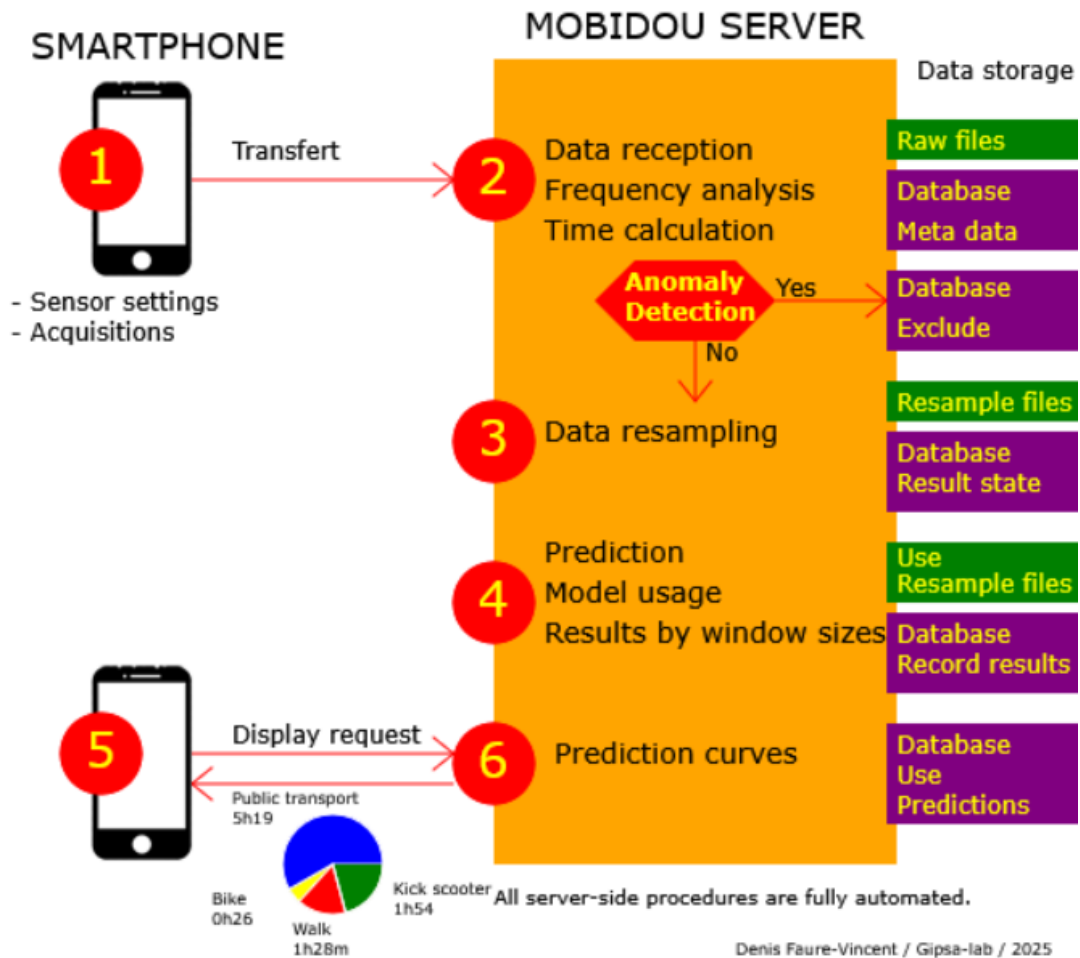


Figure 1.: Automatic acquisition cycle between smartphone application and server

### 3.1.2. Android application component

This section presents the different components of the Android application, detailing their role and interaction in the overall operation of the system. In the model-training phase, where the true transport mode is required, users manually select their current mode, as illustrated in Fig. 2a. This option will be hidden in the end-user version.

