“Is Not the Truth the Truth?”
Analyzing the Impact of User Validations for Bus In/Out Detection in Smartphone-Based Surveys

Servizi, Valentino; Persson, Dan Roland; Pereira, Francisco Camara; Villadsen, Hannah; Bækgaard, Per; Peled, Inon; Nielsen, Otto Anker

Published in:
IEEE Transactions on Intelligent Transportation Systems

Link to article, DOI:
10.1109/TITS.2023.3291493

Publication date:
2023

Document Version
Peer reviewed version

"Is Not the Truth the Truth?": Analyzing the Impact of User Validations for Bus In/Out Detection in Smartphone-Based Surveys

Valentino Servizi, Dan Roland Persson, Francisco Camara Pereira, Hannah Villadsen, Per Bækgaard, Inon Peled, and Otto Anker Nielsen

***Abstract*— Knowledge of passenger flow underpins any optimal public transport application, such as new facilities design and operations. The interactions between passengers and sensors may provide the foundation to measure this flow and dismiss users and staff from the measures' validation loop. Removing humans and their errors may impact significantly and improve measures' cost and quality. The literature considered smartphones the leading enabler due to market penetration and embodied sensors. Smartphones allow users' localization, identification, authentication, and billing. Via Bluetooth, smartphones detect short-range implicit interactions, device-to-device. We model passenger states on buses, either be-in or be-out (BIBO). The BIBO use case identifies a fundamental building block of continuously-valued passenger flow, which this paper describes through a Human-Computer interaction experimental setting involving two autonomous buses and a proprietary smartphone-Bluetooth sensing platform. The resulting dataset of 14,000 observations/sensors contains two ground-truth levels: the first is the passengers' validation; the second is validation by video cameras surveilling buses and tracks. This study verifies separately classification based on Bluetooth and GPS signals, as well as an inertial navigation system, evaluating signals and related machine learning (ML) classifiers against measurement and ground-truth noise. The paper contributes a Monte Carlo simulation of labels-flip to emulate human errors in the labeling process, as in smartphone surveys, and a novel unsupervised variational auto-encoder classifier. Experimental results indicate error-free human validation is unlikely. The impact of mistakes on model performance bias can be significant. This use case supports the potential substitution of human validation with independent Bluetooth validation.**

***Index Terms*— Ground-truth, D2D interactions, autonomous vehicles, bluetooth low energy, Internet of Things.**

Manuscript received 23 March 2022; revised 17 September 2022 and 27 April 2023; accepted 8 June 2023. This work was supported by the European Regional Development Fund through the Urban Innovative Actions Initiative. The Associate Editor for this article was K. C. Leung. *(Corresponding author: Valentino Servizi.)*

This work involved human subjects or animals in its research. The author(s) confirm(s) that all human/animal subject research procedures and protocols are exempt from review board approval.

Valentino Servizi, Francisco Camara Pereira, Inon Peled, and Otto Anker Nielsen are with the Department of Technology, Management and Economics, Technical University of Denmark, 2800 Lyngby, Denmark (e-mail: byvalentino@icloud.com).

Dan Roland Persson and Per Bækgaard are with the Department of Applied Mathematics and Computer Science, Technical University of Denmark, 2800 Lyngby, Denmark.

Hannah Villadsen is with the Department of People and Technology, Roskilde University, 4000 Roskilde, Denmark.

Digital Object Identifier 10.1109/TITS.2023.3291493

I. Introduction

PASSENGER flow is a fundamental component for capacity estimation of public transport and for designing adequate infrastructure and services [1]. On bus transport, this flow measures passengers' variations in time and space, on the vehicles [2]. Multiple approaches promise real-time passenger flow estimation, but even the most advanced ones struggle with imprecision due to counting passengers indirectly, such as when paying by cash, or traveling without ticket [2]. Although autonomous buses and the internet of things (IoT) offer the opportunity of exploiting D2D (device to device) interactions for passenger flow beyond ticketing [3], available solutions such as check-in/check-out (CICO), walk-in/walk-out (WIWO), or be-in/be-out (BIBO) [4], [5] all seem prone to errors. For example in the CICO case, using radio-frequency identification (RFID) technology for the interactions between smart-cards and readers, people often forget either the CI or CO action. In the WIWO case, multiple users can enter the same gate at the same time and confuse the counter. To contribute improving passengers' count accuracy and user experience in public transportation, we focus on enabling next generation BIBO for ticket-less trips. This application could allow passengers, for example, to pay with contact-less, radio-based identification, and communication via smartphone-Bluetooth without human intervention and without explicit interaction [6]. This approach has the added advantage of being the most user-friendly for the growing population of smartphone-users, which is above 40% worldwide and up to 80% in western countries [7], since it depends only on the user carrying his/her device as he/she would normally do.

Although the global positioning system (GPS) is one of the most reliable and adopted technologies for outdoor tracking [8], GPS shows important limitations in urban areas [9]. The specific radio-signal frequency requires line of sight between sender and receiver, thus being affected by reflections from tall buildings and clouds. Consequently, GPS measurements require independent validation. Similarly, Bluetooth is one of the principal technologies for proximity detection [10] applied to indoor tracking, and the specific radio-signal frequency brings other limitations. For example, a smartphone-based travel survey on the Silver Line bus rapid transit in Boston, Massachusetts, deployed BIBO technology [11],

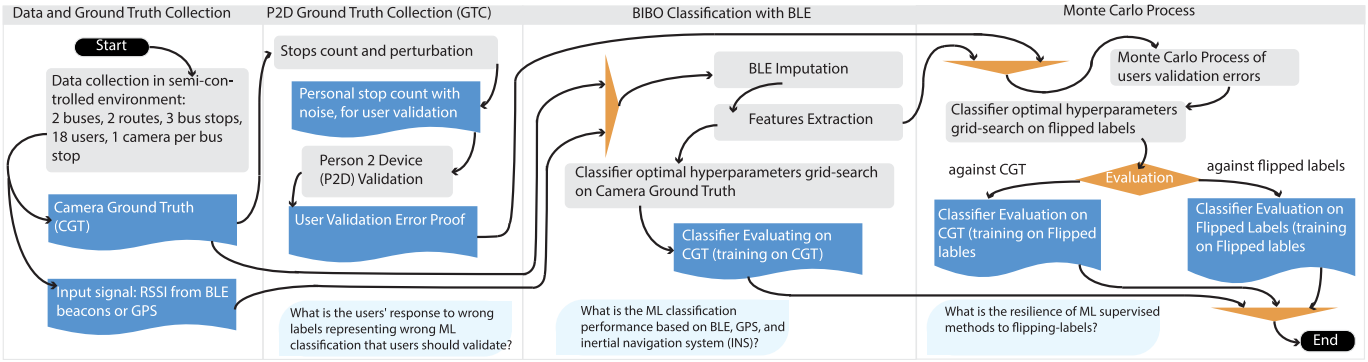


Fig. 1. Experiment workflow.

as a context detection system for a service quality survey, and so avoided collecting ground-truth (GT), in form of labels, from surveyed passengers; D2D implicit interaction between smartphones and Bluetooth devices installed on buses verified passengers' presence aboard, independently from GPS sensors. In the same study, the authors expose cases where BIBO verification was successful via GPS but not Bluetooth, potentially caused by human bodies impedance of signals within the bus. While a large body of literature presents a successful case for Bluetooth as indoor positioning technology, no previous work that we are aware of analyses in detail its use as independent measurement for labels and the impact of labeling errors on the BIBO classifier performance. In this use case, Bluetooth reception errors might present themselves as flipping- and outlying-labels [12], negatively impacting ML training and magnifying misclassifications.

Flipping-labels are known as items that human or machine classifiers labeled with a wrong class, despite the true one existing in the dataset; outlying-labels are items that belong to none of the classes in the dataset, but were mistakenly labeled as one of these classes [13]. The impact of these two problems on ML classifiers is extensively studied for independent and identically distributed (IID) datasets [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], such as for images, but not for time-series, such as Bluetooth or space-time GPS trajectories.

To train and deploy any ML classifier based on a set of signals, the literature showed that measurements' validation requires ground-truth collected explicitly from humans. To bridge the gap between the limitations mentioned in the preceding two paragraphs, we conducted a case study on a BIBO, smartphone and Bluetooth Low Energy (BLE) based system, with the following research question:

> *Can BLE-smartphone implicit interaction consistently substitute person-to-device (P2D) with device-to-device (D2D) ground-truth collection for the BIBO use case?*

To answer, we inquire the following aspects:

(i) *During the ground-truth collection, what is the users' response to wrong labels representing wrong ML classification that users should validate?*

(ii) *What is the ML classification performance based on BLE, GPS, and inertial navigation system (INS)?*

(iii) *What is the resilience of ML supervised methods to flipping-labels?*

Fig. 1 shows the process we executed through the following steps. First, we designed and implemented a smartphone-BLE platform. Second, we set up and ran an experiment involving a simple transport network composed of two autonomous buses operating on two routes and three bus stops, with a BLE device on each bus and bus stop; the experiment ran for two days. Third, we involved users, and we video-recorded each of their trips through this network; simultaneously, users' smartphone native (Android and iOS) application programming interface (API) read BLE devices' signal strength and classified the transport mode from the time-series of the INS. Our proprietary application stored these trajectories in a database. Fourth, we labeled the trajectories using video recordings as ground-truth with BIBO binary labels (be-in or be-out). Fifth, we created a Monte Carlo (MC) process to simulate labeling errors, i.e., flipping-labels, with various noise levels on recorded trip time-series. Finally, to understand the error tolerance, we evaluated and compared multiple classifiers, both on true and noisy labels. We create a unique dataset where the experimental setting incorporates multiple real-world conditions typical of urban high-density contexts: overlapping BLE fields, multiple makes and types of smartphones, native applications for the two main operating systems (OS), bus switching routes, bus moving at low speed only ($< 15\ km/h$), subjective preferences on how and where users carry their smartphones, or where they stand, both while traveling on bus and waiting at the bus stop. In such a BIBO system setup, we yield results (see Sec. IV) suggesting that BLE signal alone is robust to labeling errors and performs significantly better than commercially-available classifiers based on INS. Consequently, the use of BLE in combination with traditional signals, such as GPS, has the potential to dismiss humans from the validation loop.

A. Contribution

Despite the wide body of previous work (see Sec. II), in particular on the impact of labeling errors, this use case

did not receive enough attention, and thus no conclusions can be drawn as of its potential.

Whereas datasets are exclusively IID in existing work, we investigate dependent observations over time. In contrast to the literature studying the flipping labels' problem using true ground-truth from a synthetic generation of datasets, we provide a broader analysis over the impact of noisy labels on Machine Learning training and evaluation steps, and mostly we can carry out this analysis on real trajectories.

(i) Labels, BLE and GPS signals are obtained from a realistic setup involving real vehicles, devices, and people, producing real time-series.
(ii) Video-cameras collect high-quality ground-truth identifying be-in/be-out (BIBO) labels, i.e., presence inside (BI) or outside (BO) the bus.
(iii) Top-of-the-shelf smartphone OS provide INS-based classifier BIBO labels, which constitute the baseline for our analysis on both BLE- and GPS-based classifiers.
(iv) We collect experimental evidence about the types of errors in person-to-device labels collection: P2D is typical of real-world smartphone-based travel surveys.
(v) We introduce a methodology to augment high-quality labels by propagating errors, compatibly with experimental evidence (see Sec. III).
(vi) We identify ideal performance of ML supervised models evaluated and trained on ideal labels. We compare the ideal performance with both the perceived and the actual performance of the same models. In this case the perceived performance is the one we measure "assuming" existence of labels' errors in both model evaluation and training; the actual performance is the one we measure on ideal labels quality, after training a model on labels containing errors (see Sec. IV).
(vii) Assuming a state-of-the-art P2D labels collection, we expose and quantify the bias that labels errors induce on the ML models at hand.
(viii) We propose a novel neural network configuration that avoids using labels, trained with GPS against Bluetooth as pseudo-labels, and vice-versa.

