

AndroCon: Conning Location Services in Android

Soham Nag

Center of Excellence in Cyber Systems
and Information Assurance

Indian Institute of Technology Delhi

New Delhi, India

jcs222663@csia.iitd.ac.in

Smruti R. Sarangi

Computer Science and Engineering

Indian Institute of Technology Delhi

New Delhi, India

srsarangi@cse.iitd.ac.in

Abstract—Ambient sensing, human activity recognition and indoor floor mapping are common targets for attackers who hack mobile devices. Other than overt signals such as microphones and cameras, other covert channels such as WiFi, Bluetooth and assisted GPS signal strengths have been used to infer this information. In the space of passive, receive-only satellite GPS-based sensing, the state-of-the-art was limited to using only the signal strength and location information up till now. This paper shows that semi-processed GPS data (with 39 features) that is now accessible to apps since the release of Android 7 (with precise location permissions) can be used as a strong leaky channel to sense the ambient, recognize human activity and map indoor spaces with a very high accuracy (99%+ in many cases). This paper presents the results of a longitudinal study where semi-processed GPS measurements were taken over the course of a year using different mobile devices spread out over a 40,000 sq. km geographical region. Data was also collected on flights, cruise ships and high-altitude locations. We thoroughly characterize all the satellite GPS signals and based on cross-correlation analysis extract the best set of features that preserve vital information.

We propose a novel method, AndroCon, that comprises linear discriminant analysis, unscented Kalman filtering, gradient boosting and random forest based learning to yield a highly accurate ambient and human activity sensor. AndroCon relies on simple ML algorithms that can run surreptitiously and provide partially explainable results. We easily identify difficult scenarios such as being inside a metro, a person waving a hand in front of the mobile device, being in front of a staircase and the presence of people in the room (not necessarily holding mobile phones). This is the most comprehensive study on satellite GPS-based sensing till date.

I. INTRODUCTION

The integration of advanced wireless technologies such as WiFi, 4G, and 5G along with precise-positioning technologies such as the Global Positioning System (GPS) in mobile devices has fuelled the growth of a plethora of applications that use location-based services (LBS) [1], [2]. The market for such applications is currently 50 billion USD and is anticipated to soar to 400 billion USD by 2030 (CAGR of 24.6%) [3], [4].

LBS services are used by applications that provide the following services: navigation, local search, traffic alerts, weather updates, home delivery, ride-sharing and device tracking – they enhance the user experience significantly. As of today, such applications have become indispensable. 94% of smartphone users rely on them daily and 84% of small to medium-sized

businesses are reported to have seen a surge in the footfall due to location-based marketing [5].

We shall show in this paper that while LBSs offer a lot of convenience, they pose significant privacy risks – many of them were hitherto unknown and undiscovered. Let us first look at the known privacy risks of the *precise location signal*. It conveys location information, which for obvious reasons leaks privacy information [6]. Our claim in this paper is that GPS signals that are used to find the location can be used for other surreptitious activities as well: sensing the ambient, tracking human activity, and finding more about the layout (floorplan). This can be done because modern GPS chips share a lot of information with applications such as the signal strength, Doppler shift, signal-to-noise ratio (SNR), etc. Modules in the GPS chip use this information to deduce the precise location (x , y , and z coordinates) of the device. Our claim in this paper is that we can find alternative uses for this information that is readily available at the application level. This will allow us to sense the user’s environment. Using such semi-processed GPS signals – already provided by all GPS chips to Android applications – as an ambient sensor is entirely novel (to the best of our knowledge).

There is some work in the field of ambient sensing using other modalities. For example, smart speakers [7] and WiFi signals [8], [9] have been used to find out details of the users’ ambience. In the former case, background sounds in the environment have been used to guess the nature of the user’s surroundings and in the latter case, the WiFi signal strengths have been used as an information-leaking channel. Similar work exists with other more overt channels such as mobile phone cameras [10]–[12]. Bluetooth beacons emit signals that can be picked up by nearby devices. This allows attackers to infer proximity and movement patterns of users [13]. Such ambient information is quite beneficial. One of the innocuous uses is that it allows third parties to influence online shopping behavior through targeted advertising [14]. However, there are more pernicious applications as well that pose genuine privacy risks such as figuring out that a person of interest is having a meeting in a small room, when he was actually supposed to be there at a party 10 meters away. The basic cell phone location information will not be able to discriminate between these two situations, whereas our solution *AndroCon* will be

able to.

Let us further elaborate on our use case. Assume that somehow a user was conned into installing an Android app. She further gave it precise location permissions. Almost all e-commerce, ride-sharing apps and some games like Pokemon Go [15] necessitate such permissions. Now, our claim is that even if the device is in flight mode and all communication channels are off (WiFi, mobile data, NFC, Bluetooth), if we can still continue to read semi-processed GPS data, then an important information-leaking side-channel exists. The GPS data can be used to figure out important information about the environment that includes (but is not limited to) whether the user is in a closed space or open space, is the place crowded, is the user sitting or standing, moving quickly or slowly, underground or overground, inside a flight, within a lift or close to a staircase (refer to Section VI for the full list). There is no need to take the help of any other kind of sensor such as the camera, accelerometer or microphone. Our claim is that semi-processed GPS data (in pure receive-only mode) is sufficient. In this paper, we shall advance various arguments to convince the reader that this is indeed possible. We shall make theoretical arguments, show the results of simulations, show real measurement data using GPS sensors and finally show extensive evaluation results using 5 phones collected over a period of 1 year.

In this paper, we shall demonstrate the potential of semi-processed GPS parameters, both independently and collectively, to discern user activity (both static and dynamic) and the ambience in diverse settings (crowded, open area, indoor, metro tunnel, flight). As GPS signals bounce off the user and their surroundings, they capture the environmental context, generating a unique pattern for each setting [16]–[18]. These details are intrinsic to semi-processed GPS parameters, which we fingerprint for the purpose of ambient sensing. However, GPS signals, are inherently noisy due to multipath effects and interference with other signals [19] – this necessitates data cleaning prior to the fingerprinting exercise. We employ a nonlinear Kalman filter for noise filtering. It preserves the essential signal variations that our algorithm needs [20]. Subsequently, we apply Linear Discriminant Analysis (LDA) for feature reduction and identify linear combinations of semi-processed parameters that effectively distinguish between different ambient contexts. The resulting data is then fed into an ML model for ambient and activity classification. Additionally, we use the semi-processed parameters and GPS signal strength, in conjunction with user trajectories and graph optimization (GO) techniques, to detect the indoor layout of the user’s location.

This paper makes the following contributions:

- ① To the best of our knowledge, this is the first study that characterizes in-depth the potential of semi-processed GPS signal for ambient sensing.
- ② We show how apps using GPS sensing can covertly capture semi-processed GPS data without consent and utilize it to discern the user’s ambience (99.6%) and activities (87%) across *diverse* settings, effectively jeopardizing privacy. At the

moment, this vulnerability affects 90% of Android users.

③ Evaluation of ML models to effectively classify different user activities and ambient settings using 8 semi-processed GPS parameters as inputs.

④ The ability of attackers to infer floor maps/indoor layouts (error margin of 4 meters), using semi-processed GPS data and user trajectories, without needing access to other ambient information leaking sources such as cameras.

The structure of this paper is as follows: section II provides the necessary background information. In section III, we discuss the semi-processed GPS parameters that we use and their effectiveness in classifying activities and environments followed by characterization and validation of semi-processed data readings in section IV. The architecture of the model is shown in section V and its performance is evaluated in section VI. The overview and evaluation of the layout estimation algorithm are presented in section VII. We review the relevant literature in section IX, and finally, conclude in section X.

II. BACKGROUND

A. Overview of GPS

The Global Positioning System (GPS) was established by the US Department of Defense in 1973. It is a member of the Global Navigation Satellite System (GNSS) constellation. GPS, in 2024, has 31 satellites that orbit the Earth in six planes of rotation. All of them are in the medium earth orbit (20,000 kms above sea level [21]). These satellites transmit navigation signals, which contain precise information about their position, velocity and current time (PVT). This information is generally resistant to inclement weather conditions. To accurately calculate the geographical location, a GPS receiver requires signals from at least four satellites.

Let us explain the physics. Every GPS satellite has an atomic clock that very accurately maintains the time. The uncertainty is less than 1 part in 10^{16} . Assume that a message is sent from a satellite at time t_s and it is received at time t_r . Then the distance d between the sending satellite and the receiver is shown in Equation 1.

$$d = (t_s - t_r + \Delta) \times c \quad (1)$$

Here, c is the speed of light and Δ is the clock skew between the non-ideal receiver clock and the ideal sender clock. Now, the Euclidean distance d is also give by another equation (see Equation 2).

$$d = \sqrt{(x_s - x_r)^2 + (y_s - y_r)^2 + (z_s - z_r)^2} \quad (2)$$

Here, $\langle x_s, y_s, z_s \rangle$ are the coordinates of the satellite and $\langle x_r, y_r, z_r \rangle$ are the coordinates of the receiver. There are four unknowns here: three of them are the coordinates of the receiver and the fourth unknown is the clock skew Δ . For four unknowns, we need four equations. We thus need data from four satellites.