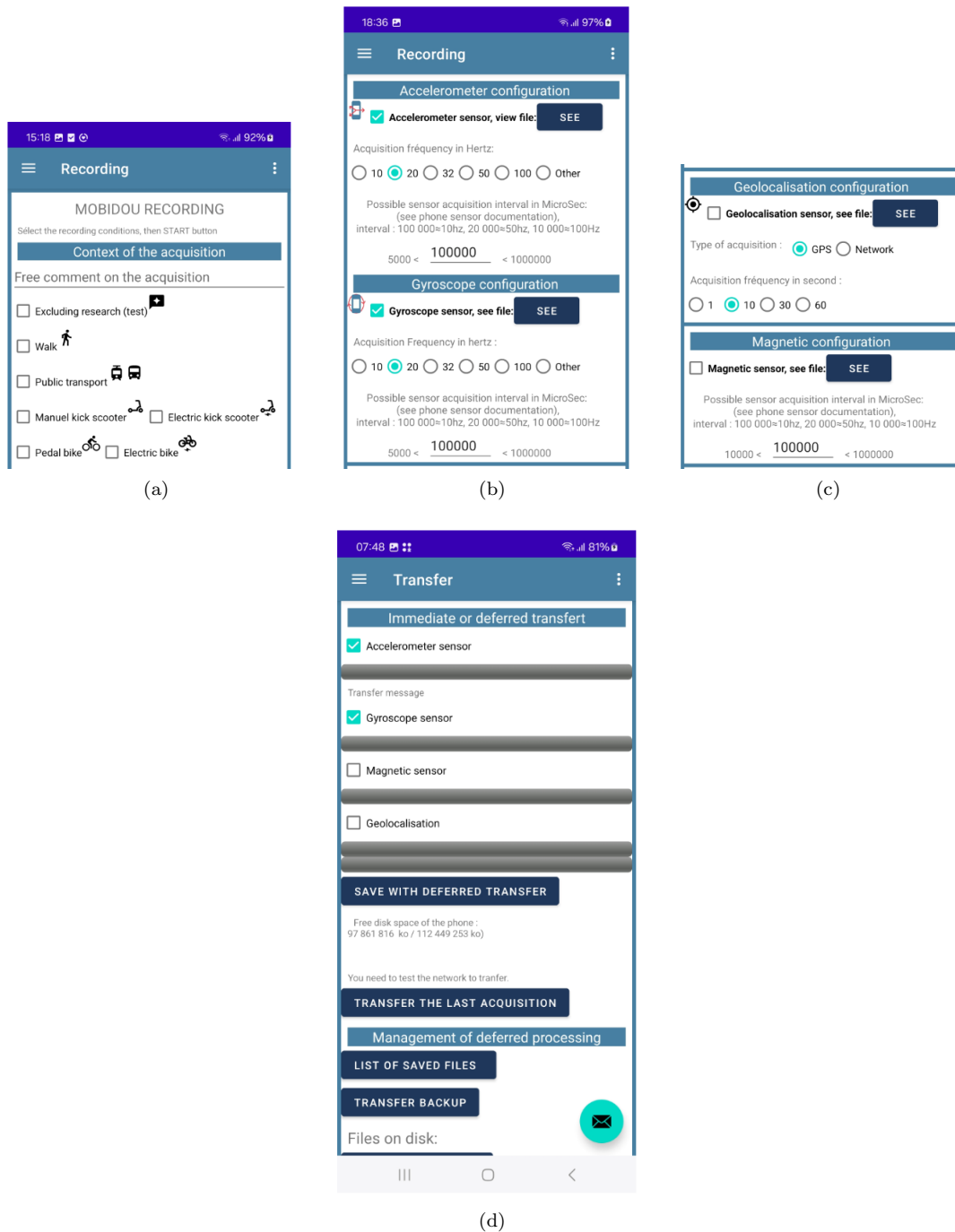


Figure 2.: Components of the Mobidou Android application: (a), (b) and (c) recording data, (d) transfer data to the server.

The user interface allows:

- Selecting the transportation mode (walk, bike, kick scooter or public transport). We also planned to consider electric mobility options, but for now they have not been used yet.
- Providing free-text comments on the acquisition conditions (weather, anomalies,

- specific context, etc.).
- Excluding certain recordings used solely for testing or verification purposes (option “Excluding research (test)”).
  - Starting the data recording once the conditions have been selected. For accelerometer settings (Fig. 2b), users can choose either the sampling frequency or the time interval between measurements (since sensors operate based on time intervals, a conversion is performed to calculate the frequency). Similarly, for gyroscope settings (Fig. 2b), both frequency and time interval options are available. For geolocation settings (Fig. 2c), the user can configure the geolocation method and sampling frequency. When the recording is finished, you can view the first lines of the recording file by activating the “see” button. Data transfers can be set to occur either immediately or with a delay (Fig. 2d). Automatic transfer will be considered in a future version of the application.

### 3.2. Challenges encountered during recording data

#### 3.2.1. Sampling frequencies

The actual sampling frequencies of the sensors (i.e., the time intervals between measurements) do not always match the requested sampling frequency. This discrepancy can occur because the sensor may be accessed by multiple applications simultaneously, causing variations in sampling rates. Additionally, it seems that some smartphones tend to revert to default sampling frequencies typically 50 or 100 Hz after a certain period of time, regardless of the requested settings.

- **Solution:** The raw data is transferred to the Mobidou server without any processing. They are then resampled in the server at the desired frequency.

#### 3.2.2. Parallel data recording

The main application program cannot simultaneously record multiple sensors without blocking the main application.

- **Solution:** An autonomous service was developed to operate independently from the main application. Within this service, multiple subprograms, implemented as threads, are launched in parallel to independently record data from each sensor. Inter-process communication is used to manage and terminate all threads simultaneously when needed.

#### 3.2.3. Sleep and doze modes

During the smartphone’s sleep mode, sensors can remain active and continue recording data through the use of a background service. However, probably to optimize energy consumption, smartphones may enter a deeper power-saving state known as doze mode. In this mode, the system applies stricter power rules on background services to preserve battery life. At regular but progressively longer intervals, the system briefly exits doze mode during maintenance windows to perform specific tasks before returning to deep sleep. During our tests, we observed data loss from the sensors when the device entered doze mode, even when using an autonomous background service.

- **Solution:** During the data acquisitions, we programmatically disabled sleep and doze modes to ensure that the application’s recordings were performed without

any interruption. This restriction helps ensure better data quality, which is essential for building a reliable and accurate model.

#### *3.2.4. Application interface*

Since the smartphone must remain in sleep mode, placing it in a pocket can cause friction that may unintentionally interrupt data recordings.

- **Solution:** A dedicated button is used to disable the stop recording function. This means that accidental friction must first disable the lock button before the recording can be stopped. This double-lock mechanism significantly reduces the risk of unintentional interruptions. Additionally, a scrolling interface ensures that the recording button is not immediately accessible, further minimizing the chance of accidental activation.

#### *3.2.5. Data transfers*

Depending on network bandwidth and the smartphone’s processing load, data transfers may be delayed, disrupted, or even interrupted, for example, if the server becomes temporarily inaccessible.

- **Solution:** Files are divided into multiple numbered sub-files, which are individually sent to the Mobidou server. A server-side process then reconstructs the original file from these parts. In case of a transfer interruption, the user has the option to either resume or cancel the transfer (see the analysis below regarding reliability rates).

#### *3.2.6. Implementation of monitoring tools*

How can we detect anomalies in data acquisition and transformation?

- **Solution:** To refine the data acquisition and transformation process, we implemented dashboards for daily monitoring. The first dashboard tracks the acquisition process, and the second monitors predictions. In addition, we established some statistical indicators for further analysis.

## **4. Dataset specifications and classification**

### *4.1. Collected data*

Data is collected by the Mobidou application using two sensors, the accelerometer and the gyroscope, in the city of Grenoble in France. To respect user privacy, GPS data was not recorded in this study. In addition, we focused on reducing sensor dependency and minimizing energy consumption, thereby improving the smartphone’s battery life. In the data collection, we have considered 13 smartphone models. The brands of smartphones used in the acquisition phases are : Samsung, Xiaomi, Google and Oppo. The main keys of data are:

- Data is collected from 20 participants.
- Sensors data is acceleration ( $m/s^2$ ) from accelerometer and angular rate ( $^\circ/s$ ) from gyroscope.
- An amount of 16.5 hours of data sampled at a frequency of 20 Hz was recorded.