In smartphone-based travel surveys, errors in labels and consequent bias in passengers transport records translates directly, e.g., in sub-optimal transport applications, design of new facilities, and operations. The above contributions expose a substantial risk of bias, identify bias due to labels error – which previous work seldom contemplated – and thus offer a methodology to reduce the need of labels for the BIBO case and beyond.

II. Related Work

This section focuses on two main bodies of literature that contribute to expose perspectives relevant for this use case: one on deployment of BLE beacons networks and signal processing for location prediction and activity classification, the other on the problem of label noise for ML classifiers. This section pinpoints candidate parameters and methods considered for designing the experimental setup of this work.

A. Bluetooth Applications

BLE stems from Bluetooth and WiFi protocols, and specializes in IoT applications; the communication is one-to-many, involves few bits of data to be broadcast frequently, and requires no pairing operation with other devices. All these properties make BLE technology particularly suitable for proximity detection [8]. Although transport mode detection is heavily unbalanced towards other sensors, such as GPS and INS [8], BLE and WiFi are considered promising technologies even in complex multimodal transport chains [22], [23], [24], [25], [26], [27]. For example, Bjerre-Nielsen et al. [27] perform transport mode detection based on received signal strength (RSSI) from Wi-Fi and Bluetooth signals, measured in decibels. The study analyzes and compares three supervised classifiers: random forest, logistic regression, and support vector machines. None of these methods involves artificial neural networks.

For implicit BIBO classification, Narzt et al. [6] propose an architecture where Bluetooth receivers are inside the bus while passengers carry a BLE device. The study carries out several experiments to recreate bus-space realistic conditions and analyses multiple configurations for Bluetooth receivers and device positions. No real users nor vehicles are involved in the study. The conclusion cautiously supports the hypothesis that larger-scale deployment of such a system is feasible. To further investigate potential interactions with the environment, the study highlights the need for a survey under realistic conditions from a larger-scale deployment perspective. An important limitation of Narzt et al. setup is that current BLE transmission in smartphones enables mac address randomization which means devices cannot be identified when not activated. Therefore, in our study, each smartphone is a receiver of signals generated with BLE devices installed in the transport infrastructure, i.e., buses and bus stops.

An independent, complementary, and substantial body of literature focuses on multiple sensors and algorithms for Mobile Anchor Node Assisted Localization [28], where WiFi and BLE signals are extensively studied in general, and in particular for indoor tracking [29]. Among the methods available, geometric approaches are widespread, e.g., based on the Friis equation [30], and trilateration [31]. These methods rest on the knowledge of each device position and radio-signal propagation physics to approximate a receiver's location based on reception strength. Prevalent RSSI fingerprints approaches are ML-based, e.g., on k-nearest-neighbor and Kalman-Filters (KL) [32], [33], [34]. These algorithms rely on mapping a geo-spatial context with a sample of signal-strength-records, received from the devices on the range; grid resolution on the mapped space and signal-sample-size depend on the location accuracy required by the use case.

A natural extension of these technologies in the field of intelligent transport systems, is the study of vehicles to anything (V2X) communication. Whereas Bluetooth in general is not considered optimal for bi-directional communication due to slow paring process [35], BLE technology is substantially different and is able to trigger events in smartphones' OS, without any paring operation [36].

In summation, the above works indicate several pre-requisites for successful BLE application: (i) BLE signal transmission rate above 0.3 Hz; (ii) The density of the Bluetooth beacons network above one device every 30 square meters; (iii) Appropriate imputation of RSSI readings.

B. Noisy Labels in Machine Learning Classifiers

The problem of noisy data receives a lot of attention from the research community. The cause of noise in labels is manifold and use case dependent. For example, crowd-sourced labeling of images relies on expertise and attention of labelers, which they may not always have [19]. Similarly, in prompted recall surveys, users validate travel diaries with different dedication levels, and may therefore, negatively affect the quality of what is often perceived as ground-truth [8]. Consequently, noisy labels in turn negatively affect the classification accuracy of supervised or semi-supervised ML methods, which depend on these labels in the training process.

Previous systematic studies on noisy labels compare multiple supervised classifiers on multiple synthetic datasets [18], and analyze how robust learning algorithms are to noise [37], [38], [39]. Another research line works on noise cleansing or labels correction methods [40], [41], [42]. Numerous alternative approaches exist for improving classification accuracy in the presence of noisy labels, for example: (i) To pinpoint wrong labels, majority voting across multiple neural networks [43]. (ii) To learn labels' noise distribution, specialized layers for artificial neural networks [19], [44], [45]. (iii) To predict the noise affecting training, conditional noise models [46]. (iv) To reduce the number of labels necessary for training, semi-supervised approach achieved with generative models [47]. (v) To leverage on existing high-quality sub-set of labels, propagation methods of these labels [48]. (vi) To learn labels on the fly and reduce human errors, graph-based label propagation methods [49].

III. Methods and Materials

To assess BIBO error tolerance under the experiment setup, first we need to understand how users collect faulty ground-truth, and then use this knowledge to derive a Monte Carlo process generating the same noise on the labels. Fig. 1 describes the methodological process we adopted; Figure 2 presents the BIBO platform we designed, implemented and deployed for data collection.

A. Procedure

The experiment took place in a private area because the two autonomous buses in use were not yet allowed in public roads, and no public service operations were in place. Therefore, users were recruited among the campus personnel and students.

First, we distributed a paper-based form to each user. The form included the experiment description and the information on data collection and use exclusive for research purposes (GDPR complainant). Then we briefed each participant on the following steps: (i) Install on smartphones the application we published to the application stores (for beta testing). (ii) Read general conditions and grant the application permission to access smartphone sensors and activity recognition. The latter performs transport mode detection [50], [51]. (iii) Wear a sleeve number to ease the ground-truth collection from video recordings. (iv) Use the transport network with the commitment to enter and exit the bus more than once, and with the possibility of walking between bus stops. (v) Count the total number of stops, defined as the discontinuities between transportation modes, and expect a message stating our count, in the following days, with the request to validate or correct such a count.

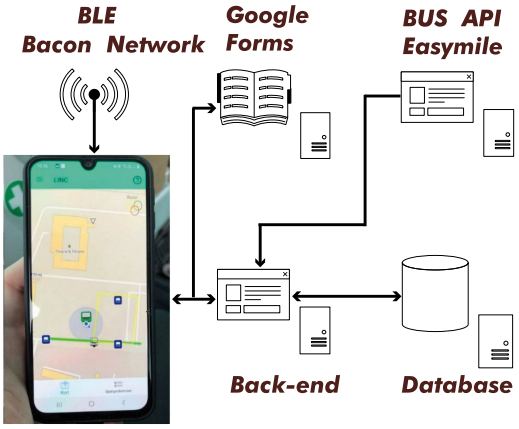


Fig. 2. Sensing platform.

Next, we answered any questions raised by the participants, and from all the participants willing to participate we collected a paper-based signed authorization to proceed with experiment and data collection.[1]

B. Wizard of Oz (WoZ) for P2D Ground-Truth Collection

In this case, WoZ refers to the experimenter pretending that a BIBO system is operational on the test-bed [52]. The role of the user is to validate the measurements of such a BIBO system. Therefore, the user is briefed to count how many times he or she stopped according to the definition of stop provided in Sec. III-A. To observe the P2D validation dynamic, the experimenter then provides WoZ's count to the user.

C. Qualitative Survey on P2D Error Types

Active ground-truth collection P2D, which users provided in the days following the experiment, included fourteen valid replies, which is twice the number recommended for similar qualitative studies [53].

To collect P2D labels from users after the experiment and link each user's feedback to the other data collected from smartphones, we rely on electronic forms with a pre-filled unique identifier corresponding to the user.

[1]This project is a social science study, includes data and numbers only, is not a health science project, and does not include human biological material nor medical devices. Consequently, in Denmark, where the data collection took place, the Health Research Ethics Act provides a dispensation for notification to any research ethics committee.

To explore the users' commitment and the type of errors present in P2D ground-truth collection, first we counted the total number of stops for each user from video-recordings; next, we introduced a level of noise in these counts before submitting the validation request, on a random sample of users. For the noise distribution, we assumed that validation errors could be Poisson distributed, similarly to OD matrix counts [54], as each error event is discrete and has minimal probability.

D. Experiment Setup and Data

The possibility of replicating beacons' density of indoor settings, which should be $> \frac{1}{30\ m^2}$, is not realistic from this use case's scale-up perspective (see Sec. II-A). However, installing devices both on the bus and on bus stops, which is more realistic, allows a temporary and nearly-optimal density of the beacons' network, at least between passengers' boarding and alighting, and when the bus stations in front of a bus stop.

The setup consisted of two autonomous vehicles operational on two distinct routes and three bus stops. To allow passengers' transfer between the buses, one bus-stop was shared between the routes, additionally sharing a segment of the test track; during the experiment, the buses' assignment to the route has been switched for technical problems, similar to real world settings. The BLE beacon network counted one device per bus and one device per bus stop, with five devices in total (see Fig. 2). Each device transmitted at the rate of 1.667 Hz and $-8\ dBm$ power. As these two settings affect the battery life of BLE beacons, the decision considers a realistic battery life expectation above one year, within the frequency recommendations from indoor studies (see Sec. II-A).

1) Dataset Signals and Baseline Collection: Smartphone onboard sensors collected trajectories for a total of $13,723\ points$. We stored for each of these timestamps: User Pseudonym, GPS longitude and latitude; 5 RSSI readings from the BLE beacons network, one for each of the 5 BLE devices, and the transport-mode as detected by top-of-the-shelf INS-based classifiers available on both Android and iOS operating system. The sampling rate for all the sensors is approximately 1 second. We don't know what is the sampling rate underpinning the proprietary INS-based classification. The literature shows a range between 20 Hz and 50 Hz. Raw acceleromenter, gyroscope and magnetometer data were out of the scope. Completed the data collection, we found a high number of unavailable trajectories, approximately $\frac{1}{3}$ of the total: Four users did not grant the permission to access location sensors, turning in no database records.

2) High Quality Ground Truth: To count passengers' flow, we installed a high-resolution video-camera pointing to the buses' doors at each bus stop as the principal ground-truth. However, the three cameras in combination also allowed the full surveillance of the track. A problem with the video-cameras, prevented the determination of high quality ground-truth for three users. Using video footage as ground-truth for the trajectories successfully collected from the remaining users. On each point we provided a set of binary labels consistent with the BIBO model [6]: inside (BI) or outside (BO) the bus.

3) Sensing Platform: The smartphone sensing platform's main components are the front-end applications and back-end. The front-end is specific, or native, for Android and iOS. The apps contain the following features: data collection from onboard sensors and native APIs, such as users' transport activities classification; data transfer to the back-end from a local buffer that avoids data loss in case of external connectivity problems; and lastly, real-time tracking of buses on a map, with bus stops. The sensors we target are GPS, and BLE signal strength perceived from the BLE beacons network, which is external to the smartphone and independent. To improve smartphones' battery efficiency, we monitor users' activities. We switch on and off smartphone GPS and data transfer, when the user is active and inactive. We collect such an INS-based activity recognition that any OS offers via APIs to discriminate between states, such as: automotive, bicycling, walking, running, stationary and unknown [50], [51]. The back-end, which includes several specialized APIs, is responsible for exposing the information from the autonomous vehicles to the smartphone and handing the data from smartphones to the database.