We need to note that Equation 1 is an ideal relationship. There are delays induced due to the ionosphere, signal interference, multipath effects, etc. Hence, modern receivers need to

apply many different correction factors. The actual relationship is thus more complex. Hence, there is a need to also transmit many other parameters (described in Section III) as well such that receivers can correct their data. These include the pseudorange along with SNR, phase shifts, Doppler shifts and details about the satellite and the constellation (see Table I). A GPS message also contains other parameters about the receiver's clock (other than the time itself).

1) *GPS pipeline*: Figure 1 shows the GPS processing pipeline at the receiver. Each GPS satellite uses a unique pseudorandom code (*PRN*) to modulate its signal for unique identification, ensuring security and enhancing the robustness [22]. Code and carrier tracking loops monitor all such incoming GPS signals, demodulate and identify them. The code tracking loop compares the incoming signal's code with the codes of one of the replicas. Through cross-correlation analysis, the receiver finds the code (and satellite) corresponding to the signal. This aids in calculating the pseudorange – apparent distance between the satellite and receiver [22] – which is one of the most important parameters of interest for distance calculation.

Modern GPS chipsets allow an application to tap the information at any point in the GPS processing pipeline. In general, applications do not tap the information available at the intermediate stages because they don't find it useful. Only the output of the final stage is used by applications – it is the device's precise location information. We decided to tap the pipeline in the middle and extract 9 parameters of interest out of 32 parameters (refer to Table I). Hence, we refer to these parameters as *semi-processed GPS data*.

A brief taxonomy of these parameters is presented in Figure 2. We broadly use five types of parameters that can be classified into the following buckets: received signal power, carrier phase shift, multipath interference, signal-to-noise ratio (SNR) and the Doppler shift (due to satellite motion). These parameters can be used to extract information about the surroundings because they are influenced by it. For instance, if there are a lot of objects around the receiver, the multi-path interference and SNR will be high.

B. Location Accuracy Permissions in Android

In Android, the location accuracy permission, facilitated by `ACCESS_COARSE_LOCATION` and `ACCESS_FINE_LOCATION`, enables apps to specify the desired precision to access the location of the device. `ACCESS_COARSE_LOCATION` utilizes cell tower or Wi-Fi network data for positioning, offering 100 meters accuracy [23], suitable for weather or news apps. `ACCESS_FINE_LOCATION` provides higher precision through GPS-based positioning methods, typically within 50 meters, occasionally as fine as 3 meters or better [24], crucial for mapping or navigation apps. In Android 12 and beyond, apps must request both permissions, irrespective of their precision requirements, to suit user preferences. Notably, accessing semi-processed GPS measurements requires `ACCESS_FINE_LOCATION` permission.

C. Linear Discriminant Analysis (LDA)

Any kind of learning that involves a lot of parameters tends to be inefficient unless we have a large number of training examples. Hence, dimensionality reduction needs to be done. We use LDA, which is a widely used supervised-learning technique for dimensionality reduction in pattern-classification applications [25]–[27]. It projects a dataset into a lower-dimensional space that enhances *class separability* through a linear combination of features. This is done to address the curse of dimensionality and reduce computational costs [28]. Unlike a cognate technique namely Principal Component Analysis (PCA), which identifies directions of maximum variance, LDA creates a transformation that maximizes the separation between different classes.

LDA operates by finding a projection matrix that transforms the original data into a new space where samples from different classes are *more distinct*. This is achieved by maximizing the between-class variance while minimizing the within-class variance. The projection matrix is determined through an optimization process involving scatter matrices that represent variances. This involves solving an eigenvalue problem, where the eigenvectors form the projection matrix. By transforming the data in this manner, LDA facilitates a more effective and efficient classification, making it a valuable tool for the exploitation of the semi-processed GPS data in our research.

D. Unscented Kalman Filter

The Unscented Kalman Filter (UKF) is an advanced unsupervised technique used for noise filtering in *nonlinear* systems. It predicts the future state of a nonlinear system and updates this prediction using incoming noisy measurements. Unlike the traditional Kalman Filter, which assumes linearity, the UKF employs a deterministic sampling approach to generate sample points around the mean, which are then passed as inputs to nonlinear functions. This method accurately captures the mean and covariance accurately, enhancing its effectiveness in complex applications. The UKF is widely used in fields such as robotics, and navigation, where precise estimation in the presence of noise is crucial.

Consider a nonlinear state-space model at time t :

$$x_t = f(x_{t-1}) + w_t \quad (3)$$

$$z_t = h(x_t) + v_t \quad (4)$$

x_t is the state vector and z_t is the corresponding noisy measurement at time t ; $f(\cdot)$ is the state transition function; $h(\cdot)$ is the observation function; $w_t \sim \mathcal{N}(0, Q)$ is the process noise representing the uncertainty in the model's state transition with covariance Q ; $v_t \sim \mathcal{N}(0, R)$ is the measurement noise representing the uncertainty in the measurements with covariance R .

The Unscented Transform approximates the state distribution using a set of sigma points. These points are passed as inputs to the the nonlinear functions f and h .

1) Sigma Point Generation

Sigma points $\chi_{t-1}^{(i)}$ are generated around the mean \hat{x}_{t-1} :

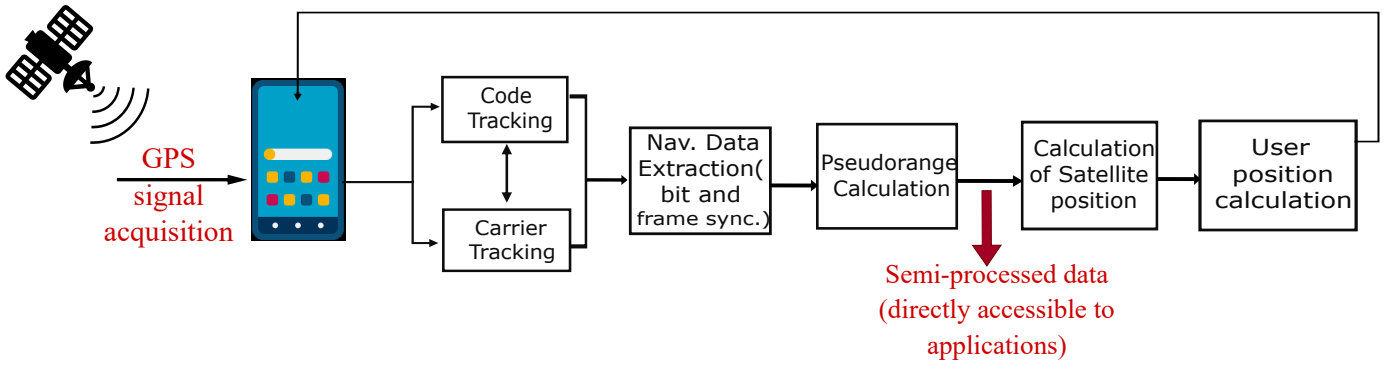


Fig. 1: GPS pipeline: From signal acquisition to position calculation

TABLE I: List of semi-processed GPS signal parameters

Field	Notations	Description	Unit
<i>PseudorangeRate</i>	PR	Pseudorange rate at the associated timestamp.	m/s
<i>PseudorangeRateUncertainty</i>	PRU	Pseudorange's rate uncertainty ($1-\sigma$).	m/s
<i>ReceivedSvTimeUncertainty</i>	RecSvTU	Error estimate ($1-\sigma$) for the received GNSS time.	ns
<i>AccumulatedDeltaRange</i>	ADRng	Accumulated delta range since the last channel reset.	m
<i>AccumulatedDeltaRangeUncertainty</i>	ADRngU	Uncertainty of the accumulated delta range ($1-\sigma$).	m
CN0	CN0	Carrier-to-noise density in the range [0,63].	dB-Hz
<i>BasebandCn0DbHz</i>	BbCN0	Baseband carrier-to-noise density (Added in API level 30).	dB-Hz
<i>AgcDb</i>	Agc	Incoming signal power.	dB
<i>State</i>	State	Integer representing satellite sync state, with each bit indicating a specific measurement status.	-