- Experiments were conducted throughout the day and varied weather conditions.
- There were no restrictions on user behavior during data collection. Participants were able to move freely, ensuring that the dataset captures a wide range of real-world variations, making the dataset more representative of real-life conditions. Fig. 3 shows the duration in minutes of each transportation mode use.

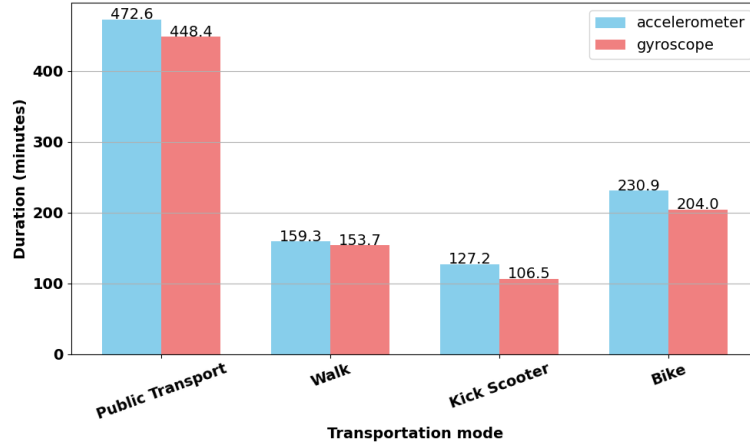


Figure 3.: Duration in minutes of each transportation mode use

The data concerning smartphone acquisitions are accessible according to the Findable, Accessible, Interoperable, Reusable (FAIR) principles (CCSD - CNRS, n.d.):

- **Findable:** Each file is assigned a unique and persistent identifier (id\_file). Meta-data is stored in a structured relational database.
- **Accessible:** The data is accessible on a web server at the University of Grenoble Alpes. Currently, access is restricted to specific computers with authorized IP addresses. External access can be granted upon request by researchers.
- **Interoperable:** The data is available for download in an open and free JavaScript Object Notation (JSON) format, ensuring compatibility with a wide range of tools and platforms.
- **Reusable:** The data can be reused for other research projects, in accordance with applicable licensing and ethical guidelines.

For now, the dataset has not yet been published online due to the ongoing finalization of ethical considerations. But the presented results in this paper are based on this dataset. However, we used our previous Captimove dataset (Taia-Alaoui et al., 2020) for initial testing and performance evaluation of the proposed algorithms.

#### 4.2. Reliability of data collection

The goal is to assess the reliability from the data collection on the smartphone to the prediction and display the final curve. Throughout the process, errors may be detected and lead either to corrections (resampling) or to the exclusion of the data (e.g., abnormal frequency deviation reported by the sensor or transfer issues). The pie chart in Fig. 4 shows that 94% of the files successfully passed through all phases. This result corresponds to a dataset recorded until 15 March 2025, and it may still evolve as new recordings continue to be added to the dataset.

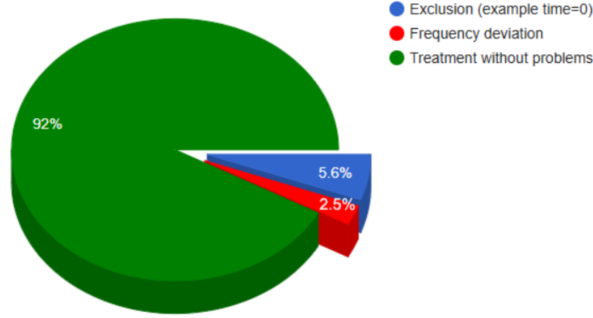


Figure 4.: Reliability of data collection

### 4.3. Data filtering

The quality of the raw data provided by inertial sensors has been improved, by identifying and removing potentially noisy portions. First, we removed a portion of the data at the beginning and end of the recording. This step is important because these segments often contain non-representative values, due, for example, to the start or stop of the recording, or to the manipulation of the smartphone. They are often sources of noise or artifacts. For each acquisition file, the first 10 s and last 10 s are removed. Second, we identified noisy windows by dividing the data into segments of fixed size and evaluating several statistical indicators. For each window, we calculated the magnitude of the triaxial signal as shown in Eq. (1) so we can capture the overall intensity of the movement, regardless of its direction.

$$M_p = \sqrt{\hat{d}_{(x,p)}^2 + \hat{d}_{(y,p)}^2 + \hat{d}_{(z,p)}^2}, \quad (1)$$

where  $\hat{d}_{(x,p)}$ ,  $\hat{d}_{(y,p)}$ , and  $\hat{d}_{(z,p)}$  are the data points from x, y, and z-axes of the sensor (accelerometer or gyroscope).

Then we apply several tests. A minimum magnitude that is too low, high variability, a concentrated or asymmetric distribution (kurtosis and skewness), and abrupt variations are all signs of noise. A minimum magnitude that is too low indicates that the smartphone is probably stationary or poorly positioned, with no significant activity. This kind of flat signal may correspond to rest phases or measurement errors. Also, significant signal fluctuations (measured by a high standard deviation) can reflect sudden jolts, often linked to involuntary manipulation of the device or measurement artifacts, rather than real and consistent activity. In addition, a low kurtosis indicates a flat or weakly structured distribution, suggesting that the signal may be affected by noise. A marked asymmetry in the distribution (high skewness) may indicate signal distortion, often caused by bias or a poorly calibrated sensor. Finally, sudden variations in magnitude (quantified by the derivative of the signal) can indicate sudden anomalies that do not correspond to natural movement. By combining these criteria, we can remove robustly noisy windows, and thus improve the overall quality of the dataset for ML models. Then, we applied the Z-Score normalization to standardize the sensor data, as shown in Eq. (2):

$$X_{\text{norm}} = \frac{X - \mu(X)}{\sigma(X)}, \quad (2)$$

where  $\mu(X)$  and  $\sigma(X)$  represent the mean and the standard deviation of each element of the sensor data, respectively.

With this pre-processing, we make the data more reliable before training the classification model.