As opposed to another architecture presented for BIBO [6], where users carry BLE devices, and buses are equipped with signal processing devices, we decide to follow the architecture of smartphone-based travel surveys [8], where users carry their smartphone, which is also a signal processing device, while buses are equipped with BLE devices. This configuration presents two main advantages. First, we extend the beacons' network outside the bus, at the bus stops. Second, we allow users to carry their phones and not interfere with their normal behavior while picking up signal from multiple sources. From the perspective of a larger-scale deployment, the experimental setup seems more realistic under these two conditions. The installation cost of a beacon device should be a fraction of the Raspberry-Pi deployed in [6]; furthermore, users should carry their smartphones only and not a new device. This last element introduces a new random variable: The varying quality of sensors installed in different smartphone models, and the sample collected in this experiment is not representative of this broad population. Thus, to collect consistent data, we rely on the standardization of sensors and protocols represented on the aforementioned OS.

E. Data Preparation and Classifiers for BIBO

To assess BLE beacons' signal performance in determining users' presence inside (BI) or outside (BO) the buses, we use INS and GPS as benchmarks.

From smartphones' OS we collect the binary classification automotive vs. everything else, compatible with BIBO in this context, which is based at least on accelerometer.

From GPS we extract the following features: distance between points, bearing, and speed [55], [56], which we process as time series. For BLE beacons we apply the same methodology. Table V presents the list of features collected in 10 seconds sliding window.

Concerning the choice of ML models, we would like to highlight that we choose two top-of-the-shelf algorithms [8].

To benchmark the performance of the underlying data streams, i.e., BLE against INS and GPS, we consider these two algorithms sufficient baselines.

1) Framework: Scikit-learn is a popular python-based framework that includes several effective ML models. Random Forests (RF) represent a reliable and scalable supervised method for this task [57]. At the same time, Multi-layer perceptron (MLP) can be considered a building block of generative models, which can operate semi-supervised or unsupervised [47]. Therefore, we include these two supervised classifiers in the study. For more advanced neural network configurations [58], we rely on Pytorch. The following sections present further details, on preparation, training, and validation of the classifiers.

2) Random Forest: RF evolve from decision tree predictors, averaging results from multiple of these predictors. The effect is a more accurate classifier less prone to over-fitting. The training phase starts with bootstrapping [59], which consists of several sub-samples with replacement from the training dataset. Each training sub-set is then split into in-bag [59] (IB) and out-of-bag [59] (OOB). The latter's size is one-third of such a sub-set, while the former accounts for the rest. A decision tree is constructed from each IB, while the attributes are sampled randomly to determine the decision split [59]. Finally, the RF output is aggregated over all individual trees, and the output is the class with the highest average probability, whereas in classical majority voting the output is the most common class prediction among trees.

3) Multi-Layer Perceptron: Perhaps we can consider it the most simple feed-forward artificial neural network [60]. MLP incorporates multiple layers for logistic regression. Multiple perceptrons, or neurons, compose each layer and handle nonlinearities through activation functions, such as sigmoids and rectified linear units (ReLU). For classification, each neuron's weight and bias is trained by minimizing the cross-entropy between the class predicted by the network and the ground-truth. These parameters are iteratively updated at each classification attempt, defined epoch, by back-propagating the resulting stochastic gradient towards the cross-entropy local minimum.

4) Split-Brain Variational Autoencoder (SVAE): Inspired by the split-brain autoencoder architecture [61], we define two variational autoencoders. The encoder and decoder of the first are defined as $q_\phi(z|d_{\text{BLE}})$ and $p_\theta(d_{\text{GPS}}|z)$; the encoder and decoder of the second, as $q_{\hat{\phi}}(\hat{z}|d_{\text{GPS}})$ and $p_{\hat{\theta}}(d_{\text{BLE}}|\hat{z})$. d_{BLE} and d_{GPS} identify any signal extracted respectively from Bluetooth low-energy and GPS. This architecture (see Fig. 3) reflects a simple form of causal directed acyclic graph [62], which we implement with a convolutional neural network: The position in space/time of each smartphone in the experimental context (cause) determines the BLE signal strength that each smartphone detects (effect).

This two variational autoencoders are trained minimizing at the same time [63] the two Variational Maximum Mean Discrepancy (MMD) losses [64] (1), which are expressed in (2) and (3) as $\mathcal{L}_{\text{MMD}}$ and $\mathcal{L}_{\hat{\text{MMD}}}$, where $\beta > 0$ in the first term is a scaling factor similar to $\beta-$**VAE** [65] and the second term is the reconstruction loss. MMD is expressed in (4), where

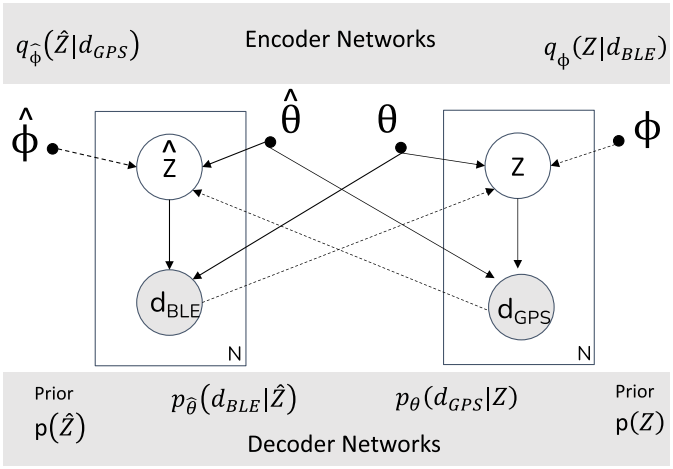


Fig. 3. Split-brain variational autoencoder (SVAE).

$k(z,z') = e^{-\frac{||z-z'||^2}{2\gamma^2}}$ and γ is an hyperparamenter controlling the smoothness of the Gaussian kernel k (see Tab. III). k' is a true sample from a standard distribution. In (1), σ_1 and σ_2 are optimal weights found together with ϕ, $\hat{\phi}$, θ, and $\hat{\theta}$ (2), (3), leveraging the same back-propagation algorithm (see Appendix, Alg. 1). With MMD we aim to match the latent space distribution of all the possible signals $q(Z)$ (5), which reflects distribution over all the observed signals d_X, with the distribution $p(Z)$. The choice of MMD divergence (4) for, e.g., the loss $\mathcal{L}_{\text{MMD}}$ (2), over the Kullback–Leibler divergence of the loss $\mathcal{L}_{\text{ELBO}}$, the Evidence Lower Bound (ELBO) [47], derives from the better performance in disentangling classes representations in the variational autoencoder's latent space.

$$\mathcal{O}_{SVAE} = arg\min(\frac{1}{2\sigma_1^2}\cdot\mathcal{L}_{\text{MMD}} + \frac{1}{2\sigma_2^2}\cdot\mathcal{L}_{\hat{\text{MMD}}}) + \ln\sigma_1 + \ln\sigma_2 \quad (1)$$

$$\mathcal{L}_{\text{MMD}} = \beta\cdot\text{MMD}(q_\phi(Z)||p(Z)) + \mathop{\mathbb{E}}_{p_{\text{data}}(d_{\text{GPS}})}\cdot\mathop{\mathbb{E}}_{q_\phi(Z|d_{\text{BLE}})}\cdot[\log p_\theta(d_{\text{GPS}}|Z)] \quad (2)$$

$$\mathcal{L}_{\hat{\text{MMD}}} = \beta\cdot\text{MMD}(q_{\hat{\phi}}(\hat{Z})||p(\hat{Z})) + \mathop{\mathbb{E}}_{p_{\text{data}}(d_{\text{BLE}})}\cdot\mathop{\mathbb{E}}_{q_{\hat{\phi}}(\hat{Z}|d_{\text{GPS}})}\cdot[\log p_{\hat{\theta}}(d_{\text{BLE}}|\hat{Z})] \quad (3)$$

$$\text{MMD}(q_\phi(Z)||p(Z)) = \mathop{\mathbb{E}}_{p(z),p(z')}[k(z,z')] + \mathop{\mathbb{E}}_{q(z),q(z')}[k(z,z')] - 2\mathop{\mathbb{E}}_{p(z),q(z')}[k(z,z')] \quad (4)$$

$$q_\phi(Z) = \mathop{\mathbb{E}}_{p_{\text{data}}(d_X)} q_\phi(Z|d_X) \quad (5)$$

Since this architecture allows the reconstruction of the BLE features from the GPS features and vice-versa, instead of labels, we train this neural network using one group of sensors, i.e., BLE, as pseudo-labels for the other group of sensors, i.e., GPS, and vice-versa.

TABLE I
RANDOM FOREST HYPERPARAMETERS SEARCH SPACE

Number of estimators	$\in \{10, 20, 100, 200, 500\}$
Max features	$\in \{\text{auto}, \text{sqrt}, \log2\}$
Max depth	$\in \{3, 4, 6, 7, 8\}$
Criterion	$\in \{\text{gini}, \text{entropy}\}$

TABLE II
MULTI LAYER PERCEPTRON HYPERPARAMETERS SEARCH SPACE

Hidden layers/sizes	$\in \{1 \text{ layer } (L) \rightarrow [50 \text{ neurons } (N)],$ $3L \rightarrow [10N, 50N, 10N],$ $4L \rightarrow [10N, 50N, 50N, 10N]\}$
Learning rate strategy	$\in \{\text{constant}, \text{invscaling}\}$
Learning rate coefficient	$\in \{10^{-2}, 10^{-3}\}$
Activation funcions	$\in \{\text{ReLU}\}$
Optimizer	$\in \{\text{adam}\}$

TABLE III
SVAE HYPERPARAMETERS SEARCH SPACE

Encoder	
Convolutional Neural Network (CNN) Layers	$\in [1, 3]$
Activation Function	$\in ReLU, LeakyReLU$
Fully connected Layers	$\in [0, 4]$
Dropout	$\in [0.25, 0.45]$
Decoder	
Transposed CNN Layers	$\in [1, 3]$
Activation Function	$\in ReLU, LeakyReLU$
Fully connected Layers	$\in [0, 4]$
Dropout	$\in [0.25, 0.45]$
Optimizer	Adam
Epochs	$\in [50, 300]$
Batch Size	$\in [8, 128]$
Learning Rate	$\in [10^-4, 10^-1]$
Dropout	$\in [0.25, 0.45]$
γ^2_{MMD}	$\frac{\text{Batsh Size}}{2}$
β	$\frac{1}{\text{Batsh Size}^2}$

The resulting latent space $Z \oplus \hat{Z}$ represents the space $d_{GPS} \oplus d_{BLE}$, but with only few dimensions. Such a latent space can represent the BIBO classes disentangled and enable unsupervised classification. In this work we use a Density-based clustering based on hierarchical density estimates (HDBSCAN) [66], which can identify clusters of varying density.

5) Optimal Model Hyperparameters Search: To perform this task on RF and MLP we used *GridSearchCV*, a specialized library available in Sklearn. To obtain a set of optimal hyperparameters, we perform a 5-fold cross-validation on the training-set exploring those that Table I and Table II describe for RF and MLP. In a following step we train the classifier on the training-set, fixing these optimal hyperparameters, and we perform the evaluation out-of-sample (OOS) on the test-set. Sec. III-G provides further details on this process within the simulation of ground-truth collection errors causing flipping-labels.

For SVAE, optimal hyperparameters search was performed manually and focused exclusively on convolutal neural netowrks (CNN) [60] (see Tab. III).