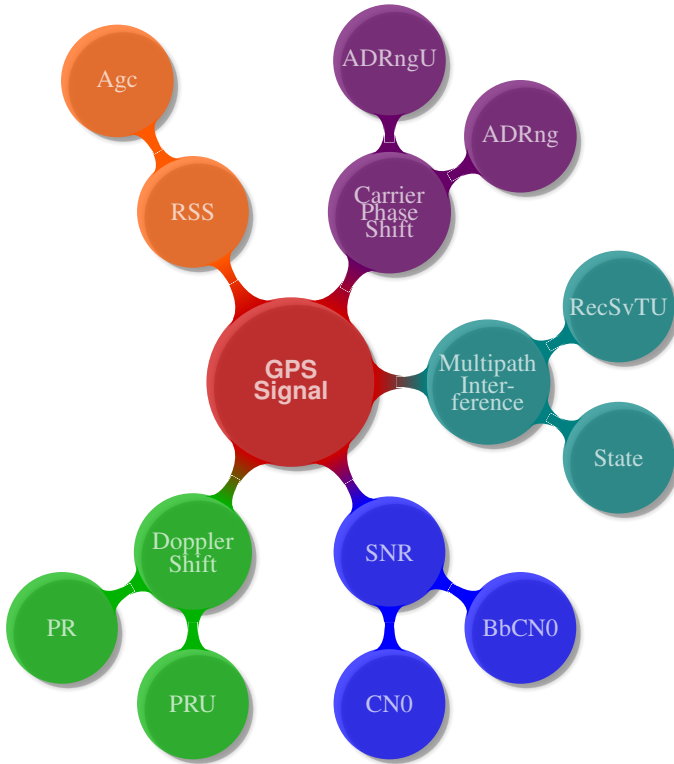


Fig. 2: Representation of the semi-processed GPS signal parameters that we use. RSS is “Received Signal Strength”

$$\chi_{t-1}^{(0)} = \hat{x}_{t-1} \quad (5)$$

$$\chi_{t-1}^{(i)} = \hat{x}_{t-1} + \sqrt{(L + \lambda)P_{t-1}}, \quad i = 1, \dots, L \quad (6)$$

$$\chi_{t-1}^{(i+L)} = \hat{x}_{t-1} - \sqrt{(L + \lambda)P_{t-1}}, \quad i = 1, \dots, L \quad (7)$$

P_{t-1} is the error covariance matrix; L is the dimension of the state vector, and λ is a scaling parameter describing the spread of the sigma points around \hat{x}_{t-1} . Note that in the first case we are adding a scaled version of the error, and in the second case we are subtract it.

② Prediction Step

The sigma points are next passed to the state transition function to generate a new set of samples.

$$\chi_{t|t-1}^{(i)} = f(\chi_{t-1}^{(i)}) \quad (i = 0, 1, 2, \dots, 2L) \quad (8)$$

The predicted mean and covariance are computed as:

$$\hat{x}_{t|t-1} = \sum_{i=0}^{2L} W_m^{(i)} \chi_{t|t-1}^{(i)} \quad (9)$$

$$P_{t|t-1} = \sum_{i=0}^{2L} W_c^{(i)} \left[\chi_{t|t-1}^{(i)} - \hat{x}_{t|t-1} \right] \times \left[\chi_{t|t-1}^{(i)} - \hat{x}_{t|t-1} \right]^T + Q \quad (10)$$

$$\begin{cases} W_m^{(i)} = 1 - \frac{1}{\lambda^2} i = 0 \\ W_c^{(i)} = \frac{1}{2L\lambda^2} i = 1, 2, \dots, 2L \end{cases}$$

$W_m^{(i)}$ and $W_c^{(i)}$ are the weights for the mean and covariance, respectively.

② Update Step

The predicted measurements are obtained by passing the sigma points to the observation function (h).

$$\gamma_{t|t-1}^{(i)} = h(\chi_{t|t-1}^{(i)}) \quad (11)$$

The predicted measurement's mean and covariance are as follows:

$$\hat{z}_{t|t-1} = \sum_{i=0}^{2L} W_m^{(i)} \gamma_{t|t-1}^{(i)} \quad (12)$$

$$S_t = \sum_{i=0}^{2L} W_c^{(i)} \left[\gamma_{t|t-1}^{(i)} - \hat{z}_{t|t-1} \right] \left[\gamma_{t|t-1}^{(i)} - \hat{z}_{t|t-1} \right]^T + R \quad (13)$$

$$C_t = \sum_{i=0}^{2L} W_c^{(i)} \left[\chi_{t|t-1}^{(i)} - \hat{x}_{t|t-1} \right] \left[\gamma_{t|t-1}^{(i)} - \hat{z}_{t|t-1} \right]^T \quad (14)$$

C_t is the cross-covariance matrix that represents the covariance between the state and the sigma points. S_t is the residual covariance matrix that includes the measurement noise covariance. The Kalman gain K_t and the updated state and covariance are as follows:

$$K_t = C_t S_t^{-1} \quad (15)$$

$$\hat{x}_t = \hat{x}_{t|t-1} + K_t(z_t - \hat{z}_{t|t-1}) \quad (16)$$

$$P_t = P_{t|t-1} - K_t S_t K_t^T \quad (17)$$

We keep repeating this process from the sigma point generation step for the subsequent inputs until all the data is processed.

III. THE VALUE OF SEMI-PROCESSED GPS DATA

In this section, we begin by examining the way semi-processed GPS data is retrieved on an Android platform. Subsequently, we explain all the 9 GPS parameters, and finally explain how ambient sensing can be achieved.

A. Accessing Semi-Processed GPS Measurements

Prior to Android 7, developers accessed location details via the android.gsm.location API, which provided basic satellite information like C/N0 (carrier-to-noise ratio), azimuth and elevation, along with the fundamental National Marine Electronics Association (NMEA) sentences containing the PVT (position, velocity and time) solution [29]. Multipath effects and potential interference greatly degrade the positioning accuracy by altering the true distance between satellites and the user; this causes pseudorange errors. GPS receivers attempt to correct these errors, but since these measurements aren't directly accessible to users, they rely on often inadequate receiver-embedded correction models [30]. To foster the development of improved correction models for higher accuracy, Google has made these details public on Android phones. This information comprises the semi-processed GPS data, which is available to developers via the android.location.GnssMeasurements API within the android.location package (starting from Android 7 [31]).

B. Information Present in Semi-Processed GPS Measurements

Even though similar problems have been solved in the case of Wi-Fi and Bluetooth signals, the same techniques cannot be reused. We need to devise novel methods to select features, identify the nature of the noise, clean the data, reduce the dimensionality, choose the settings and ML models of interest

and comprehensively evaluate the design. Let us understand in more detail the 9 GPS parameters of interest. The rest 23 (out of 32) parameters were eliminated because they were not recorded on all the phones (variability across chipsets), the values were quite unstable or were of no use (such as the details of the satellite).

① Doppler Shift :

The Doppler shift indicates a change in frequency due to the relative motion between the signal source and the receiver. Given that the satellites are moving and the receiver itself may be non-stationary, a Doppler shift is expected. A positive shift indicates that the satellite is moving towards the receiver – this increases the perceived signal frequency. Conversely, a negative shift occurs as the satellite moves away, effectively decreasing the frequency.

$$PR = -k \times dopplershift \text{ (where } k \text{ is a constant)} \quad (18)$$

This metric is significant in discerning the following scenarios.

Identifying Motion State: A high value of PR suggests movement, while a low or near-zero value indicates a stationary state or slow movement.

Distinguishing Environmental Context: Multipath phenomena, where signals reflect off surfaces before reaching the receiver affect Doppler shifts. In open areas with minimal multipath effects, the PR measurements remain stable, while in crowded or indoor areas, they can fluctuate significantly due to multipath effects.

Similarly, PRU quantifies the uncertainty in PR measurements, which varies with environmental conditions. In crowded or mobile environments, multipath effects and irregular Doppler measurements increase this uncertainty, resulting in high PRU values. In open areas with minimal multipath effects, the uncertainty is lower, leading to low PRU values. This analysis underscores the utility of PR and PRU in inferring both the state of motion and the environmental context of individuals based on semi-processed GPS measurements.

② Carrier Phase:

This quantifies the accumulated error in the distance between the satellite and the receiver. It is calculated by measuring the phase shift. A positive value indicates that the satellite vehicle (SV) is moving away from the receiver, while a negative value signifies that the SV is moving towards the receiver.

$$ADRN_g = -k * carrierphase \text{ (where } k \text{ is a constant)} \quad (19)$$

This facilitates the differentiation between the following scenarios.

Identifying Motion State: A high change in the ADRng suggests motion, while a low or near-zero change indicates stationary or slow movement.

Distinguishing Environmental Context: In open areas with minimal multipath interference, ADRng values are stable. Conversely, in crowded or indoor environments, reflections cause phase shifts, altering the carrier phase and resulting in variable ADRng measurements.

Similarly to PRU, ADRngU reflects the uncertainty in ADRng measurements.

④ Signal To Noise Ratio:

CN0, or Carrier-to-Noise (C/N) density in dB-Hz, is a crucial metric for evaluating GNSS signal quality. It typically ranges from 10 to 50 dB-Hz with a potential spectrum extending from 0 to 63 dB-Hz in exceptional circumstances [32]. Variances in CN0 allow differentiation between the following environmental conditions.

In congested or enclosed areas with signal interference, CN0 levels decrease, indicating heightened noise in the GPS signal and suggesting the individual's presence in such environments. Conversely, open-sky scenarios with minimal interference increase CN0 levels, signifying reduced noise in the GPS signal and implying the individual's presence in open-sky or less congested areas.

Similarly, BbCN0 represents the C/N of the signal at the baseband, obtained after demodulating the GPS signal received by the antenna. This value is typically slightly weaker than the C/N measured at the antenna port (CN0) [32]. Our experiments demonstrated a strong correlation coefficient(0.98) between CN0 and BbCN0, indicating similar behavior in distinguishing environments/activities.