#### *4.4. Segmentation and feature extraction*

In order to transform the raw signals into representations that can be used by the classification model, a feature extraction function was implemented. This splits the data into time windows of fixed sizes, defined by window size, and then calculates, for each segment, a set of statistical and frequency features based on the magnitude of the inertial signal. From the magnitude several characteristics are extracted, such as the mean, standard deviation, median, maximum, and minimum values, kurtosis, and skewness. Energy measures are also calculated, including the spectral energy obtained by a Fast Fourier Transform (FFT), and the dominant frequency (corresponding to the main peak in the spectrum, excluding zero frequency). In addition, we calculate the number of peaks detected on the z-axis signal, which may correspond to steps or jolts characteristic of certain transportation modes. Each window is therefore represented by a vector of features and associated with a label (transportation mode). All extracted vectors and their labels are then returned in the form of NumPy tables, ready to be used for training or evaluating a classification model. We have chosen a window length of 10 s, so it is not too small that can not capture all the characteristics of each mode of transportation and is not a larger window that may introduce noise (Gharbi et al., 2025).

#### *4.5. Selected classifiers for TMD*

Many classifiers have been used for TMD purposes (Gharbi et al., 2024; Saha et al., 2021; Wu et al., 2018). The most widely used are Random Forest (RF), Decision Tree (DT), Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Gradient Boosting (GB) and Naive Bayesian (NB).

In our work, three supervised learning models were selected to classify the four modes of transportation effectively based on sensor data, which are RF, GB, and SVM. These models represent different learning approaches (random trees, boosting, and margins). They were chosen as they have been proven in the literature to give the best results with TMD (Alam, Ahmed, Hossain, Emo, Bidhan, Reza, Alam, Hassan, Pupo and Fortino, 2023; Wang and Ross, 2018; Louk and Tama, 2022; Li et al., 2021).

RF model is an ensemble model based on decision trees, widely used for its robustness to noisy data, offering good results without the necessity of a complex tuning. GB model is another ensemble model, based on the principle of boosting, which involves combining several weak models (often shallow decision trees) to create a strong and high-performance model. SVM model is another supervised learning algorithm. Its goal is to determine the hyperplane with the maximum separation margin. Systematic comparisons of all relevant metrics enable us to identify the one that offers the best performance to detect all modes and, therefore, to use it to predict the user's transportation modes.

#### 4.6. Evaluation metrics

To evaluate the models considered, we used accuracy as the primary metric, precision, recall, and F1-score to calculate the detection performance rate of the predicted classes as previously done in (Alam, Ahmed, Hossain, Emo, Bidhan, Reza, Alam, Hassan, Pupo and Fortino, 2023; Sadeghian et al., 2022; Kumar et al., 2021).

$$\text{Accuracy} = \frac{\sum_{j=1}^k TP_j}{N}, \quad (3)$$

where  $k$  is the number of modes of transportation,  $N$  is the number of samples, and  $TP_j$  is the number of samples correctly assigned to the mode of transportation  $j$ .

$$\text{Precision} = \frac{TP}{TP + FP}, \quad (4)$$

$$\text{Recall} = \frac{TP}{TP + FN}, \quad (5)$$

where TP represents True Positives, FP represents False Positives, and FN represents False Negatives.

$$\text{F1-score} = \frac{2 \times \text{recall} \times \text{precision}}{\text{recall} + \text{precision}}. \quad (6)$$

In order to validate the trained models, we used the stratified k-fold cross-validation method (Witten et al., 2005). The training set is split into k groups for cross-validation. Each part has the same class distribution as the full dataset. Then, each part is used once for testing while the model is trained on the other k-1 parts.

### 5. Experimental results and discussion

In this section, we present the performance evaluation of the chosen models using the collected dataset. Tables 1, 2 and 3 resume the models key hyper-parameters, adjusted using the performance metrics described in Section 4.6.

Table 1.: RF Hyper-parameters

n_estimators	max_depth	min_samples_split	min_samples_leaf
200	20	5	1

Table 2.: GB Hyper-parameters

n_estimators	learning_rate	max_depth
100	0.1	10

Table 3.: SVM Hyper-parameters

C	kernel	gamma
10	rbf	scale

The three models selected for evaluation were trained using the collected data. To assess the classification accuracy, the considered features were employed. The confusion matrices of models are presented in Fig. 5. We can conclude that the GB model shows high performances to detect the four transportation modes. The RF model performed very similarly, but with slightly more confusion between modes. Conversely, the SVM model shows more confusion between classes, which degrades its overall performance. Fig. 6 and Table 4 show the performance comparison between the three models.