6) Validation Process: The risk of information spill-over between training- and validation-set is higher when working with time-series. Reference [67] shows that the violation of the OOS principle is not rare in the existing literature. Such a violation yields a virtual higher performance when evaluating a classifier, resulting in a biased measurement. Even in the assumption of non-violation of the OOS principle, researchers have several options for assessing a classifier, such as hold-out, leave-one-out, and cross-validation. (i) In the hold-out case, typically, the training-set should use approximately $\frac{2}{3}$ of the dataset; the validation-set, the remaining $\frac{1}{3}$. Training and validation proceed only once and yield the model performance based on the sole validation-set. (ii) In the leave-one-out case, the training-set should use a dataset's random sample of size $M - 1$, where M is the dataset's cardinality; the validation-set, the remaining one sample. Training and validation proceed M times and yield the model performance as a distribution over M-validations. (iii) In the cross-validation case, the dataset is split into N equal partitions; the training-set uses N-1 partitions, while the validation-set uses the remaining one partition. Training and validation proceed N times and yield the model performance as a distribution over N-validations. The approach (i) is computationally light-weight, but the resulting performance estimation might be negatively biased; (ii) is unbiased but could present a large performance variance, and the method is computationally expensive; (iii) is a good compromise between the previous two [68]. Sec. III-G explains how our simulation combines these three methods with the hyperparameters grid-search to provide an optimal and unbiased performance estimation and how we sample training- and validation-set to avoid OOS violation. Sec. III-H, describes the validation for SVAE, which is an unsupervised method, thus allows more flexibility in the use of data.

F. Comparison Metrics

To compare the results of the classifiers, we chose the traditional area under the receiver operating characteristic (AUC). AUC's domain is $\in [0, 1]$. For BIBO binary classification, AUC is strictly around 0.5 for Random Classifiers. AUC is > 0.5 for a classifier consistent in classifying correctly the target class. AUC is < 0.5 for a classifier consistent in classifying the opposite of the target class. Motivations and details on what AUC is and why we chose it over other metrics, are available in Appendix (see Sec. A).

G. Classification Performance Over Error Simulations

After data preparation, as summarized in Alg. 2, we proceed with the simulation (MS), as detailed in Alg. 4.

With the following three steps, we obtain labels augmentation: (i) Systematic sampling user-by-user; (ii) Sampling of errors' number per user; and (iii) Error propagation on the labels of each users' trip. At each run we train and evaluate ML classifiers against the Random Classifier, over features extracted from BLE sensors, versus features extracted from GPS sensors. We ensure the OOS validation principle on both grid-search and methods' evaluation by randomly sampling 20% of the users and then picking all their trajectories to

compose the validation set. Thus, we take the complement for the training-set.

We yield performance's unbiased estimation by applying an hold-out scheme within each run, where the training partition allows a grid-search (see Sec. III-E.5) for optimal models' hyperparameters through a 5-fold cross-validation. The optimal hyperparameters are used for models' evaluation OOS on the validation-set. Following a fine-grained step sequence of error rates inducing total flipped-labels $\in (0\%, 100\%)$, we repeat the error propagation process 100 times per step, and each time we plot the mean of the model evaluation performance (see Figs 3-6).

H. SVAE Classification Performance

After data preparation, we sample 20% of the users to compose the test-set. The remaining users compose the training-set. After training SVAE, we process the test-set and we apply HDBSCAN clustering in the SVAE's latent space. Based on the test-set we assign the clusters to the BIBO state. Next, we assign to these clusters the points of the whole dataset. Since labels are never used in the training phase, to compute the distribution of this unsupervised classification, we can include the points of the training-set.

IV. Results

In this section we organize the results according to the three problems listed in Sec. I, directed to answer the research question.

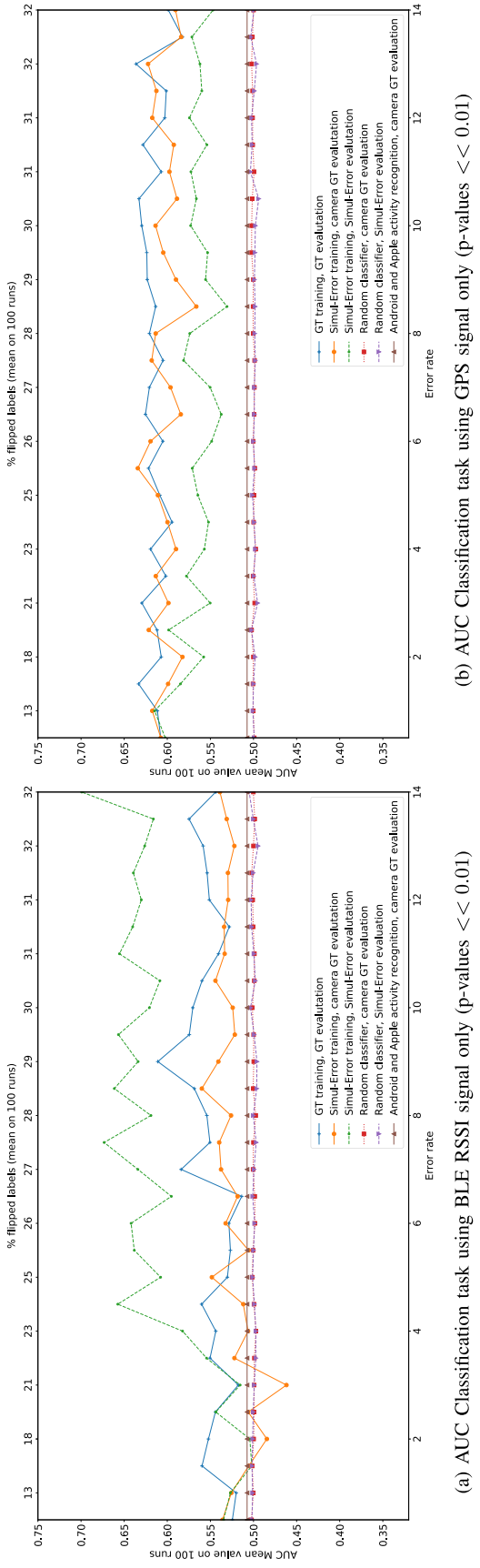
A. People Errors Types During Ground-Truth Collection

The experiment included video recordings of the ground-truth for eighteen users in total, that we used for the P2D validation experiment. The resulting confusion matrix on error distributions for labels (see Table IV), shows that 50% of the received replies were perturbed. Nearly 60% of the user modified the counts, while the remaining population confirmed the counts as received. One user confirmed the perturbed count, and two users modified the correct counts. Overall, more than 40% of the validations contained at least one error, with average 0.7. After users' correction, the number of errors is six; before the correction, seven.

These results show that labels in P2D interactions can quickly flip from correct to wrong and hardly from wrong to correct. We could test only one noise level on a relatively small sample of users: a broader study would be precious and require a specific effort that is out of this study's scope. However, based on these results, we can already simulate the impact of these types of errors on ML algorithms.

B. BIBO Classification Performance

This section presents BLE- and GPS-based classifier's performance (CP) compared with INS-based classifiers, as baseline. It shows RF and MLP algorithms' performance, and their bias due to training with flipped labels.

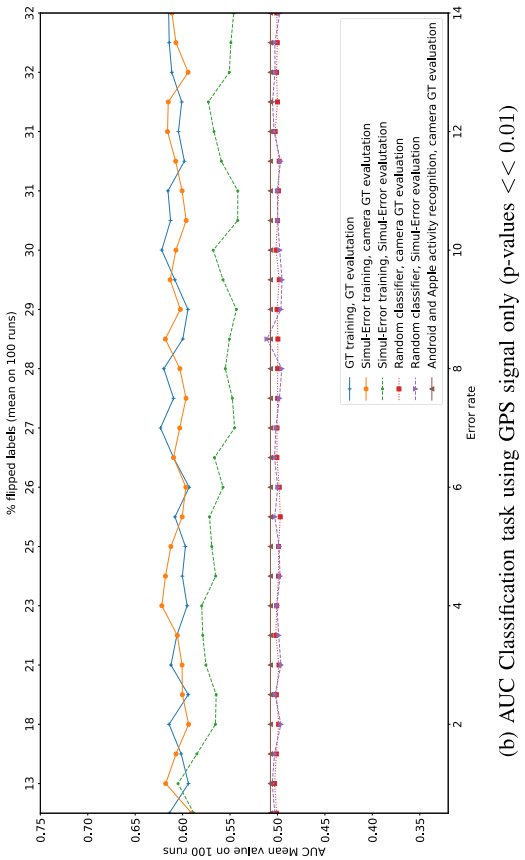


(b) AUC Classification task using GPS signal only (p-values $<< 0.01$)

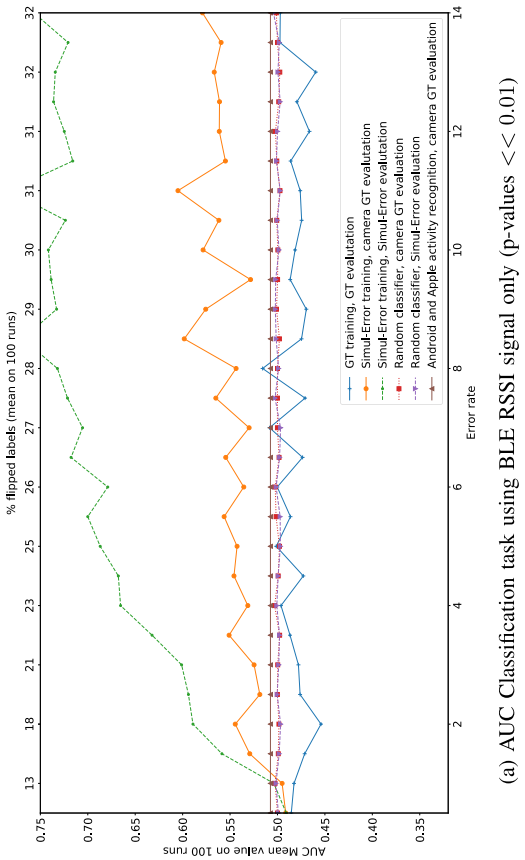
(a) AUC Classification task using BLE RSSI signal only (p-values $<< 0.01$)

Fig. 4. Random forest one flip experiment.