④ Received Signal Strength (RSS):

The Agc acts as a dynamic amplifier, adjusting the power of incoming signals. Negative or low Agc value indicates potential interference or jamming [32]. Notably, this value remains consistent under the same level of incoming signal power, helping differentiate user activity (motion/rest).

Furthermore, Agc can distinguish between different ambient conditions, such as crowded urban areas or open spaces near electromagnetic transmission towers. In highly interfered environments like crowded areas, Agc levels are low, while in less congested or minimally interfered spaces, Agc values are higher.

⑤ Multipath Interference:

The *State* field indicates the current synchronization state for each satellite signal. It can assume a value of either 0 or a combination of different synchronization states [32].

When the *State* is STATE_MSEC_AMBIGUOUS (value: 16), it implies millisecond-level ambiguity in the GPS measurement's tracking state due to multipath effects [32], indicating a congested or enclosed environment.

RecSvTU represents the error estimate ($1-\sigma$) for the received GPS time of a particular *SV*, typically influenced by multipath interference. As it quantifies uncertainty, its effectiveness in distinguishing user activity and environmental conditions mirrors that of PRU And ADRngU.

IV. EMPIRICAL STUDY OF THE SEMI-PROCESSED GPS SIGNAL CHARACTERISTICS

This section aims to establish a mapping between EM wave parameters, such as RSS and Doppler shift, which are utilized for ambient sensing, and the corresponding semi-processed GPS parameters. Subsequently, it examines the correlation

among semi-processed parameters to reinforce the idea that they can be *fused* to achieve improved classification outcomes.

A. Real-World Data Collection

Semi-processed GPS data was logged using the GnssLogger [33] Android application on five different Android phones (refer to Table II). Note that phones that use the Samsung Snapdragon chipset do not allow the user to retrieve the ADRng measurements [34]. We deliberately used two phones with the same version (Redmi Note 9 Pro Max) to characterize variations across two phones of the same model. There was some variation even in this case. Therefore, noise filtering is required to ensure that our technique works across phones in a robust manner.

TABLE II: Details of the phones

Model Name	Android version	Chipset	Year
Redmi Note 9 Pro Max	Android 11	Snapdragon 720G SoC	2020
Redmi Note 9 Pro Max	Android 11	Snapdragon 720G SoC	2021
Galaxy A54	Android 14	Exynos 1380	2023
OnePlus Nord CE2	Android 13	MediaTek Dimensity 900	2022
Redmi Note K20 Pro	Android 12	Snapdragon 855	2020

B. Datasets: Synthetic (Kaggle) and Real-World

Along with the data that we collect using these phones, we use open-source datasets as well. This is needed to ensure that similar effects are also being seen in those datasets and our noise filtering techniques work. We utilized the Kaggle GNSS dataset [35]; it contains 39 traces collected using the Pixel 4, Pixel 4 XL and Xiaomi Mi8 phones (resp.). This dataset primarily contains measurements collected during motion. To simulate diverse user activities and environments, we generated supplementary data at different sites using our 5 phones. The settings include an open ground, indoors, while standing and sitting. The data collection process lasted approximately 45-150 minutes at each site on an average.

The ionosphere's electron density variations cause fluctuations in GPS signals [36], resulting in measurement variability. Furthermore, diverse weather conditions can introduce additional fluctuations. This leads to erroneous measurements and potential misclassification. To simulate real-world settings and mitigate these effects, semi-processed data was collected three times over the course of a year under various atmospheric conditions and at different times of the day. We tried to minimize the experimental noise by ensuring the ambient setting was the same. Of course, there were variations due to the weather, position of satellites, etc. We did not try to achieve consistency in these parameters.

C. Calibration with Real Sensors: Signal Power

Researchers have primarily used the RSS, the Doppler shift and SNR for designing human activity recognition (HAR) and ambient sensing systems for WiFi, Bluetooth and cellular tower signals [16]–[18]. We, of course, use many more parameters.

However, we stick to these three parameters for the purpose of characterization and calibration of our setup. We map these parameters to the corresponding semi-processed GPS parameters. The GPS signal inherently captures the Agc (RSS or signal power). We used an RF Explorer Spectrum Analyzer (Model B34J7ML7J58J9MD6) [37], which records the signal RSS. We set the RF Explorer to the L1 GPS band frequency range (1575.42 MHz) [38]. We logged the GPS's RSS using the measurement device (RF Explorer) and simultaneously, we logged the Agc. We found a linear relationship between them (refer to Figure 3).

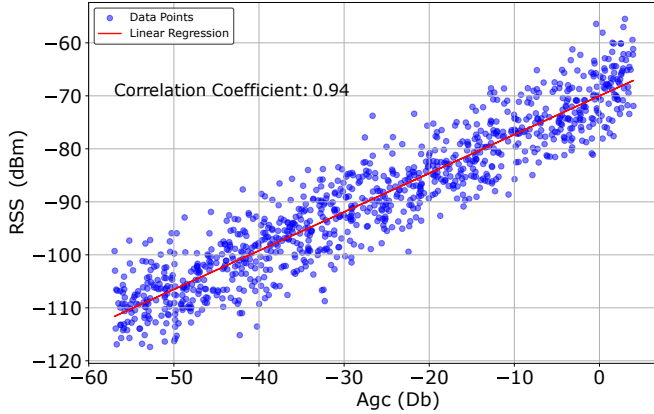


Fig. 3: Correlation between Agc and RSS

D. Correlation Results across the GPS Parameters: Feature Selection and LDA

Next, we consider all our data: our traces and the Kaggle GNSS traces. The trends are roughly similar. We plot the correlation for all pairs of GPS signals (results shown in Figure 4). CN0 exhibits a strong correlation with BbCN0 (0.98). Hence, only one of them is required. We choose CN0. The correlation of CN0 with other signals is as follows: ADRngU (0.68), PRU (0.48), and RecSvTU (-0.53). PR shows a correlation of 0.63 and 0.54 with ADRng and RecSvTU, respectively. This means that we have a lot of pairs of parameters that have reasonably high correlations. This motivates the use of a feature reduction algorithm such as LDA.

V. MODEL ARCHITECTURE

We shall use the same ML model to classify both the ambience as well as human activity. Henceforth, we will refer to both an ambient setting as well as human activity

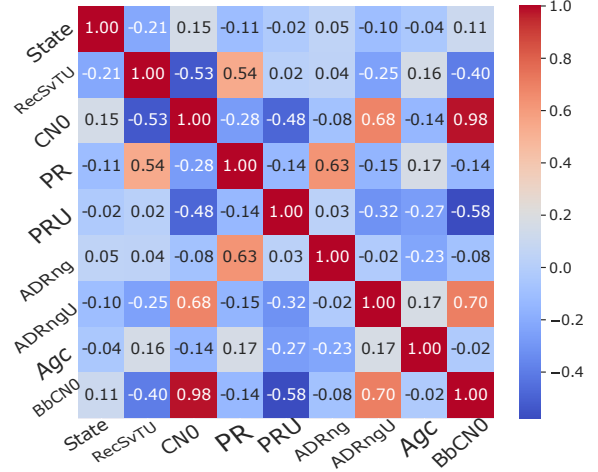


Fig. 4: Correlation of various semi-processed GPS parameters

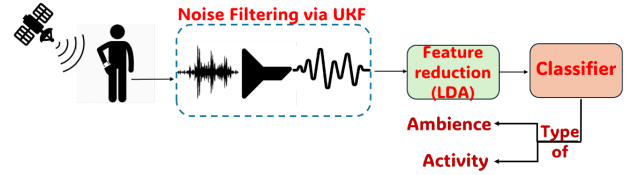


Fig. 5: Flow diagram of our ML model

as an *event*. Figure 5 shows our proposed ambient sensing framework.

We first collect 8 features from the phone. Recall that we had discarded BbCN0. Then, we eliminate the nonlinear noise using Unscented Kalman filtering (UKF). Subsequently, we perform feature reduction using the LDA algorithm. We need to classify the features into C classes, then the output is a vector with $C - 1$ elements (dimensions reduced). Then we use an ML-based classifier to map the reduced set of features to a class label. In the evaluation section (Section VI), we shall evaluate different types of ML models.

A. Preprocessing

a) **Noise Filtering:** GPS signals are susceptible to multipath effects, resulting in noise that affects the accuracy of activity or ambient-recognition models. While the Kalman Filter effectively filters noise in linear systems, it is inadequate for the nonlinear dynamics inherent in GPS signals [39]. We thus use the Unscented Kalman Filter (UKF) [20], which is adept at handling these nonlinearities by employing a deterministic sampling approach. By selectively filtering the *non-critical* noise components, the UKF maintains a balance between noise reduction and the retention of essential signal features.

B. Feature Reduction

Prior to applying LDA, StandardScaler [40] normalization was used to address the issue of differing scales and normalize

the semi-processed parameters. The output is a $C - 1$ element vector.