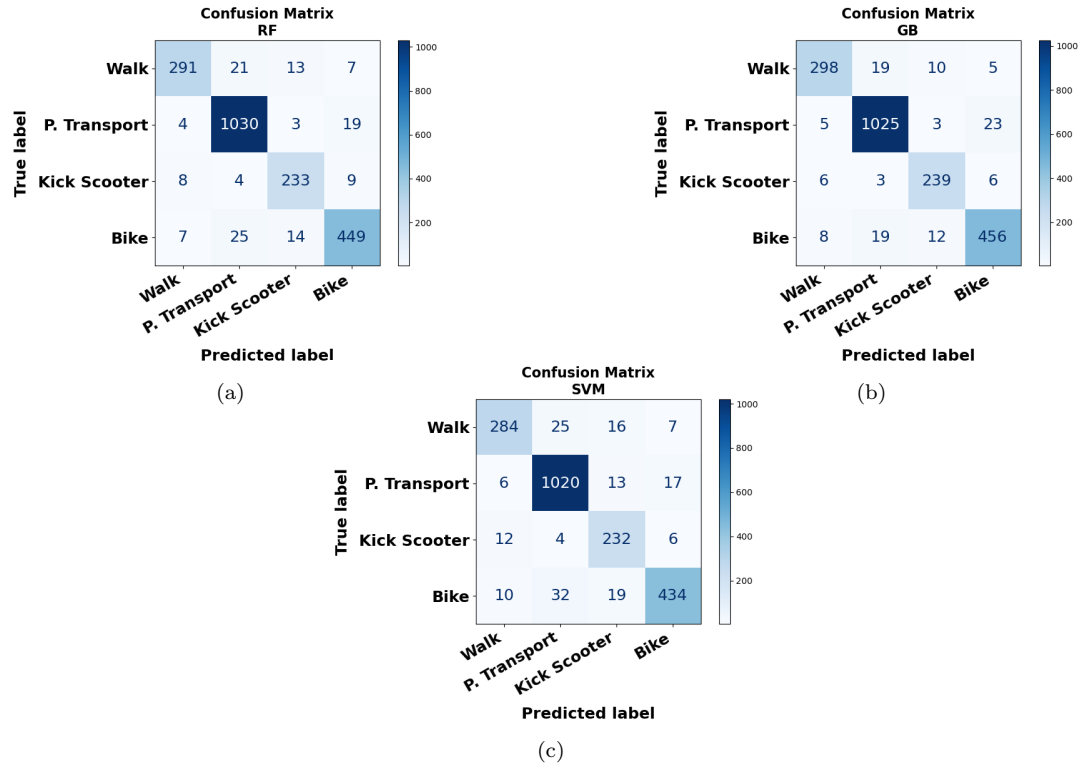


Figure 5.: Confusion matrices for (a) RF model, (b) GB model, and (c) SVM model.

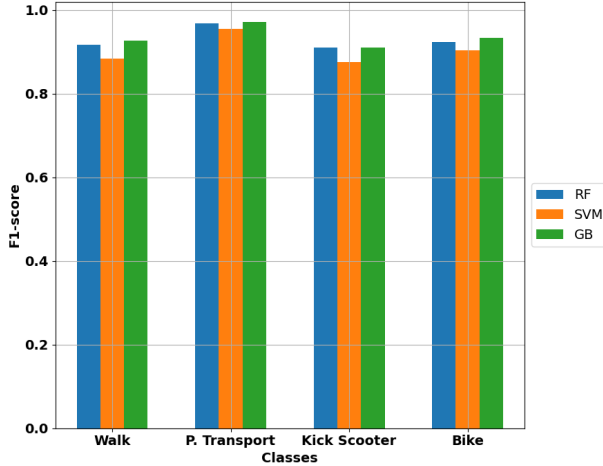


Figure 6.: F1-score for each class of the three models.

Table 4.: Performance comparison of the three models

	RF			GB			SVM		
	Precision (%)	Recall (%)	Accuracy (%)	Precision (%)	Recall (%)	Accuracy (%)	Precision (%)	Recall (%)	Accuracy (%)
C0	94	89	94	94	91	95	91	86	92
C1	96	98		96	98		94	97	
C2	91	91		92	92		86	92	
C3	93	92		94	93		93	87	

In Table 4, C0 is walk, C1 is public transport, C2 is kick scooter, and C3 is bike. The results show that the GB model achieved the highest overall accuracy of 95% and F1-score across all classes, particularly for public transport (0.97) and walk (0.92). The RF model performed similarly, with an accuracy of 94% and slightly lower F1-score. The SVM model reached an overall accuracy of 92%, showing a good performance on public transport (0.95) but lower performance on the kick scooter (0.89). These results demonstrate that tree-based ensemble methods are effective and reliable in managing the variability of the real-world sensor data. Among the three models, the GB model consistently offers the best overall performance, achieving a strong balance between precision and recall across all classes but nothing prevents using the RF model.

To further analyzed the performance of the classifiers, notably with class imbalance, we analyze the Precision-Recall (PR) curves for each model in Fig. 7. These curves provide a more detailed view of the trade-off between precision and recall, and are particularly relevant for imbalanced datasets where accuracy alone may be misleading. To analyze and evaluate the classification results of the different models, we plotted the PR curves for the RF, SVM, and GB classifiers. As shown in Fig. 7, all models achieved high Average Precision (AP), with the GB model slightly outperforming the others (AP = 0.99), followed by the RF model (AP = 0.98) and the SVM model (AP = 0.97). These results indicate that all models are capable of achieving both high recall and precision. The GB model offers the best trade-off and is therefore selected as the most effective classifier for the automatic prediction process and the Android application. For the purpose of validating the process and displaying the prediction within the application, we selected test files provided by a user with data collected over a period of four months from December 2024 to March 2025 and computed the predictions shown in the Fig. 8.

The duration displayed the Fig. 8 corresponds to the cumulative time of the pre-

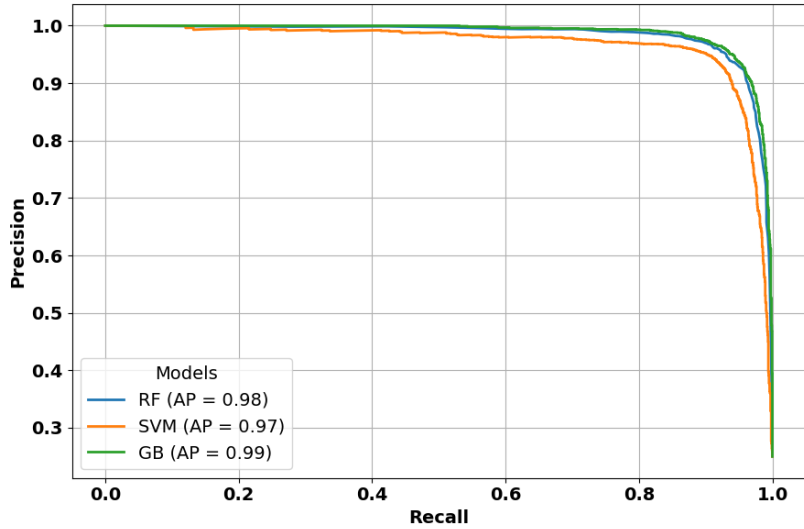


Figure 7.: Precision-recall curves for the three models

dictions as a function of window size and frequency as demonstrated in Fig. 9. The model generated predictions based on a size window, with each window consisting of 200 samples at a frequency of 20 Hz, corresponding to 10 s intervals. As new predictions were made for each window, the cumulative time was updated incrementally for the predicted class. The cumulative time per class, shown in the last column, reflects the total duration associated with each transportation mode as predictions accumulate over time.