The "Flipping-label" bias emerges when we compare algorithms trained and evaluated using ideal labels, which are

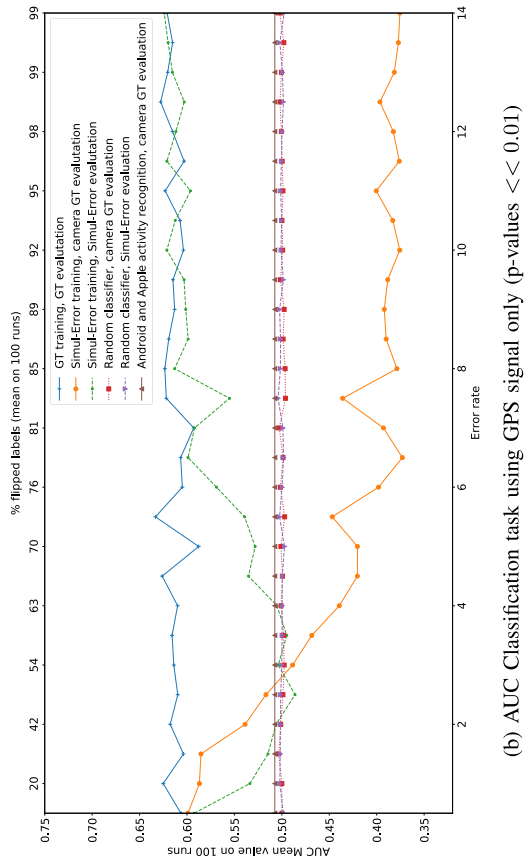


(b) AUC Classification task using GPS signal only (p-values $<< 0.01$)

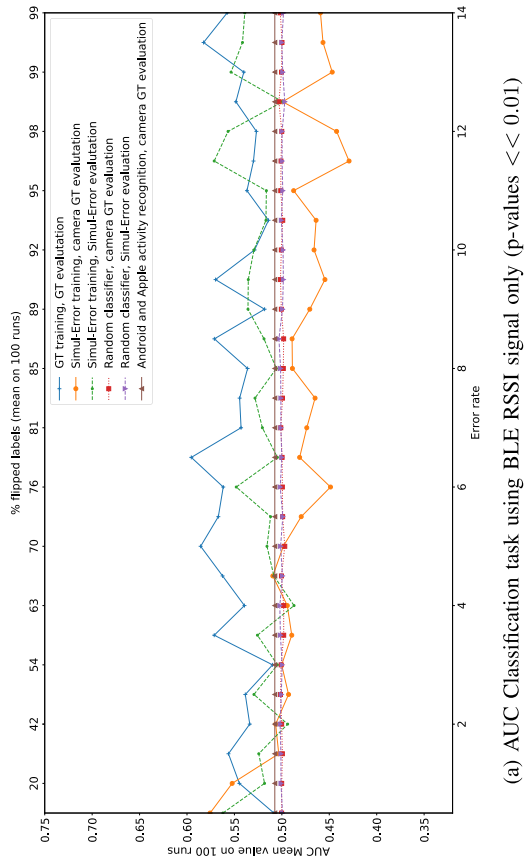


(a) AUC Classification task using BLE RSSI signal only (p-values $<< 0.01$)

Fig. 5. Multi layer perceptron one flip experiment.



(b) AUC Classification task using GPS signal only (p-values $<< 0.01$)



(a) AUC Classification task using BLE RSSI signal only (p-values $<< 0.01$)

Fig. 6. Random forest two flips experiment.

extracted from high-quality ground truth, and the same algorithms trained with labels containing a controlled fraction of errors and evaluated with ideal labels. (i) "Android and Apple activity recognition, camera GT evaluation", (ii) "GT

TABLE IV
PERSON TO DEVICE GROUND-TRUTH VALIDATION

	Modified	Confirmed	Correct	Wrong	
Perturbed	6	1	3	4	7
Not Perturbed	2	5	5	2	7
	8	6	8	6	14
	14		14		

training and GT evaluation", and (iii) "Simul-Error training, camera GT evaluation" plots describe this case in Figs 3-6. These Figures show the number of wrong labels in the abscissae and AUC error in the ordinate. We show two metrics for the number of wrong labels. The error rate, on the bottom, represents the Poisson distributed mean error. This error distribution translates into a percentage of wrong labels on the total labels, which is reported on the top of the Figure.

Results show that the INS-based BIBO CP is 0.51 AUC, significantly different and very close to the Random Classifier, which is 0.5 AUC.

GPS-based CP is $> 0.60 \pm 0.02$ AUC both for RF and MLP.

(i) In the "one flip" error type – where wrong users' counts will cause some segments to flip their correct class and match the previous or following segment's label – the bias is significant and > 0.01 AUC. In this experiment, the maximum number of labels we can flip is $\leq 32\%$ of the total (see Figs 4b-5b).

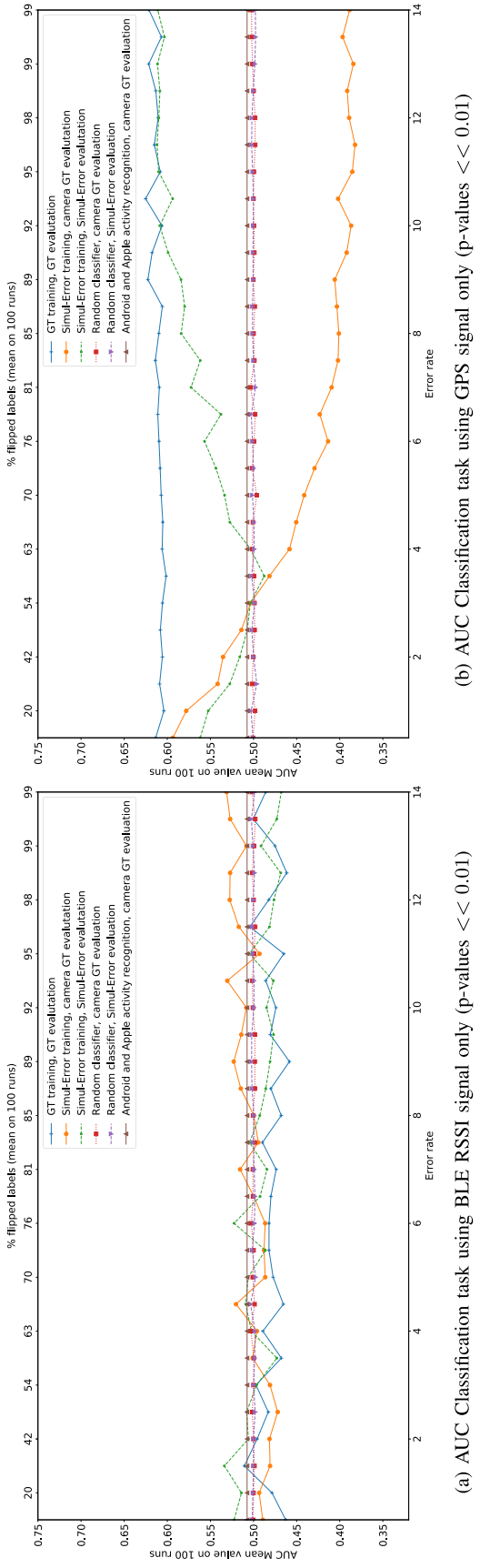
(ii) In the "two flips" error type – where the discrete number of errors sampled from Poisson propagates to a random sample of trip segments by flipping the label from the correct class to the alternative – the bias is close to 0 when the ratio of flipped labels is around 0%. The bias is approx. 0.3 (maximum) when the ratio is around 100% (see Figs 6b-7b).

BLE-based CP is $> 0.55 \pm 0.02$ AUC for RF and $> 0.48 \pm 0.02$ AUC for MLP.

(i) In the "one flip" labels error type the RF bias is significant and > 0.03 AUC (see Fig. 4a); MLP bias is $\in [0.01, 0.15]$ (see Fig. 5a).

(ii) In the "two flips" labels error type the RF bias is significant and close to 0 when the ratio of flipped labels is around 0%. The bias is 0.15 (maximum) when the ratio is around 100% see Fig. 6a). MLP bias is $\in [0.01, 0.15]$ (see Fig. 7a).

For BLE and GPS classification, the Random Forest algorithm trained and evaluated with ideal labels performs significantly better than the INS-based classifier. In the same conditions, BLE-based multi-layer perceptron performs consistently worse than the Random Classifier. By switching consistently BLE-based MLP output class, this classifier would be comparable with the INS baseline. As opposed to the other examples – including the BLE-based RF classifier – BLE-based MLP proximity to the Random Classifier may justify its flat performance and bias with respect to various labels' flipping rates. In contrast, GPS-based MLP performs similarly to RF. Overall, the low performance of production-level



(b) AUC Classification task using GPS signal only (p-values $<< 0.01$)

(a) AUC Classification task using BLE RSSI signal only (p-values $<< 0.01$)

Fig. 7. Multi layer perceptron two flip experiment.

INS-based classifiers reflects this challenging and realistic experiment setup.

C. BIBO Resilience to Label Flipping

This section presents BLE- and GPS-based classifiers' performance and their bias due to training with labels containing a controlled fraction of errors. (i) "Simul-Error training, Simul-Error evaluation", (ii) "Simul-Error training, camera GT evaluation", and (iii) "GT training and GT evaluation" plots describe this case in Figs 3-6. As in the previous section, Figs 3-4 refer to the "one flip" error type; Figs 5-6, to the "two flips". The "model" bias emerges as the distance between the "actual" and the ideal performance, comparing models' evaluated with labels containing a controlled fraction of errors and the same algorithms evaluated with ideal labels. The "unperceived" bias emerges as the distance between the "perceived" and ideal performance of a model, when we compare algorithms evaluated with labels containing a controlled fraction of errors and the same algorithms trained and evaluated with ideal labels.

When flipped labels rate increases from 0% to 30%, for both error types, GPS-based RF and MLP classifiers show a constant increase of the "unperceived" bias, which is approx. $\in [0, 0.1]$ AUC; beyond 30%, in the second error type, the models' maximum unperceived bias corresponds to 50% of flipped labels. At 100% of flipped labels, the bias is equivalent to the 0% case, which is approx. 0 AUC. The model bias is approx. null at 0% and 50% of flipped labels rates, and it is maximum at 100%. The interval of flipped labels $\in [0\%, 30\%]$ is the most interesting. For the "one flip" error type, the model bias is constantly increasing approx. between 0 and 0.1. For the "two flips" error type, somewhere $\in [20\%, 40\%]$ of flipped labels, the model bias presents its maximum value, which is between 0.05 and 0.1 (see Figs 4b, 6b, 5b, and 7b).

For the "one flip" error type, BLE-based RF and MLP classifiers' unperceived bias increases significantly for RF, and dramatically for MLP, after 20% of flipped labels. For RF this bias is $\in [0, 0.1]$; for MLP, $\in [0, 0.3]$ (see Figs 4a and 5a). The model bias in this case is negligible compared to the unperceived bias. For the "two flip" error type, BLE-based classifiers' behavior is consistent with the description provided for GPS-based classifiers. However, due to the proximity between ideal and Random Classifiers plots, both unperceived and model bias are smaller.

The distance between the origin and the point where the "actual" performance remains below the curve of the Random Classifier, measured on the abscissae, is a measure of the model tolerance to flipped labels. The BLE-based RF classifier seems mostly above the Random Classifier up to 30% of flipped labels in both error types (see Fig. 4a, and Fig. 6a).

D. BIBO via Clustering on the SVAE Latent Space

The classification performance of the unsupervised HDBSCAN classifier III-E.4 after dimensionality reduction via SVAE, relies on sensors' signals device-to-device as pseudo-labels, instead of person-to-device collected labels. d_{GPS} features include smartphone speed, bearing, and time interval between two GPS points. d_{BLE} features include RSSI from the devices installed at the three bus stops. Surprisingly, we yield the best results excluding the BLE signals from the devices installed in the buses. HDBSCAN default configuration with euclidean metric yields four clusters on SVAE latent space. After visual examination, we assign three clusters to BI class and the remaining to BO class. The AUC distribution over the users in the test set only is 0.63 ± 0.07. The AUC distribution over the users in the whole dataset that results from automatic assignment of the points to these clusters, is 0.58 ± 0.05. The results are comparable to the performance of the RF supervised classifier trained with GPS features and flipped labels $\leq 20\%$ (see Figs 4b, 6b). In the open area where the experiment took place, a bus between two bus stops, users in front of the bus stops or inside the bus, may create meaningful signal variation patterns for BLE devices at the bus stops rather than devices in the buses. For the future, buses GPS could allow the comparison between bus/smartphone distance with the BLE RSSI from the devices installed in the buses.