VI. AMBIENT SENSING AND HUMAN ACTIVITY RECOGNITION

We group ambient sensing and human activity recognition (HAR) into one section because the techniques that are used are similar.

A. Evaluation Setup

1) *ML Models*: We evaluated six machine learning algorithms: Random Forest (RF), K-Nearest Neighbor (KNN), Support Vector Machine (SVM), Decision Tree (DT), Naive Bayes (NB) and Gradient Boosting (GB). Given that we were getting a high accuracy, we did not use more sophisticated CNN-based models. These algorithms were chosen due to their popularity in the related work [41], [42]. Some of them also produce explainable models and results. To ensure the model's generalizability to new data and avoid overfitting, extensive empirical analyses were conducted to determine the optimal hyperparameters for each model using GridSearchCV [43] (refer to Table III).

TABLE III: Hyperparameters used for each classifier

Classifier	Hyperparameters
Random Forest	n_estimators: 100, max_depth: 10, min_samples_split: 10, min_samples_leaf: 4, bootstrap: True
Support Vector Machine	C: 110, kernel: rbf, tol: 0.001, break_ties: True
K-Nearest Neighbors	n_neighbors: 10, weights: distance, algorithm: auto, leaf_size: 30
Decision Tree	max_depth: 20, min_samples_split: 20
Naive Bayes	var_smoothing: 1e-9, fit_prior: True, class_prior: None
Gradient Boosting Classifier	n_estimators: 100, learning_rate: 0.01, max_depth: 10, min_samples_split: 10, min_samples_leaf: 4

Notably, for SVM, we adopted the one-versus-one (OVO) strategy to address class imbalance, transforming the multi-class classification problem into multiple binary classification tasks [44]. Additionally, a ten-fold cross-validation method was used to minimize overfitting. The dataset was randomly partitioned into 80% for training and 20% for testing.

2) *Details of the Platform*: The learning models were trained on the Google Collaboratory cloud platform¹. The cloud-based system was equipped with a dual-core Intel® Xeon® CPU running at 2.20GHz, 13GB of RAM, and a disk capacity of 107.72 GB. The hyperparameter tuning task was slow and time consuming. Hence, we used a NVIDIA® Tesla® T4 GPU equipped with 15GB of dedicated RAM.

¹<https://colab.research.google.com/>

B. Ambient Sensing

1) *Data collection*: For each ambient setting, over 100K samples were collected in various locations such as a dormitory floor, stadium, bustling market and underground metro tunnel. The data collection involved 20 volunteers who were graduate and undergraduate students in a premier university. They were aged between 19 and 28. The volunteers were first rigorously trained on how to take measurements. Furthermore, to ensure the generality of the dataset, the volunteers were requested to vary the settings within limits. For example, one group put the mobile phones in their shirt pockets, some kept it in their pant pockets and some in their purses.

2) *Class Labels*: Five user environments were defined: flight¹, indoor, metro tunnel, open ground and outdoor crowded area. Indoors, open areas and outdoor crowded areas are distinguishable using precise location coordinates and mapping services like Google Earth². This is because multiple locations belonging to different classes maybe in close proximity and it proved to be hard to distinguish them in our experiments. In any case, the objective of this work is to use GPS semi-processed data alone. For the open space and outdoor crowded area settings, samples were collected in an open-air stadium. We considered two cases: empty and filled (with people).

3) *Results with Parameters Considered Individually*: We evaluated the effectiveness of each of the semi-processed GPS parameters (after Kalman Filtering (UKF)) for characterizing the ambient environment (see Figure 6).

In the interest of space, only the most accurate results for each parameter (across all models) have been shown. Agc classifies the ambient with an average accuracy of 86.7% while ADRng performs the worst with an accuracy of 40.6%. All other parameters have an accuracy range between 57.9-82.1%. This accuracy is on the lower side because we are considering the parameters *individually*. We combine the parameters to achieve better results.

4) *Results with Fused Parameters*: The accuracy of the models after all the steps (UKF + scaling + LDA) is shown in Figure 7. Although SVM accurately predicted the results for three settings, it failed to classify the Indoor and Metro environments with misclassification rates of 57.6% and 79.3%, respectively. Similarly, KNN achieved over 90% accuracy for all the settings other than Metro, where it had a misclassification rate of 43.4%. The NB (Naive Bayes) model also underperformed for Metro with a misclassification rate of 32.4%. The Metro setting is hard to classify because several parameters such as ADRng and State are not available. In contrast, RF (Random Forest) and GB (Gradient Boosting) exhibited the best performance with accuracies exceeding 90% for all the settings.

Table IV shows the same data for each setting using several popular ML metrics: accuracy, sensitivity, specificity and the

¹All the signals were collected in passive, receive-only mode. There was no signal transmission. This is allowed as per rules.

²<https://earth.google.com/web/>

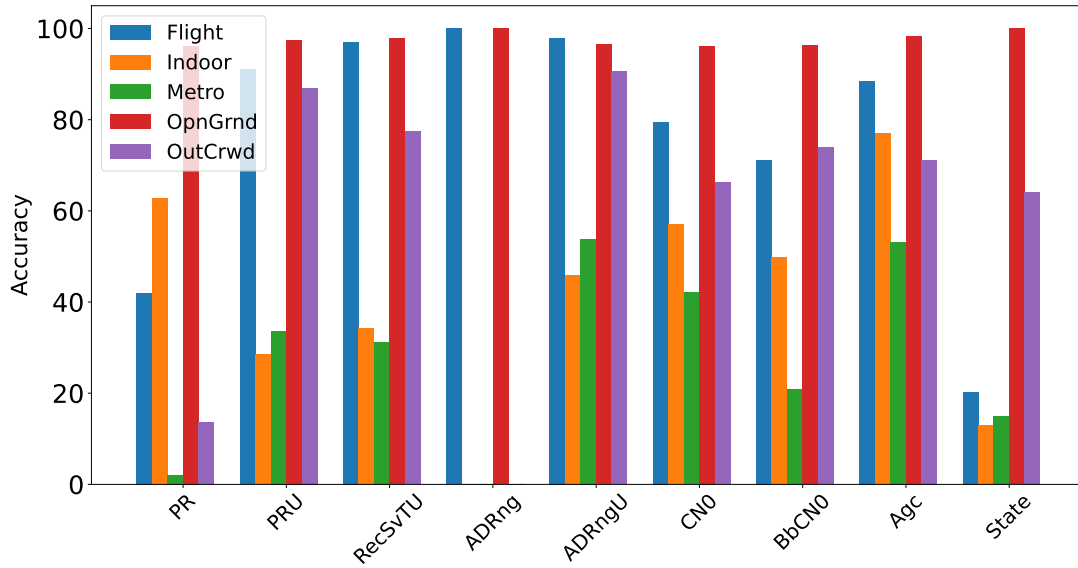


Fig. 6: Ambient sensing accuracy for each semi-processed GPS parameter

TABLE IV: Performance metrics of ambient classifier models: GB and RF are the best. The results are averaged across all the phones and volunteers.

Classifier	Ambience	Acc	Sen	Spe	F-score
GB	Flight	99.8	99.8	99.9	99.8
	Indoor	99.6	99.6	99.8	99.4
	Metro	91.6	91.6	99.9	93.9
	OpnGrnd	99.9	99.9	100.0	99.9
	OutCrwd	98.8	98.8	99.8	98.7
RF	Flight	99.5	99.9	99.9	99.8
	Indoor	99.3	99.3	99.8	99.2
	Metro	88	88.0	100.0	93.3
	OpnGrnd	99.8	99.8	100.0	99.9
	OutCrwd	98.8	98.8	99.7	98.4
DT	Flight	99.9	99.9	99.9	99.8
	Indoor	99.2	99.2	99.8	99.1
	Metro	91.6	91.6	99.9	93.2
	OpnGrnd	99.8	99.8	99.9	99.9
	OutCrwd	98.8	98.8	99.9	98.9
KNN	Flight	97.5	97.5	98.7	97.2
	Indoor	92.8	92.8	98.4	92.4
	Metro	56.7	56.7	99.7	62.2
	OpnGrnd	98	98.0	99.0	98.3
	OutCrwd	89.7	89.7	98.4	89.1
NB	Flight	79.1	79.1	99.6	87.8
	Indoor	89.7	89.7	93.0	79.6
	Metro	67.6	67.6	98.7	51.0
	OpnGrnd	97.6	97.6	99.1	98.2
	OutCrwd	90.2	90.2	97.4	86.0
SVM	Flight	96.4	96.4	88.4	85.4
	Indoor	42.4	42.4	97.7	55.0
	Metro	20.7	20.7	100.0	33.7
	OpnGrnd	97.9	97.9	97.4	97.2
	OutCrwd	84.9	84.9	97.3	82.9

F-score. We find RF and GB to be the best models across metrics.

5) *Characterization of the Indoor Environment:* Additionally, we further characterized the indoor environment based on population density. We consider two settings: nobody is present in a small (12 ft x 8 ft) room and two people are present. The GB model accurately classified both scenarios with an accuracy of 99.25%, as shown in Figure 8.