The results show a good prediction performance. Since the ground-truth durations are known, we compared them with the predicted durations and obtained low error rates: 1% for the walking class, 4% for the biking class, 1% for public transport, and 2% for the kick scooter. These results validate the effectiveness of the proposed method.



Figure 8.: Prediction displayed on the Mobidou application

File Number	Phone Number	Prediction	Samples Number	Sampling Frequency	Windows size	Cumulative time per class
2388	63	Class=1	Samples number=200	20Hz	10 seconds	Cumul [1] =10 cumul
2388	63	Class=1	Samples number =200	20Hz	10 seconds	Cumul [1] =20 cumul
2388	63	Class=1	Samples number =200	20Hz	10 seconds	Cumul [1] =30 cumul
2388	63	Class=1	Samples number =200	20Hz	10 seconds	Cumul [1] =40 cumul
2388	63	Class=1	Samples number =200	20Hz	10 seconds	Cumul [1] =50 cumul
2388	63	Class=1	Samples number =200	20Hz	10 seconds	Cumul [1] =60 cumul
2396	63	Class=1	Samples number =200	20Hz	10 seconds	Cumul [1] =70 cumul
2396	63	Class=1	Samples number =200	20Hz	10 seconds	Cumul [1] =80 cumul
2396	63	Class=1	Samples number =200	20Hz	10 seconds	Cumul [1] =90 cumul
2398	63	Class=3	Samples number =200	20Hz	10 seconds	Cumul [1] =10 cumul
2398	63	Class=3	Samples number =200	20Hz	10 seconds	Cumul [1] =20 cumul
2398	63	Class=3	Samples number =200	20Hz	10 seconds	Cumul [1] =30 cumul
2398	63	Class=3	Samples number =200	20Hz	10 seconds	Cumul [1] =40 cumul
2398	63	Class=3	Samples number =200	20Hz	10 seconds	Cumul [1] =50 cumul
2398	63	Class=3	Samples number =200	20Hz	10 seconds	Cumul [1] =60 cumul

Figure 9.: Screenshot of the calculation method for the durations shown in the Fig. 8

## 6. Conclusion

This work focuses on the development of an automated transportation mode prediction Android application based on inertial data collected from smartphone sensors (accelerometer and gyroscope). Beyond the modeling task, particular attention was given to two key aspects: first, minimizing the number of sensors to reduce hardware dependency and energy consumption; and second, addressing the challenges associated with real-world data acquisition. These challenges include sensor noise, variability in sampling frequency, issues with parallel data recording, interruptions caused by sleep or doze modes, accidental friction that may halt data collection, delays in data transfer, and more. We proposed a series of practical solutions to mitigate these issues. Our study specifically targets four green transportation modes characterized by low carbon emissions and minimal environmental impact. Three supervised learning models RF, GB, and SVM were trained and compared using features computed from sliding windows of raw sensor data. Among these, the GB model emerged as the best-performing one, with an accuracy of 95%. The selected model was integrated into the Mobidou server in order to predict the transportation modes and display their durations on the Android application. The results show good prediction performance with an error rate below than 4%. This deployment demonstrates the feasibility of the application for TMD. In addition, the application contains other sensors not used in this study, so it can be used by other researchers for different purposes.

## Acknowledgment(s)

This work was supported by the French National Research Agency, within the framework of the “Investissements d’avenir” program (ANR-10-AIRT-05 and ANR-15-IDEX-02) (CAPTIMOVE project), and under the France 2030 programme, reference MIAI Cluster ANR-23-IACL-0006. The sponsors had no involvement in the study design, the collection, analysis, and interpretation of data, or in writing the manuscript. This work also forms part of the broader translational and interdisciplinary GaitAlps research program.

## References

- Alam, M. G. R., Haque, M., Hassan, M. R., Huda, S., Hassan, M. M., Strickland, F. L. and AlQahtani, S. A. (2023), ‘Feature cloning and feature fusion based transportation mode detection using convolutional neural network’, *IEEE Transactions on Intelligent Transportation Systems* **24**(4), 4671–4681.
- Alam, M. M., Ahmed, T., Hossain, M., Emo, M. H., Bidhan, M. K. I., Reza, M. T., Alam, M. G. R., Hassan, M. M., Pupo, F. and Fortino, G. (2023), ‘Federated ensemble-learning for transport mode detection in vehicular edge network’, *Future Generation Computer Systems* **149**, 89–104.
- Alaoui, F. T., Fourati, H., Kibangou, A., Robu, B. and Vuillerme, N. (2021), ‘Urban transportation mode detection from inertial and barometric data in pedestrian mobility’, *IEEE Sensors Journal* **22**(6), 4772–4780.
- Alaoui, F. T., Fourati, H., Kibangou, A., Robu, B. and Vuillerme, N. (2022), ‘Kick-scooters identification in the context of transportation mode detection using inertial sensors: Methods and accuracy’, *Journal of Intelligent Transportation Systems* pp. 1–21.
- Banani Ardecani, F., Mahmoudzadeh, A. and Mesbah, M. (2025), ‘Fuzing multiple erroneous sensors to estimate travel time’, *Journal of Intelligent Transportation Systems* **29**(5), 491–504.
- Bike Chattanooga Trip Data* (2025), <https://internal.chattadata.org/Recreation/Bike-Chattanooga-TripData/tdrg-39c4>. Accessed: 2025-04-22.
- Bjerre-Nielsen, A., Minor, K., Sapieżyński, P., Lehmann, S. and Lassen, D. D. (2020), ‘Inferring transportation mode from smartphone sensors: Evaluating the potential of wi-fi and bluetooth’, *PloS one* **15**(7), e0234003.
- Carpinetti, C., Lomonaco, V., Bedogni, L., Di Felice, M. and Bononi, L. (2018), Custom dual transportation mode detection by smartphone devices exploiting sensor diversity, in ‘2018 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)’, IEEE, pp. 367–372.
- CCSD - CNRS (n.d.), ‘Les principes fair’, <https://www.ccsd.cnrs.fr/principes-fair/>. Accessed: 2025-05-07.
- Choi, M. (2013), ‘Power and performance analysis of smart devices’, *International Journal of Smart Home* **7**(3), 57–67.
- Coroamă, V. C., Türk, C. and Mattern, F. (2019), Exploring the usefulness of bluetooth and wifi proximity for transportation mode recognition, in ‘Adjunct Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2019 ACM International Symposium on Wearable Computers’, pp. 37–40.
- Deilami, K., Rudner, J., Butt, A., MacLeod, T., Williams, G., Romeijn, H. and Amati, M. (2020), ‘Allowing users to benefit from tree shading: Using a smartphone app to allow adaptive route planning during extreme heat’, *Forests* **11**(9), 998.
- Demagh, L., Garda, P., Gilbert, C. and Hachicha, K. (2023), Tinyimm: Multimodal-multitask machine learning on low-power mcus for smart glasses, in ‘2023 IEEE SENSORS’, IEEE,