TABLE V

FEATURES [69] EXTRACTED FROM SENSORS' SIGNALS, WITHIN 10 SECONDS SLIDING WINDOW: BLE RSSI AND GPS SPEED, SPACE- AND TIME-GAP

1	Mean value
2	Max value
3	Min value
4	Position where the minimum value is located
5	Position where the maximum value is located
6	Amplitude between min and max value
7	Number of points beyond one standard deviation
8	Number of points below one standard deviation
9	Number of points above one standard deviation
10	Number of peaks in 10 seconds window
11	Number of peaks 5 seconds window
12	Number of peaks above 1 standard deviation
13	Peak distance within the same time window
14	Slope

V. CONCLUSION

This paper investigates the realistic large-scale deployment of a BIBO system based on BLE beacons, and analyzes the sensitivity of its ML components to errors on labels. The experimental setup recreates challenging conditions with a high density of BLE signals and low speeds of both users and vehicles present in the transport network.

We test hypotheses on person-to-device labeling types of error, typical of current smartphone-based travel surveys. We find that users' validation errors may affect both wrong and correct predictions. In the first case users are often unable to correct all the errors. In the second case, users introduce errors by amending correct predictions. Results show that, in urban dense context, the likelihood of wrong labels presented to users for validation is very high, and so is the risk of bias for classifiers trained with flipped labels.

To reduce such a risk and increase the perception of models' bias, we evaluate GPS-, BLE- and INS-based classifiers, the latter available in native Android and iOS APIs.

On the one hand, BLE is significantly better than the INS baseline, and show resilience to labeling errors up to 30%. Overall, Random Forest performs significantly better on both BLE and GPS. At the same time, RF proves to be robust to noise, more on GPS than BLE. One recommendation stemming from the results on traditional supervised methods is that design and evaluation should include a sensitivity analysis on the classifiers performance with respect to flipping labels.

On the other hand, the proposed Split-brain variational autoencoder supports the potential substitution of person-to-device validated labels with device-to-device. In this case, BLE pseudo-labels supervise GPS and vice-versa, allow a disentangled low-dimension representation of the BIBO classes, and thus enable unsupervised classification applying, i.e., density based clustering. The performance is comparable with supervised classifiers trained with $\leq$ 20% of flipped labels.

Appendix A
Validation Metrics

As performance estimation metrics for binary classifiers, the literature presents a broad use of precision (6), recall (7), F1-score (8) and accuracy (9). Although these metrics are often sufficient, we introduce the measure of the area under the receiver operating characteristic curve (AUC). This curve describes the true-positive-rate (TPR) (7), which is another identification for the recall, as a function of the false-positive-rate (FPR) (10), within the domain of any possible FPR $\in [0, 1]$. We derive these metrics directly from the confusion matrix, i.e., true positives (T_p), true negatives (T_n), false positives (F_p), and false negatives (F_n).

The binary BIBO classes are quite imbalanced and the classification task is rather challenging given the experiment's realistic conditions. We recreate a congested urban context with multiple buses operating at speed similar to walking pace, in proximity to various bus stops. Whereas F1-score identifies cases where the Random Classifier is better than our classifier, AUC identifies also cases where the classifier only predicts the larger class. The domain of precision (6), recall (7), F1-score (8), and accuracy (9) is $\in [0, 1]$, the higher the value the better. AUC's domain is also $\in [0, 1]$. The interpretation of AUC coefficient for Random Classifiers results in the same distribution of the F1-score, strictly around 0.5. In contrast with F1-score, for cases where classifiers predict only one class AUC presents the same distribution of the Random Classifier. Therefore, with AUC we expect good classifiers above 0.5 threshold, with higher values being better. Below this threshold a classifier would be consistent in predicting the wrong class. Both random and trivial classifiers should score 0.5 AUC in average. To assess our simulation results against both the Random Classifier and the single-class-predictor, AUC measures how well predictions are ranked, and is invariant to scale and classification-threshold [70]. Since at this stage we are agnostic on the cost of false positives and false negatives, these two properties are not a disadvantage, as opposed to the advantages in assessing the classification performance with different levels of errors on the labels, over a large number of samples.

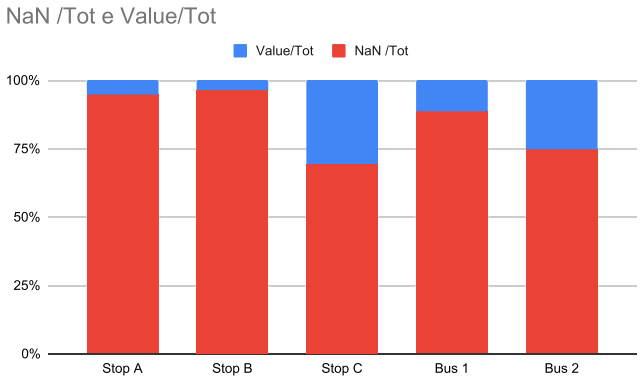


Fig. 8. Beacons RSSI timestamp with values vs. Not a Number (NaN), on total points available.

$$P = \frac{T_p}{T_p + F_p} \tag{6}$$

$$TPR = R = \frac{T_p}{T_p + F_n} \tag{7}$$

$$\text{F1} = 2\frac{P \cdot R}{P + R} \tag{8}$$

$$\text{A} = \frac{T_p + T_n}{Total\ \ population} \tag{9}$$

$$FPR = \frac{F_p}{T_n + F_p} \tag{10}$$

Appendix B
Imputation

Fig. 8 shows that BLE beacons readings are present only on a fraction of the points where GPS is present. BLE signal goes undetected when the receiver device is not in the beacon range. However, the relative position of the two devices to the user's body often leads to the same result even when the two are in range [11]. We need to perform imputation and fill the gaps whenever appropriate. Existing work shows multiple techniques. Although Kalman-filters might seem the obvious choice from indoor experience [32], this use case requires simplicity. Therefore, we consider exponential-weighted-moving-average (EWMA), which consists of computing the average of the readings within a time window, where points close to the center window have a higher weight than points at the end of the window [71]. For EWMA, the weight depends on the window size and the decay rate. From the perspective of a fingerprinting approach (see Sec. II-A), especially on large-scale deployments, we need to inform the classifier on points where the imputation algorithm could not fill the gaps. We cannot use zero, because BLE beacons signal domain can be found, empirically, in the following domain $RSSI \in (-100, -50)$ [72]. Further, smartphones record the null value when on the fringe of a BLE range, which is counter-intuitive given the signal's domain. Because of the meaning of null value and the expected amount of gaps, filling these gaps

Algorithm 1 Mini-Batch Version of the Split-Brain Variational Autoencoder Algorithm

Result: Parameters $\sigma_1, \sigma_2, \phi, \hat{\phi}, \theta, \hat{\theta}$

Input : d_{GPS}, d_{BLE}, Hyperparameters (See. Tab. III)

$\sigma_1, \sigma_2, \phi, \hat{\phi}, \theta, \hat{\theta} \leftarrow$ Initialize Parameters
repeat
  $d_{BLE}, d_{GPS} \leftarrow$ Random mini-batch of 16 data points
  $Z \leftarrow$ Random samples from noise distributions $p(Z)$
  $\hat{Z} \leftarrow$ Random samples from noise distributions $p(\hat{Z})$
  $g \leftarrow \nabla_{\sigma_1,\sigma_2,\phi,\hat{\phi},\theta,\hat{\theta}} \mathcal{O}_{SVAE}$ Grad. of mini-batch estimator (1)
  $(\sigma_1, \sigma_2, \phi, \hat{\phi}, \theta, \hat{\theta}) \leftarrow$ Update parameters using gradients g
until *Convergence of Parameters* $(\sigma_1, \sigma_2, \phi, \hat{\phi}, \theta, \hat{\theta})$
return $\sigma_1, \sigma_2, \phi, \hat{\phi}, \theta, \hat{\theta}$

with zero is likely to poison any classifier. Instead, we apply a mask filling these positions with an arbitrary constant and augment the fingerprint vector reporting a weight 0 in the position filled with the arbitrary constant and 1 otherwise [73]. (11) defines the fingerprint vector at time t as FP_t, where v_i represents the RSSI signal received from the i^{th} BLE beacon, while $v_{j_{GAP}}$ is the gap of signal from the j^{th} BLE beacon. $FP_t \in R^{m+n}$ can be augmented, resulting in a new vector $FPA_t \in R^{2\cdot(m+n)}$ (12), where $v_{j_{IMP}}$ corresponds to the signal imputation of the j^{th} BLE beacon gap, for example with EWMA, while $v_{k_{CONST}}$ represent the remaining signal gap from the j^{th} BLE beacon, filled with an arbitrary not null constant. The augmented vector FPA_t passes all the information.

$$\begin{aligned} FP_t = (&v_0, v_1, \ldots, v_{0_{GAP}}, \\ &\ldots, v_i, \ldots, v_{n_{GAP}}, \\ &\ldots, v_m), \\ &m > 0 \wedge n \geq 0 \wedge i \in (1, m) \end{aligned} \tag{11}$$

$$\begin{aligned} FPA_t = (&v_0, v_1, \ldots, \\ &v_{0_{IMP}}, \ldots, v_i, \ldots, v_{n_{IMP}}, \ldots, \\ &v_{0_{CONST}}, \ldots, v_j, \ldots, v_{k_{CONST}}, \\ &\ldots, v_m, \\ &1_0, 1_1, \ldots \\ &1_0, \ldots, 1_i, \ldots, 1_n, \ldots, \\ &0_0, \ldots, 1_j, \ldots, 0_k \\ &\ldots, 1_m), \\ &m > 0 \wedge n \geq 0 \wedge k \geq 0 \\ &\wedge i, j \in (1, m), i \neq j \\ &v_{s_{CONST}} = v_{p_{CONST}} = C, \\ &\forall s, p \in [0, k], s \neq p \end{aligned} \tag{12}$$

APPENDIX C
ALGORITHMS

This section lists the pseudo-code of the algorithms we implemented.

To train the propose Split-brain Variational Autoencoder (SVAE), we implement Alg. 1.

Algorithm 2 Data Preparation

Result: Clean trajectories, assign trip IDs, and extract standardized features for both GPS and BLE signals

Input : raw dataset (RD), true labels (TL)

Output: dataset with tripID labels and features vectors (CD)

$UULIST \leftarrow$ list-unique-users(RD)
foreach *user* $\in UULIST$ **do**
  $tripIDs_{\text{user}} \leftarrow$ clean-segment-trajectories(RD, user, TL)
  foreach $TS \in \{BLE, GPS\}$ **do**
    if $TS == BLE$ **then**
      $CD_{user} \leftarrow$ imputation-trick(D, user, TS, $tripIDs_{\text{user}}$)
    end
    $CD_{user} \leftarrow$ extract-standard-features(CD_{user}, TS)
  end
  CD.insert(CD_{user})
end
return CD

Algorithm 3 Simulate and Propagate P2D Validation Errors

Result: Faulty ground-truth vector

Input : true labels vector (TL), unique users list (UULIST), features-from-pre-processed-dataset (FCD, see Alg. 2)

Output: flipped labels vector FL

foreach *user* $\in UULIST$ **do**
  /* draw errors number from Poisson distribution */
  $NE \leftarrow$ draw-from-Poisson(ERR)
  /* Draw NE random TripIDs, as mislabeled trips */
  $\text{WrongTID}_{user} \leftarrow$ draw-random-tripIDs(NE, FCD_{user})
  /* Copy TL and flip labels for each trip drawn in the previous step */
  $FL \leftarrow$ TL
  foreach *trip* $\in WrongTID_{user}$ **do**
    $FL_{trip} \leftarrow$ flip-labels(FL_{trip})
  end
end
return FL

For the simulation, Alg. 2 refers to the data preparation and Alg. 3 to the error simulation and propagation. Alg. 4 encompasses both Alg. 2 and 3, and performs the following steps.