6) *Relative Importance of Features:* We further evaluated the importance of each feature. This involves randomly shuffling the values of each parameter and measuring the resulting increase in the model's prediction error [45]. A feature is deemed important if shuffling its values leads to a significant increase in model error, indicating that the model relied on that feature for accurate predictions. Agc has the highest importance score of 0.78, while all other features have importance scores ranging from 0.55 to 0.75 (see Figure 9), demonstrating their substantial contributions to classification.

C. Human Activity Recognition(HAR)

1) *Class Labels:* Prior work [17], [18], [46]–[48] has extensively explored various HAR systems. They cover both static activities (sitting, lying and standing) and dynamic activities (such as walking, running or traveling in a vehicle). The latter can be discerned on the basis of the speed or velocity. The Android platform's `getSpeed` API [49] in the `android.location.Location` package facilitates this. In our case, we shall use Doppler shifts [49] for this purpose (captured by the PR attribute).

Distinguishing static activities cannot be done on the basis of Doppler shifts. We wish to classify static activities into four classes: sitting, standing, lying down and hand waving (mobile phone not held with the moving hand). In case of hand waving, the device was mounted on a wall stand and hand movements were performed in its close proximity (within 2m).

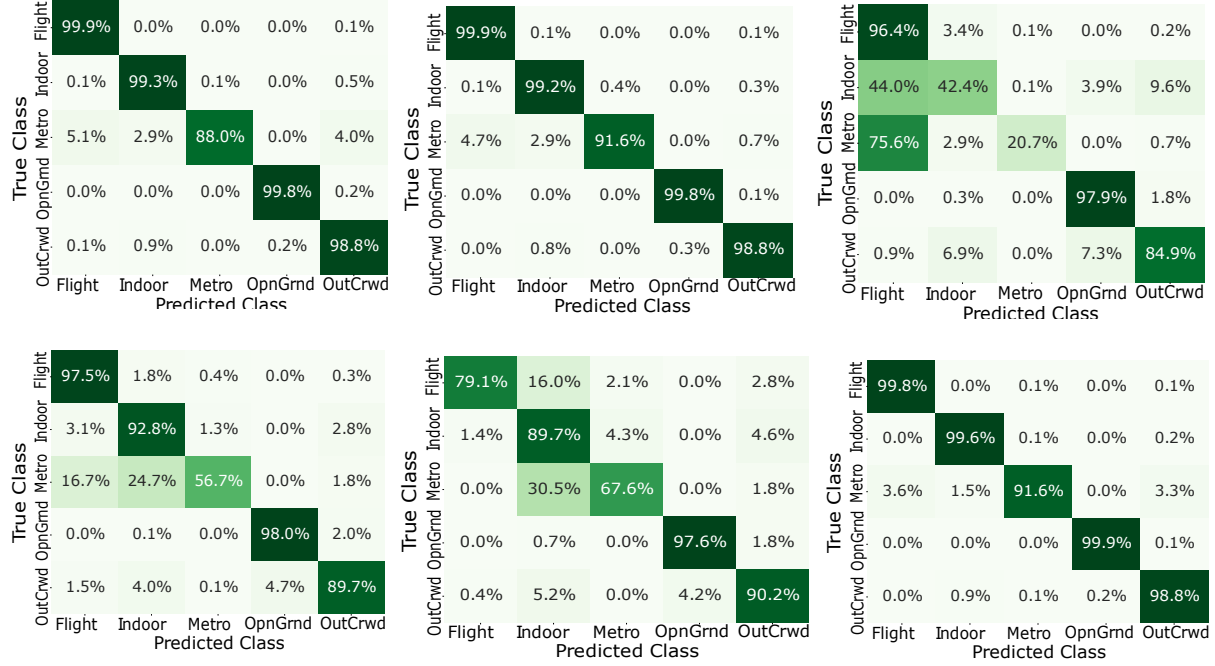


Fig. 7: Ambient classification confusion matrix: (a) RF, (b) DT, (c) SVM, (d) KNN, (e) NB, and (f) GB

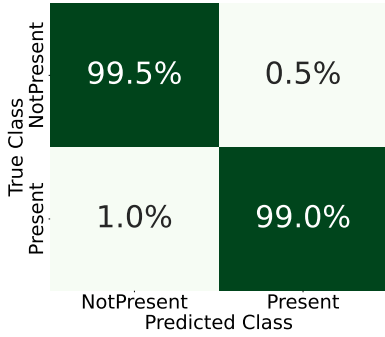


Fig. 8: Confusion matrix of indoor environment with different population density

2) *Results with Individual GPS Parameters:* Figure 10 shows the accuracy of each activity for each semi-processed GPS parameter. The results are consistent with previous observations. Agc achieves the highest accuracy of 77.2%, while ADRng performs the worst with a high misclassification rate of 72.6%. The accuracy for other parameters ranges from 35% to 56% reflecting the inability of individual parameters in classifying activities. Hence, we combine the parameters to achieve better results.

3) *Results with Fused Parameters:* Figure 11 compares model performance using the fused parameters. SVM achieves 93.4% accuracy for lying down but has an accuracy of 60% for other activities. KNN accurately classifies hand waving with an accuracy of 88.4% but has an average accuracy of

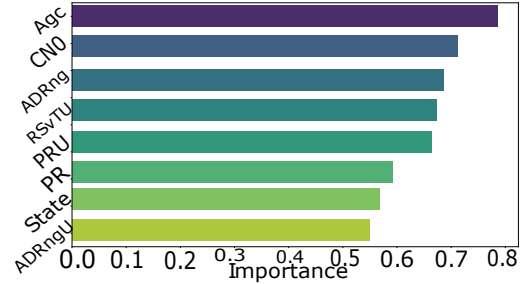


Fig. 9: Feature importance test for ambient classification (GB model)

74.4% for other activities. NB identifies the lying down posture with $\geq 80\%$ accuracy but performs poorly (about 67% on average) for the other three activities. RF, DT, and GB are the best – they achieve a minimum accuracy of 80%. The peak accuracy is 97.2% (RF).

Table V shows the same data along with other ML metrics: sensitivity, specificity and the F-score. We can observe that SVM performs the worst (accuracy = 57.4%) and RF performs the best (accuracy = 87%). Consequently, RF was selected as the model of choice for this activity.

4) *Relative Importance of Features:* Additionally, we evaluated the importance of each feature in HAR when they are fused together. The feature Agc has the highest importance score of 0.88, while all other features have importance scores in the range of 0.5 to 0.65 (see Figure 12). This further strengthens our claim – using fused parameters improves

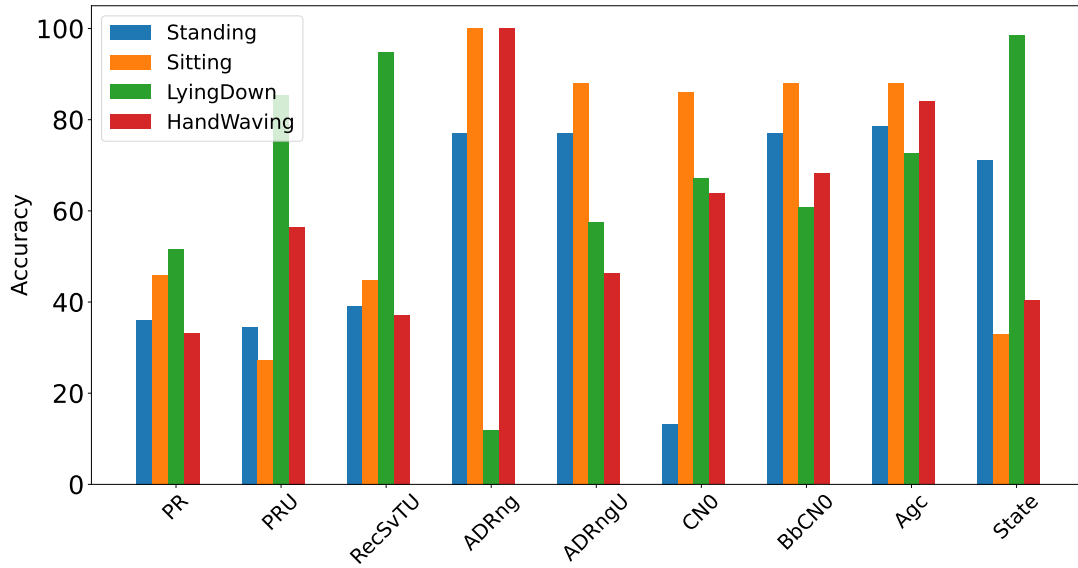


Fig. 10: Activity classification accuracy for each semi-processed GPS parameter

TABLE V: Performance metrics of activity classifier models

Classifier	Activity	Acc	Sen	Spe	F-score
RF	HandWaving	77.5	77.5	98.9	85.6
	LyingDown	97.2	97.2	96.5	92.6
	Sitting	91.2	91.2	92.3	86.4
	Standing	82.7	82.7	94.7	83.8
DT	HandWaving	80.7	80.7	94.7	81.7
	LyingDown	84.9	94.9	98.2	94.2
	Sitting	85.1	85.1	93.2	84.0
	Standing	87.9	87.9	96.3	88.8
GB	HandWaving	82.4	82.4	96.7	85.4
	LyingDown	96.9	96.9	96.7	92.8
	Sitting	86.7	86.7	94.1	85.9
	Standing	82.8	82.8	94.8	84.0
KNN	HandWaving	71.8	71.8	94.2	75.4
	LyingDown	88.4	88.4	93.4	83.3
	Sitting	68.9	68.9	85.6	66.8
	Standing	56.9	56.9	87.3	59.3
NB	HandWaving	61.4	61.4	94.6	68.8
	LyingDown	80.7	80.7	89.5	73.8
	Sitting	77.8	77.8	90.1	76.4
	Standing	63.2	63.2	86.6	63.1
SVM	HandWaving	60.3	60.3	91.0	63.8
	LyingDown	93.4	93.4	78.1	68.7
	Sitting	38.3	38.8	87.7	45.4
	Standing	44.8	44.8	86.8	49.4

accuracy.