- pp. 1–4.
- Drosouli, I., Voulodimos, A., Miaoulis, G., Mastorocostas, P. and Ghazanfarpour, D. (2021), ‘Transportation mode detection using an optimized long short-term memory model on multimodal sensor data’, *Entropy* **23**(11), 1457.
- Dutzik, T., Madsen, T. and Baxandall, P. (2013), ‘A new way to go: The transportation apps and vehicle-sharing tools that are giving more americans the freedom to drive less’.
- Erdelić, M., Carić, T., Erdelić, T. and Šemanjski, I. (2025), ‘Enhancing transport mode classification benchmark by integrating spatial independence with multimodal dataset’, *Travel Behaviour and Society* **38**, 100929.
- Fourez, T., Verstaevel, N., Migeon, F., Schettini, F. and Amblard, F. (2025), ‘Transport mode detection on gnss and accelerometer data: a temporality based workflow’, *Transportation Research Procedia* **82**, 3454–3471.
- Gao, J., Ozbay, K. and Hu, Y. (2025), ‘Real-time anomaly detection of short-term traffic disruptions in urban areas through adaptive isolation forest’, *Journal of Intelligent Transportation Systems* **29**(3), 269–286.
- Ghahramani, N., Brakewood, C. and Peters, J. (2017), An exploratory analysis of intercity travel patterns using backend data from a transit smartphone application, in ‘96th Annual Meeting of the Transportation Research Board, Washington, DC’.
- Gharbi, I., Fourati, H. and Vuillerme, N. (2025), ‘Towards optimization of inertial sensors-based green transportation mode detection: Feature choice analysis, down-sampling effect, and sensor data integration’, *Transportation Research Interdisciplinary Perspectives*. Submitted for publication.
- Gharbi, I., Taia-Alaoui, F., Fourati, H., Vuillerme, N. and Zhou, Z. (2024), ‘Transportation mode detection using learning methods and self-contained sensors’, *Sensors* **24**(22), 7369.
- Gjoreski, H., Ciliberto, M., Wang, L., Morales, F. J. O., Mekki, S., Valentin, S. and Roggen, D. (2018), ‘The university of sussex-huawei locomotion and transportation dataset for multimodal analytics with mobile devices’, *IEEE Access* **6**, 42592–42604.
- Golcarenarenji, G., Martinez-Alpiste, I., Wang, Q. and Alcaraz-Calero, J. M. (2024), ‘Robust real-time traffic light detector on small-form platform for autonomous vehicles’, *Journal of Intelligent Transportation Systems* **28**(5), 668–678.
- Hasan, R. A., Irshaid, H., Alhomaidat, F., Lee, S. and Oh, J.-S. (2022), ‘Transportation mode detection by using smartphones and smartwatches with machine learning’, *KSCCE Journal of Civil Engineering* **26**(8), 3578–3589.
- Hedemalm, E., Kor, A.-L., Hallberg, J., Andersson, K., Pattinson, C. and Chinnici, M. (2021), ‘Application of online transportation mode recognition in games’, *Applied Sciences* **11**(19), 8901.
- Huang, Y., Gao, L., Ni, A. and Liu, X. (2021), ‘Analysis of travel mode choice and trip chain pattern relationships based on multi-day gps data: A case study in shanghai, china’, *Journal of transport geography* **93**, 103070.
- Kumar, S., Damaraju, A., Kumar, A., Kumari, S. and Chen, C.-M. (2021), ‘Lstm network for transportation mode detection’, *Journal of Internet Technology* **22**(4), 891–902.
- Li, J., Pei, X., Wang, X., Yao, D., Zhang, Y. and Yue, Y. (2021), ‘Transportation mode identification with gps trajectory data and gis information’, *Tsinghua Science and Technology* **26**(4), 403–416.
- Lorintiu, O. and Vassilev, A. (2016), Transportation mode recognition based on smartphone embedded sensors for carbon footprint estimation, in ‘2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)’, IEEE, pp. 1976–1981.
- Louk, M. H. L. and Tama, B. A. (2022), ‘Revisiting gradient boosting-based approaches for learning imbalanced data: A case of anomaly detection on power grids’, *Big Data and Cognitive Computing* **6**(2), 41.
- Lo’ai, A. T., Basalamah, A., Mehmood, R. and Tawalbeh, H. (2016), ‘Greener and smarter phones for future cities: Characterizing the impact of gps signal strength on power consumption’, *IEEE access* **4**, 858–868.
- Lu, L., He, Z., Wang, J., Chen, J. and Wang, W. (2021), ‘Estimation of lane-level travel time