(i) Iterative grid-search of the optimal hyperparameteres, accomplished only once per setting, at loop $C = 0$, with 5-fold cross-validation.
(ii) Model Training, accomplished at each loop $C \geq 0$, using the same optimal hyperparameters found at loop $C = 0$.
(iii) Model Evaluation, accomplished the four settings of interest.

These four settings of interest are the following.

(i) Evaluation on camera GT of the model trained with camera GT;
(ii) Evaluation on GT with flipped labels of the model trained on GT with flipped labels;
(iii) Evaluation on camera GT of the model trained on GT with flipped labels.
(iv) Evaluation of a Random Classifier on camera GT.

Algorithm 4 Model/Sensor Performance Estimation

Result: BIBO Performance distributions of RF and MLP models, evaluated separately for BLE and GPS features, over different average error rates

Input : features-from-pre-processed-dataset (FCD, see Alg. 2), true-labels (TL), target-signal (TS), hyperparameters-search-space (HSS, see Table I, II), maximum-error-rate (MERR)

Output: F1 (8), A (9), AUC, Optimal Hyperparameters (OP)

```
/* Simulate flipping labels and evaluate model performance against true
   ground-truth                                                          */
UULIST ← list-unique-users(FCD)
ERR ← 0.5
while ERR ≤ MERR do
    while C < 100 do
        /* Simlulate users errors and propagate through trajectory labels (see Alg.
           3)                                                            */
        FL ← simulate-and-propagate-error(UULIST, TL, FCD)
        /* Create Training- and Validation-set, compliant with OOS principle */
        VA ← pick-random-user-IDs(UULIST, users-num=2)
        VA ← extract-features-trajectories-by-user-ID-from-dataset(FCD,VA)
        TR ← extract-features-trajectories-by-user-ID-from-dataset(FCD,VA^C)
        /* Evaluate classifiers against true and flipped labels (TL Vs. FL) */
        foreach (TR,VA) ∈ {(TR,VA)_GPS, (TR,VA)_BLE} do
            foreach model ∈ {RF, MLP} do
                foreach label ∈ {FL, TL} do
                    L ← label
                    M ← model
                    if C=0 then
                        /* Hyperparameters 5-fold grid-search on training-set */
                        OP_L, (F1_{L_CV}, A_{L_CV}, AUC_{L_CV})_M ← model-5-fold-grid-search(TR,L)
                    else
                        /* Train a classifier with optimal hyperparameters and labels L */
                        classifier_L ← train-model(TR,L,OP_L)
                    end
                    /* Hold-out evaluation on true labels and validation-set, of a
                       classifier M trained with input-labels L          */
                    (F1_{L_HO}, A_{L_HO}, AUC_{L_HO})_M ← evaluate-model(classifier_L, VA, TL)
                    /* Hold-out evaluation on flipped labels and validation-set, of a
                       classifier M trained with input-labels L          */
                    (F1_{L_HO}, A_{L_HO}, AUC_{L_HO})_M ← evaluate-model(classifier_L, VA, FL)
                    /* Hold-out evaluation on labels L and validation-set, of Random
                       Classifier                                        */
                    (F1_{RL_HO}, A_{RL_HO}, AUC_{RL_HO})_M ← evaluate-model(random, VA, L)
                    (F1, A, AUC, OP).insert(F1, A, AUC, OP)_M
                end
            end
        end
        C ← C+1
    end
    ERR ← ERR+0.5
end
return ( OP, F1, A, AUC )
```

REFERENCES

[1] B. D. Hankin and R. A. Wright, "Passenger flow in subways," *J. Oper. Res. Soc.*, vol. 9, no. 2, pp. 81–88, Jun. 1958.

[2] J. Zhang et al., "A real-time passenger flow estimation and prediction method for urban bus transit systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 11, pp. 3168–3178, Nov. 2017.

[3] F.-Y. Wang, "Parallel control and management for intelligent transportation systems: Concepts, architectures, and applications," *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 3, pp. 630–638, Sep. 2010.

[4] J. Dekkers and P. Rietveld, "Electronic ticketing in public transport: A field study in a rural area," *J. Intell. Transp. Syst.*, vol. 11, no. 2, pp. 69–78, Apr. 2007, doi: 10.1080/15472450701293866.

[5] M. Mezghani. (2008). *Study on Electronic Ticketing in Public Transport*. [Online]. Available: https://emta.com/IMG/pdf/EMTA-Ticketing.pdf

[6] W. Narzt, S. Mayerhofer, O. Weichselbaum, S. Haselböck, and N. Höfler, "Be-in/be-out with Bluetooth low energy: Implicit ticketing for public transportation systems," in *Proc. IEEE 18th Int. Conf. Intell. Transp. Syst. (ITSC)*, Sep. 2015, pp. 1551–1556.

[7] B. V. Newzoo and F. Danzigerkade, *Global Mobile Market Report*. Amsterdam, The Netherlands: Noord-Holland, 2020. [Online]. Available: https://newzoo.com/resources/trend-reports/newzoo-global-mobile-market-report-2020-free-version

[8] V. Servizi, F. C. Pereira, M. K. Anderson, and O. A. Nielsen, "Transport behavior-mining from smartphones: A review," *Eur. Transp. Res. Rev.*, vol. 13, no. 1, p. 57, Dec. 2021, doi: 10.1186/s12544-021-00516-z.

[9] Y. Cui and S. S. Ge, "Autonomous vehicle positioning with GPS in urban canyon environments," *IEEE Trans. Robot. Autom.*, vol. 19, no. 1, pp. 15–25, Feb. 2003.

[10] P. Sapiezynski, A. Stopczynski, D. K. Wind, J. Leskovec, and S. Lehmann, "Inferring person-to-person proximity using WiFi signals," *Proc. ACM Interact., Mobile, Wearable Ubiquitous Technol.*, vol. 1, no. 2, pp. 1–20, Jun. 2017, doi: 10.1145/3090089.

[11] C. Li et al., "Enabling bus transit service quality co-monitoring through smartphone-based platform," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 2649, no. 1, pp. 42–51, Jan. 2017.

[12] D. Rolnick, A. Veit, S. Belongie, and N. Shavit, "Deep learning is robust to massive label noise," 2017, *arXiv:1705.10694*.

[13] S. Sukhbaatar, J. Bruna, M. Paluri, L. Bourdev, and R. Fergus, "Training convolutional networks with noisy labels," 2014, *arXiv:1406.2080*.

[14] A. Ahmed, H. Yousif, R. Kays, and Z. He, "Animal species classification using deep neural networks with noise labels," *Ecol. Informat.*, vol. 57, May 2020, Art. no. 101063.

[15] D. Hendrycks, M. Mazeika, D. Wilson, and K. Gimpel, "Using trusted data to train deep networks on labels corrupted by severe noise," 2018, *arXiv:1802.05300*.

[16] N. Natarajan, I. S. Dhillon, P. Ravikumar, and A. Tewari, "Learning with noisy labels," in *Proc. 26th Int. Conf. Neural Inf. Process. Syst.*, vol. 1, San Diego, CA, USA, 2016, pp. 1196–1204. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2013/file/3871bd64012152bfb53fdf04b401193f-Paper.pdf

[17] K. Yi and J. Wu, "Probabilistic end-to-end noise correction for learning with noisy labels," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 7010–7018.

[18] D. F. Nettleton, A. Orriols-Puig, and A. Fornells, "A study of the effect of different types of noise on the precision of supervised learning techniques," *Artif. Intell. Rev.*, vol. 33, no. 4, pp. 275–306, Apr. 2010.

[19] F. Rodrigues and F. Pereira, "Deep learning from crowds," in *Proc. AAAI Conf. Artif. Intell.*, 2017, p. 10.

[20] Y. Liu and H. Guo, "Peer loss functions: Learning from noisy labels without knowing noise rates," 2019, *arXiv:1910.03231*.

[21] L. Jiang, D. Huang, M. Liu, and W. Yang, "Beyond synthetic noise: Deep learning on controlled noisy labels," 2019, *arXiv:1911.09781*.

[22] N. Brouwers and M. Woehrle, "Dwelling in the canyons: Dwelling detection in urban environments using GPS, Wi-Fi, and geolocation," *Pervas. Mobile Comput.*, vol. 9, no. 5, pp. 665–680, Oct. 2013.

[23] K. Muthukrishnan, B. J. Van Der Zwaag, and P. Havinga, "Inferring motion and location using WLAN RSSI," in *Proc. Int. Workshop Mobile Entity Localization Tracking GPS-Less Environ.*, in Lecture Notes in Computer Science: Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics, vol. 5801, 2009, pp. 163–182.

[24] M. Y. Mun, D. Estrin, J. Burke, and M. Hansen, "Parsimonious mobility classification using GSM and WiFi traces," in *Proc. 5th Workshop Embedded Netw. Sensors*, 2011, pp. 1–5.

[25] P. A. Gonzalez et al., "Automating mode detection for travel behaviour analysis by using global positioning systems-enabled mobile phones and neural networks," *IET Intell. Transp. Syst.*, vol. 4, no. 1, pp. 37–49, 2010.

[26] D. K. Wind, P. Sapiezynski, M. A. Furman, and S. Lehmann, "Inferring stop-locations from WiFi," *PLoS ONE*, vol. 11, no. 2, Feb. 2016, Art. no. e0149105.

[27] A. Bjerre-Nielsen, K. Minor, P. Sapieżyński, S. Lehmann, and D. D. Lassen, "Inferring transportation mode from smartphone sensors: Evaluating the potential of Wi-Fi and Bluetooth," *PLoS ONE*, vol. 15, no. 7, Jul. 2020, Art. no. e0234003.

[28] G. Han, J. Jiang, C. Zhang, T. Q. Duong, M. Guizani, and G. K. Karagiannidis, "A survey on mobile anchor node assisted localization in wireless sensor networks," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 3, pp. 2220–2243, 3rd Quart., 2016.

[29] A. Yassin et al., "Recent advances in indoor localization: A survey on theoretical approaches and applications," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 2, pp. 1327–1346, 2nd Quart., 2017.

[30] A. Kotanen, M. Hännikäinen, H. Leppäkoski, and T. D. Hämäläinen, "Experiments on local positioning with Bluetooth," in *Proc. Int. Conf. Inf. Technol., Coding Comput. (ITCC)*, 2003, Art. no. 1197544.

[31] F. Subhan, H. Hasbullah, A. Rozyyev, and S. T. Bakhsh, "Indoor positioning in Bluetooth networks using fingerprinting and lateration approach," in *Proc. Int. Conf. Inf. Sci. Appl. (ICISA)*, 2011, Art. no. 5772436.

[32] L. Chen et al., "Constraint Kalman filter for indoor Bluetooth localization," in *Proc. 23rd Eur. Signal Process. Conf. (EUSIPCO)*, Aug. 2015, pp. 1915–1919.

[33] F. Subhan, H. Hasbullah, and K. Ashraf, "Kalman filter-based hybrid indoor position estimation technique in Bluetooth networks," *Int. J. Navigat. Observ.*, vol. 2013, Sep. 2013, Art. no. 570964.

[34] H. J. P. Iglesias, V. Barral, and C. J. Escudero, "Indoor person localization system through RSSI Bluetooth fingerprinting," in *Proc. 19th Int. Conf. Syst., Signals Image Process. (IWSSIP)*, Apr. 2012, pp. 40–43.

[35] A. Moubayed and A. Shami, "Softwarization, virtualization, and machine learning for intelligent and effective vehicle-to-everything communications," *IEEE Intell. Transp. Syst. Mag.*, vol. 14, no. 2, pp. 156–173, Mar./Apr. 2020.