D. Assessment of the Robustness

We conduct several experiments to demonstrate the robustness of our method. We consider different environmental conditions, variations in the satellite vehicle ID (SvID) density, different test:train split, sparse fingerprinting techniques and accuracy variability across different phones.

1) *Variation in the Number of Unique SvIDs*: Figure 13 shows the impact of varying the number of SvIDs on the

prediction accuracy. A subset of SvIDs was randomly selected from the logged data and only semi-processed GPS data corresponding to these SvIDs was used for evaluation. Care was taken to maintain the integrity of the dataset by ensuring that SvIDs representing a significant portion of the logged data were not removed. The dataset was becoming really sparse.

The maximum number of unique SvIDs observed during the experiment was 28. The results show that even with a SvID density as low as 50% of the total number of unique SvIDs (14 unique SvIDs) an accuracy of 92.1% (13(a)) and 84.4% (13(b)) in ambient and activity classification, respectively, was achieved.

2) *Different Train:Test Splits*: Figure 14 shows the accuracies when we vary the train:test ratio. For example, a ratio of 80:20 indicates that the training data is 80% of the overall dataset and the remaining 20% of the data is used for testing purposes.

Even when the proportion of test data is as high as 50%, the average accuracy of ambient classification is 94.2% (14(a)), and 85.4% (14(b)) for activity classification, respectively. We can attribute this high degree of resilience to the UKF's noise removal and feature preservation capabilities. When we remove UKF, the accuracy drops to roughly by more than 50%.

3) *Unseen Events*: Additionally, the accuracy of the model was assessed by experiments performed by volunteers using another set of Android phones on settings that are similar to the settings that we have been using up till now. It is important to note that up till now we only used a set of 5 phones and the same settings even though we varied the following parameters: time of the year, time of the day, weather condition and there was some noise. For example, in a crowded area, the composition of the crowd varied. In this case, we vary everything other than the nature of the setting. For example, we

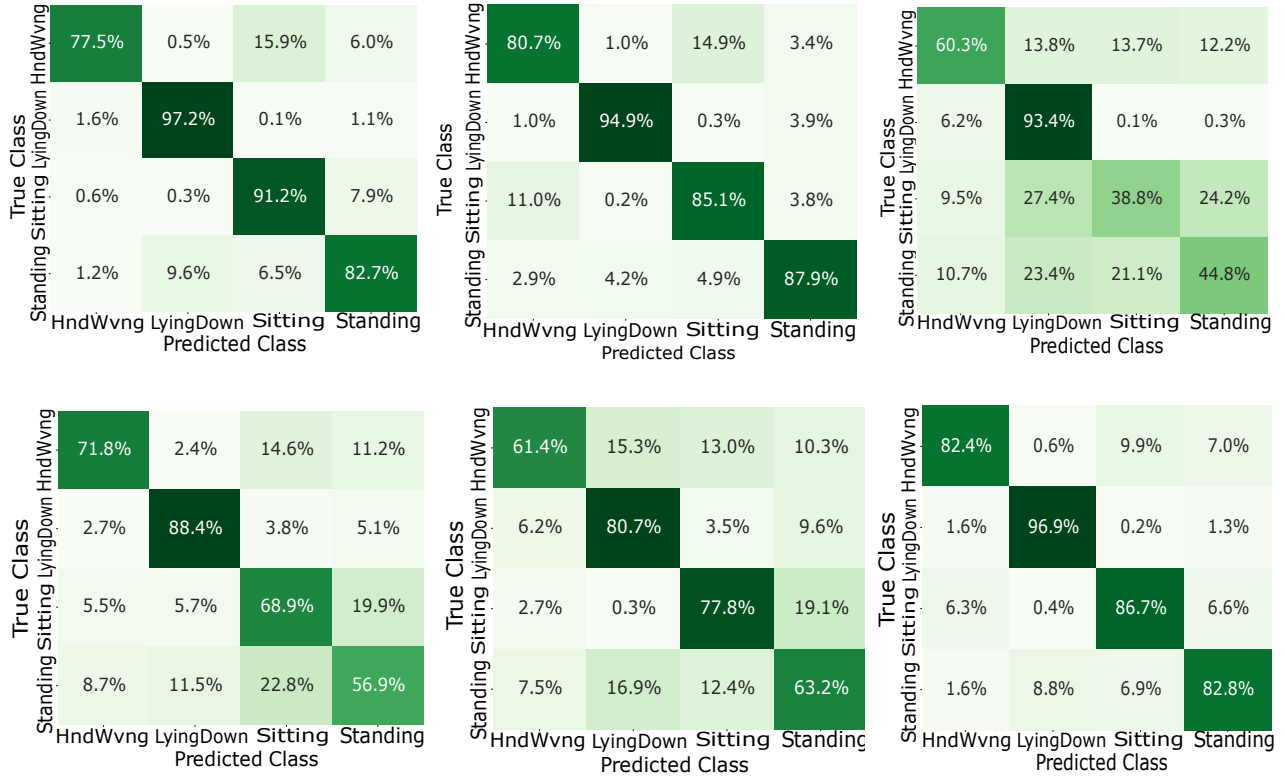


Fig. 11: Activity classification confusion matrix: (a) RF, (b) DT, (c) SVM, (d) KNN, (e) NB, and (f) GB

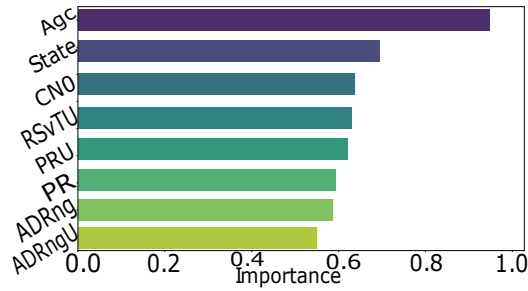


Fig. 12: Feature importance test for human activity recognition

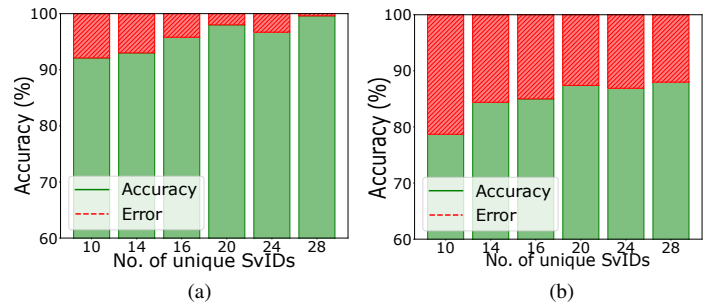


Fig. 13: Accuracy on varying the number of SvIDs for (a) ambience classification and (b) activity classification

collected measurements at locations that were roughly 1000 kms away from our original location. One location had an altitude of 3.5 kms. We also tested on a cruise ship, which we categorized as an open space.

Figure 15 shows the results. While a decrease in accuracy was observed, it remained consistently above 91.5% 15(a) for ambient recognition and 83% 15(b) for activity classification, respectively. The best-performing technique was still GB for ambient classification and RF for human activity recognition. This reduction in precision was mainly due to variations in multipath effects and the significantly different experimental setups. Moreover, these volunteers did not go through our

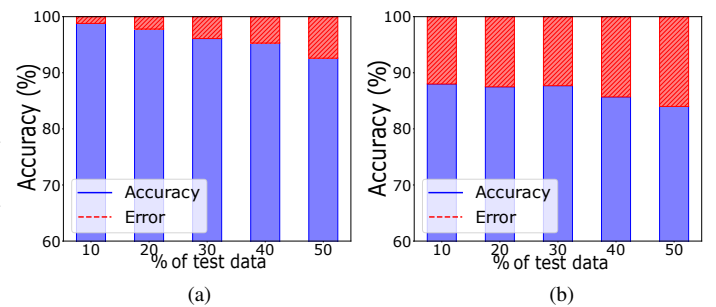


Fig. 14: Accuracy with different level of train:test ratio for (a) ambience classification and (b) activity classification

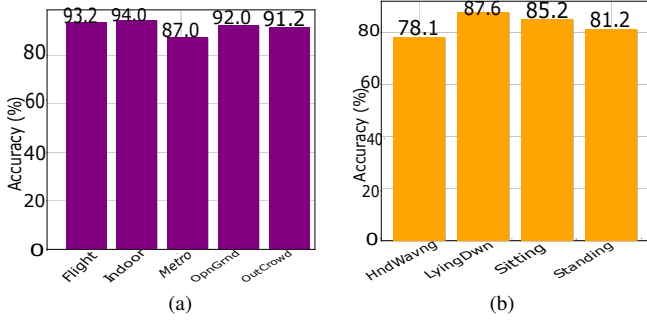


Fig. 15: Accuracy of events performed by random volunteers at random locations for (a) ambience classification and (b) activity classification

orientation process. There is a possibility that our orientation process may have sub-consciously created some degree of homogeneity, even though we tried our best to avoid it. With this fresh set of volunteers, this did not happen. It is not possible for us to say what exactly was the cause for the reduction in accuracy.