- distributions under a connected environment’, *Journal of Intelligent Transportation Systems* **25**(5), 501–512.
- Lu, W., Fu, X., Liu, J., Hainen, A., Xia, F. and Chen, O. (2025), ‘A “digital twin” traffic simulation tool for network-wide real-time traffic monitoring and management’, *Journal of Intelligent Transportation Systems* pp. 1–13.
- Mahmoud, N., Abdel-Aty, M., Cai, Q. and Yuan, J. (2022), ‘Estimating cycle-level real-time traffic movements at signalized intersections’, *Journal of intelligent transportation systems* **26**(4), 400–419.
- Merikhipour, M., Khanmohammadidoustani, S. and Abbasi, M. (2025), ‘Transportation mode detection through spatial attention-based transductive long short-term memory and off-policy feature selection’, *Expert Systems with Applications* **267**, 126196.
- Molina-Campoverde, J. J., Rivera-Campoverde, N., Molina Campoverde, P. A. and Bermeo Naula, A. K. (2024), ‘Urban mobility pattern detection: Development of a classification algorithm based on machine learning and gps’, *Sensors* **24**(12), 3884.
- Petraki, V., Ziakopoulos, A. and Yannis, G. (2020), ‘Combined impact of road and traffic characteristic on driver behavior using smartphone sensor data’, *Accident Analysis & Prevention* **144**, 105657.
- Richoz, S., Ciliberto, M., Wang, L., Birch, P., Gjoreski, H., Perez-Urbe, A. and Roggen, D. (2019), Human and machine recognition of transportation modes from body-worn camera images, in ‘2019 Joint 8th International Conference on Informatics, Electronics & Vision (ICIEV) and 2019 3rd International Conference on Imaging, Vision & Pattern Recognition (icIVPR)’, IEEE, pp. 67–72.
- Richoz, S., Wang, L., Birch, P. and Roggen, D. (2020), ‘Transportation mode recognition fusing wearable motion, sound, and vision sensors’, *IEEE Sensors Journal* **20**(16), 9314–9328.
- Sadeghian, P., Håkansson, J. and Zhao, X. (2021), ‘Review and evaluation of methods in transport mode detection based on gps tracking data’, *Journal of Traffic and Transportation Engineering (English Edition)* **8**(4), 467–482.
- Sadeghian, P., Zhao, X., Golshan, A. and Håkansson, J. (2022), ‘A stepwise methodology for transport mode detection in gps tracking data’, *Travel Behaviour and Society* **26**, 159–167.
- Saha, P., Alam, M. M., Tapotee, M. I., Baray, S. B. and Ahad, M. A. R. (2021), An empirical approach for human locomotion and transportation recognition from radio data, in ‘Adjunct Proceedings of the 2021 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2021 ACM International Symposium on Wearable Computers’, pp. 390–395.
- Sharma, A., Singh, S. K., Udmale, S. S., Singh, A. K. and Singh, R. (2020), ‘Early transportation mode detection using smartphone sensing data’, *IEEE Sensors Journal* **21**(14), 15651–15659.
- SHL Dataset - Sussex-Huawei Locomotion Dataset* (n.d.), <http://www.shl-dataset.org/>. Accessed: April 24, 2025.
- Soltani, S., Valizadeh, J., Aghdamigargari, M., Mirzaei Varzeghani, N. and Mozafari, P. (2025), ‘A cooperative game model for emergency transportation planning using the internet of things’, *Journal of Intelligent Transportation Systems* pp. 1–20.
- Sutanto, E., Sapuan, I., Yazid, M. and Fahmi (2020), Android based position tracking for car condition monitoring, in ‘AIP Conference Proceedings’, Vol. 2314, AIP Publishing LLC, p. 040003.
- Taia-Alaoui, F., Fourati, H., Vuillerme, N., Kibangou, A., Robu, B. and Villemazet, C. (2020), ‘Captimove dataset’. Captimove-TMD, <https://perscido.univ-grenoble-alpes.fr/datasets/DS310>.
- Tang, Q. and Cheng, H. (2023), ‘Feature pyramid bilstm: Using smartphone sensors for transportation mode detection’, *arXiv preprint arXiv:2310.11087*.
- Wang, C., Zhao, F., Luo, H., Sun, P. Z. and Fang, Y. (2025), ‘Cross-transportation-mode knowledge transfer for trajectory recovery with meta learning’, *IEEE Transactions on Intelligent Transportation Systems*.
- Wang, F. and Ross, C. L. (2018), ‘Machine learning travel mode choices: Comparing the

- performance of an extreme gradient boosting model with a multinomial logit model', *Transportation Research Record* **2672**(47), 35–45.
- Wang, H., Luo, H., Zhao, F., Qin, Y., Zhao, Z. and Chen, Y. (2018), Detecting transportation modes with low-power-consumption sensors using recurrent neural network, *in* '2018 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCCom/IOP/SCI)', IEEE, pp. 1098–1105.
- Wang, L., Gjoreski, H., Ciliberto, M., Mekki, S., Valentin, S. and Roggen, D. (2018), Benchmarking the sh1 recognition challenge with classical and deep-learning pipelines, *in* 'Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers', pp. 1626–1635.
- Wang, L. and Roggen, D. (2019), Sound-based transportation mode recognition with smartphones, *in* 'ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)', IEEE, pp. 930–934.
- Wang, W., Zhao, X., Gong, Z., Chen, Z., Zhang, N. and Wei, W. (2020), 'An attention-based deep learning framework for trip destination prediction of sharing bike', *IEEE Transactions on Intelligent Transportation Systems* **22**(7), 4601–4610.
- Witten, I. H., Frank, E., Hall, M. A., Pal, C. J. and Data, M. (2005), Practical machine learning tools and techniques, *in* 'Data mining', Vol. 2, Elsevier Amsterdam, The Netherlands, pp. 403–413.
- Wu, J., Akbari, A., Grimsley, R. and Jafari, R. (2018), A decision level fusion and signal analysis technique for activity segmentation and recognition on smart phones, *in* 'Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers', pp. 1571–1578.
- Yang, N., Lu, Q.-L., Yamnenko, I. and Antoniou, C. (2024), 'Efficient cloud-sourced transport mode detection using trajectory data: A semi-supervised asynchronous federated learning approach', *IEEE Internet of Things Journal*.
- Zheng, Y., Xie, X. and Ma, W.-Y. (2010), 'Geolife: A collaborative social networking service among user, location and trajectory.', *IEEE Data Eng. Bull.* **33**(2), 32–39.
- Zhou, Y., Zhang, Y., Yuan, Q., Yang, C., Guo, T. and Wang, Y. (2022), 'The smartphone-based person travel survey system: Data collection, trip extraction, and travel mode detection', *IEEE Transactions on Intelligent Transportation Systems* **23**(12), 23399–23407.
- Zhuang, Y. and Mao, X. (2025), 'Algorithm for setting the highest priority of proximity interaction commands for intelligent urban traffic control systems', *Journal of Intelligent Transportation Systems* pp. 1–11.