[36] (2021). *Apple BLE Detection API*. [Online]. Available: https://developer.apple.com/library/archive/documentation/UserExperience/Conceptual/LocationAwarenessPG/RegionMonitoring/RegionMonitoring

[37] E. Beigman and B. B. Klebanov, "Learning with annotation noise," in *Proc. Joint Conf. 47th Annu. Meeting ACL 4th Int. Joint Conf. Natural Lang. Process. (AFNLP)*, vol. 1. Stroudsburg, PA, USA: Association for Computational Linguistics, 2009, pp. 280–287.

[38] N. Manwani and P. S. Sastry, "Noise tolerance under risk minimization," *IEEE Trans. Cybern.*, vol. 43, no. 3, pp. 1146–1151, Jun. 2013.

[39] C. M. Teng, "A comparison of noise handling techniques," in *Proc. FLAIRS Conf.* Washington, DC, USA: Association for the Advancement of Artificial Intelligence, May 2001, pp. 269–273. [Online]. Available: https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=bf99282770d4a176b84268595ad8200c8ae840fd

[40] R. Barandela and E. Gasca, "Decontamination of training samples for supervised pattern recognition methods," in *Advances in Pattern Recognition*, F. J. Ferri, J. M. Iñesta, A. Amin, and P. Pudil, Eds. Berlin, Germany: Springer, 2000, pp. 621–630.

[41] C. E. Brodley and M. A. Friedl, "Identifying mislabeled training data," *J. Artif. Intell. Res.*, vol. 11, pp. 131–167, Aug. 1999.

[42] A. L. Miranda, L. P. F. Garcia, A. C. Carvalho, and A. C. Lorena, "Use of classification algorithms in noise detection and elimination," in *Proc. Int. Conf. Hybrid Artif. Intell. Syst.*, in Lecture Notes in Computer Science: Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics, vol. 5572, 2009, pp. 417–424.

[43] B. Yuan, J. Chen, W. Zhang, H. Tai, and S. McMains, "Iterative cross learning on noisy labels," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2018, pp. 757–765.

[44] S. E. Reed, H. Lee, D. Anguelov, C. Szegedy, D. Erhan, and A. Rabinovich, "Training deep neural networks on noisy labels with bootstrapping," in *Proc. 3rd Int. Conf. Learn. Represent. (ICLR)*, Y. Bengio and Y. LeCun, Eds., San Diego, CA, USA, May 2015. [Online]. Available: http://arxiv.org/abs/1412.6596 and https://dblp.org/db/conf/iclr/iclr2015w.html

[45] I. Jindal, M. Nokleby, and X. Chen, "Learning deep networks from noisy labels with dropout regularization," in *Proc. IEEE 16th Int. Conf. Data Mining (ICDM)*, Dec. 2016, pp. 967–972.

[46] T. Xiao, T. Xia, Y. Yang, C. Huang, and X. Wang, "Learning from massive noisy labeled data for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 2691–2699.

[47] D. P. Kingma, S. Mohamed, D. J. Rezende, and M. Welling, "Semi-supervised learning with deep generative models," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 4, Jan. 2014, pp. 3581–3589.

[48] X. Zhu and Z. Ghahramani, "Learning from labeled and unlabeled data with label propagation," Dept. Mach. Learn., School Comput. Sci., Carnegie Mellon Univ. (CMU), Center Automated Learn. Discovery (CALD), Pittsburgh, PA, USA, Tech. Rep. CMU-CALD-02-107, 2002. [Online]. Available: https://mlg.eng.cam.ac.uk/zoubin/papers/CMU-CALD-02-107.pdf

[49] R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman, "Learning object categories from Internet image searches," *Proc. IEEE*, vol. 98, no. 8, pp. 1453–1466, Aug. 2010.

[50] (2021). *Android Mode Detection API*. [Online]. Available: https://developers.google.com/location-context/activity-recognition

[51] (2021). *Apple Mode Detection API*. [Online]. Available: https://developer.apple.com/documentation/coremotion/cmmotionactivity

[52] L. D. Riek, "Wizard of Oz studies in HRI," *J. Hum.-Robot Interact.*, vol. 1, no. 1, pp. 119–136, 2012.

[53] J. Nielsen and T. K. Landauer, "A mathematical model of the finding of usability problems," in *Proc. Conf. Hum. Factors Comput. Syst. (INTERACT)*, 1993, pp. 206–213.

[54] E. Cascetta and S. Nguyen, "A unified framework for estimating or updating origin/destination matrices from traffic counts," *Transp. Res. B, Methodol.*, vol. 22, no. 6, pp. 437–455, 1988.

[55] V. Servizi, N. C. Petersen, F. C. Pereira, and O. A. Nielsen, "Stop detection for smartphone-based travel surveys using geo-spatial context and artificial neural networks," *Transp. Res. C, Emerg. Technol.*, vol. 121, Dec. 2020, Art. no. 102834.

[56] S. Dabiri and K. Heaslip, "Inferring transportation modes from GPS trajectories using a convolutional neural network," *Transp. Res. C, Emerg. Technol.*, vol. 86, pp. 360–371, Jan. 2018, doi: 10.1016/j.trc.2017.11.021.

[57] X. Zhou, P. L. K. Ding, and B. Li, "Improving robustness of random forest under label noise," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2019, pp. 950–958.

[58] I. Tolstikhin, O. Bousquet, S. Gelly, and B. Schoelkopf, "Wasserstein auto-encoders," 2017, *arXiv:1711.01558*.

[59] L. Breiman, "Manual on setting up, using, and understanding random forests v4.1," Dept. Statist., Univ. California Berkeley, Berkeley, CA, USA, Tech. Rep. 4.0, pp. 3–42, 2002, vol. 1, no. 58. [Online]. Available: https://www.stat.berkeley.edu/~breiman/Using_random_forests_v4.0.pdf

[60] C. M. Bishop and N. M. Nasrabadi, *Pattern Recognition and Machine Learning*, vol. 4, no. 4. New York, NY, USA: Springer, p. 738.

[61] R. Zhang, P. Isola, and A. A. Efros, "Split-brain autoencoders: Unsupervised learning by cross-channel prediction," 2016, *arXiv:1611.09842*.

[62] T. C. Williams, C. C. Bach, N. B. Matthiesen, T. B. Henriksen, and L. Gagliardi, "Directed acyclic graphs: A tool for causal studies in paediatrics," *Pediatric Res.*, vol. 84, no. 4, pp. 487–493, Oct. 2018.

[63] R. Cipolla, Y. Gal, and A. Kendall, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7482–7491.

[64] A. Gretton, K. Borgwardt, M. J. Rasch, B. Scholkopf, and A. J. Smola, "A kernel method for the two-sample problem," 2008, *arXiv:0805.2368*.

[65] I. Higgins et al., "β-VAE: Learning basic visual concepts with a constrained variational framework," in *Proc. Int. Conf. Learn. Represent.*, 2017. [Online]. Available: https://openreview.net/forum?id=Sy2fzU9gl

[66] R. J. Campello, D. Moulavi, and J. Sander, "Density-based clustering based on hierarchical density estimates," in *Proc. 17th Pacific-Asia Conf. Adv. Knowl. Discovery Data Mining (PAKDD)*, Gold Coast, QLD, Australia. Berlin, Germany: Springer, Apr. 2013, pp. 160–172.

[67] H. Tim, "New perspectives on the performance of machine learning classifiers for mode choice prediction," Transp. Mobility Lab. School Archit., Civil Environ. Eng. Ecole Polytechnique Fédérale de Lausanne, Switzerland, Tech. Rep. TRANSP-OR 200704, 2020. [Online]. Available: https://transp-or.epfl.ch/documents/technicalReports/HillelNew2020.pdf

[68] A. Baraldi, L. Bruzzone, and P. Blonda, "Quality assessment of classification and cluster maps without ground truth knowledge," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 4, pp. 857–872, Apr. 2005.

[69] C. H. Lubba, S. S. Sethi, P. Knaute, S. R. Schultz, B. D. Fulcher, and N. S. Jones, "catch22: Canonical time-series characteristics," *Data Mining Knowl. Discovery*, vol. 33, no. 6, pp. 1821–1852, Nov. 2019.

[70] G. Cantarero and R. Jarabo, "The area under the ROC curve," *Medicina Clinica*, vol. 106, no. 9, pp. 355–356, 1996.

[71] L. Shu, Y. Su, W. Jiang, and K.-L. Tsui, "A comparison of exponentially weighted moving average-based methods for monitoring increases in incidence rate with varying population size," *IIE Trans.*, vol. 46, no. 8, pp. 798–812, Aug. 2014, doi: 10.1080/0740817X.2014.894805.

[72] J. Paek, J. Ko, and H. Shin, "A measurement study of BLE iBeacon and geometric adjustment scheme for indoor location-based mobile applications," *Mobile Inf. Syst.*, vol. 2016, Oct. 2016, Art. no. 8367638, doi: 10.1155/2016/8367638.

[73] I. Malmberg, "An analysis of iBeacons and critical minimum distances in device placement," Ph.D. dissertation, School Inf. Commun. Technol. (ICT), KTH Royal Inst. Technol., Stockholm, Sweden, 2014. [Online]. Available: http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-187925

Valentino Servizi received the Ph.D. degree from the Technical University of Denmark (DTU), Machine Learning for Smart Mobility Research Group (MLSM). His Ph.D. research focused on smartphone-based travel surveys, machine learning methods for classification of time-series data generated by smartphone devices, and the Internet of Things.

Dan Roland Persson received the B.Sc. degree in software engineering and the M.Sc. degree in digital media engineering. He is currently pursuing the Ph.D. degree with the Department of Applied Mathematics and Computer Science-Cognitive Systems, Technical University of Denmark (DTU). He was with LINC Project, as a Research Assistant and a Full-Stack Developer. His research interests include user experience, context aware technologies, and micro-interventions.

Francisco Camara Pereira is currently a Professor with the Technical University of Denmark (DTU), where he leads the Machine Learning for Smart Mobility Group (MLSM). He has published more than 50 articles in both machine learning and transport research fields. His research interests include the methodological combination of machine learning and transport research, and some applications include demand modeling, traffic prediction, data collection, or anomaly detection. He is a Marie Curie Fellow.

Hannah Villadsen is currently a Post-Doctoral Researcher with Roskilde University, where she is with the Sustainable Transition Research Group within the Department of People and Technology. With a background in behavioral psychology her research and teaching focus is in the field of cognitive, behavioral and social aspects of sustainable transport and implementation of mobility technologies.

Per Bækgaard received the M.Sc. degree from Technical University of Denmark (DTU). He is currently an Associate Professor of cognitive systems section with the Department of Applied Mathematics and Computer Science (DTU Compute), DTU, and heads the Human-Centered Artificial Intelligence. His research interests include user experience and human–computer interaction and in applying machine learning and statistical approaches to large and sparse data arising from human activity.

Inon Peled received the Ph.D. degree from the Machine Learning for Smart Mobility Research Group, Technical University of Denmark (DTU), in 2021. He is currently a Data Scientist and a Former Post-Doctoral Researcher with the Machine Learning for Smart Mobility Research Group, Technical University of Denmark (DTU). His research interests include predictive modeling for abnormal conditions in the transport domain.

Otto Anker Nielsen is currently a Professor with the Technical University of Denmark (DTU), where he leads the Transport Division. His research interests include experience and applied work experience in the field of transport modeling and transport behavior research, includes research in all modes of transport, passenger as well as freight transport, and various scales of modeling from local transport to international and intercontinental models.