4) *Variability across different phones:* We investigated the variability in accuracy across different testing phones under similar conditions, noting a significant 8% difference between two Redmi Note 9 Pro Max models of different build years. Additionally, to capture environmental variability, we logged data at the same place but at different times of the day. This resulted in a 9% difference in accuracy for the same phone and a 13% difference in accuracy for different phones. Specifically, the Galaxy A53 (2023) achieved the highest average accuracy of 92.3% in ambient sensing, while the Redmi K20 Pro (2020) exhibited the lowest at 81.8%. This disparity suggests that older models generally have lower accuracy, primarily due to GPS receiver noise. However, after UKF followed by LDA, the maximum accuracy difference decreased to 3% for data logged at the same time and 4.2% for data logged at the same place but at different times. This underscores the effectiveness of our approach in mitigating noisy readings across diverse phone models and handling varying ambient conditions at the same location.

VII. FINDING THE INDOOR FLOOR LAYOUT

Our goal is to create an indoor floor map using semi-processed GPS data. Prior work [50] has used WiFi RSSI (received signal strength indicator), user activities (measured via accelerometers or pressure sensors), and graph optimization techniques to construct the indoor map. These approaches do not work for us given that the RSSI proves to be quite a feeble technique in the case of GPS signals. For WiFi the source of the signal is nearby, hence, its strength variation carries more information. However, for us the source is very far away and the variation observed is minimal. It is often affected by noise to a much greater degree. This had necessitated the need for UKF and LDA in the first place. Nevertheless, we use RSS

(Age) as the starting point of our investigation and then build on it since is clearly inadequate for our purpose.

A. Study of GPS RSS Patterns

To study GPS RSSI variations across different areas of a large indoor area, we conducted an experiment in a dormitory corridor. It measured 32.3 m \times 25.6 m (refer to Figure 16(a)).

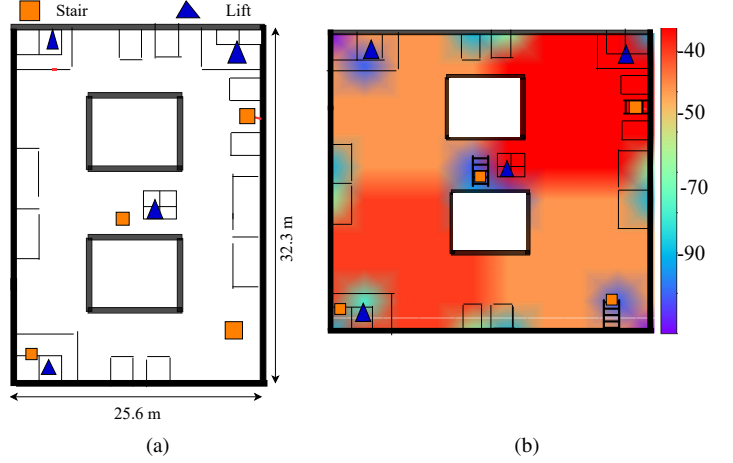


Fig. 16: (a) Floor layout of the dormitory corridor and (b) GPS RSS heatmap of the dormitory corridor

Since there are no GPS-based heatmap simulators available, and civilian GPS operates at a frequency that is very close to 2.4 GHz WiFi, we utilized NetSpot, a commercial tool designed for WiFi signal strength mapping. We generated a heatmap depicting the relative GPS RSS levels across the site (see Figure 16(b)). We assumed ideal, zero-noise conditions. To simulate a GPS setting, we introduced two access points situated outside the site, specifically in the open areas to the northeast and southwest. The differences in the RSS readings indicate their potential for inferring the layout. We scaled the data to match the real-world GPS signal intensity, which was measured using the RF Explorer Spectrum Analyzer at all points of interest: near entrances, lifts, stairs and corners. A correlation of 0.83 was observed, which is quite high. The heatmap clearly identifies the points of interest such as the three lifts at the corners, the rooms and the entrance to the stairs. But the data still does not tell us what is there at these points of interest. Even with WiFi signals, accelerometer and pressure sensor data was required. We are replacing it with semi-processed GPS data and the GPS location information, which we have not used up till now.

B. Trajectory Creation Methodology: Use Location Data

1) *Activity Landmarking:* Our aim is to identify floor landmarks such as pathways, stairs, elevators, rooms, and obstructions based on user activity and GPS-based location information. We will not use data from any other sensor. For example, users typically walk straight along pathways, turn at obstructions, and engage in different activities on stairs or in elevators. These activities can be captured using our human activity recognition (HAR) framework. Notably, elevators act

as Faraday cages [51], and they block GPS signals. We shall rely on multiple users to collect this data – semi-processed GPS parameters, HAR fingerprints and trajectory information. Then we shall merge these results to **identify the landmarks**.

2) *Trajectory Alignment*: We collected the walking trajectories of our volunteers. It is necessary to employ a trajectory *alignment* algorithm (akin to [50]) because different people will produce different trajectories. Let us briefly described the algorithm in [50]. It involves using a transformation matrix, which it creates, for adjusting the curvilinear paths, translation and rotation. This matrix is based on common “activity landmarks” in the different trajectories. Trajectory coordinates are classified as either activity landmark coordinates (ALC) or non-activity landmark coordinates (NALC). When merging trajectories, we focus on common ALCs and calculate their relative coordinates to determine the necessary translations and rotations. This iterative process ensures a more accurate and consistent overall map by aligning each new path to the existing map. The advantage of aligning trajectories is that we create a robust skeleton of the internal map. A single observation is prone to a lot of noise especially when using GPS signals.

After aligning the trajectories with themselves and with a virtual coordinate system, we apply graph optimization techniques (refer to [50]) to further refine the locations of the ALC and NALC trajectory points. The Levenberg-Marquardt algorithm [52] is used for optimization. The result is a map of the site with the ALC points (specifically) indicated.

C. Evaluation of the Accuracy of Classifying ALC Points

1) *Data collection*: 10 volunteers participated in the data logging process. The data was collected using the GnssLogger Android application, the same tool employed for event classification. Each volunteer carried a device with the GnssLogger app installed and traversed the floor, and sometimes used the services at the landmarks. To ensure robustness and mirror real-time scenarios, participants were requested to start and end their journeys at random points on the floor.

2) *Results of the Trajectory Creation Algorithm*: Figure 17 illustrates the outcome of our mapping process(setting: dorm-room corridor). The activity landmark trajectory data is shown in Figure 17(a). Given the initial random alignment of the trajectories, we realign them to eliminate inconsistencies as shown in Figure 17(b). The resultant map contains noise, which we remove using graph optimization to obtain an optimized floor map with reduced errors (see Figure 17(c)). The final step involves aligning the generated map with the ALC information. The final floor map is shown in Figure 17(d). Note that we add the real layout in the background only for the purpose of better visualization.

3) Accuracy Assessment:

a) *Layout Shape*: To evaluate the quality of the generated map, we use the following metrics [50], [53], [54]:

- **Graph Discrepancy Metric (GDM)**: This measures the differences between the landmark points of the generated

map and the real map (ground truth) using Euclidean distance. A smaller GDM indicates a closer match between the generated map and the real map.

- **Shape Discrepancy Metric (SDM)**: This assesses the differences in the overall shape of the generated map as compared to the real map. It involves uniformly sampling points along lines connecting landmarks of both maps and measuring the distances between corresponding sampling points. A smaller SDM indicates a greater similarity in shape between the two maps.

Figure 18(a) shows the cumulative distribution function (CDF) plot for the GDM. The maximum error observed is 4.10 m, and the 90th percentile error is 3.42 m. Similarly, for SDM, the maximum error is 3.41 m, with the 90th percentile error is 2.94 m (refer to Figure 18(b)). Our results align closely with the related work [50], [53], affirming the efficacy of our method in constructing floor maps.

b) *Landmark detection accuracy*: We utilized a GPS-based Human Activity Recognition (HAR) system to accurately detect landmarks categorized as stairs, rooms, lifts, and empty corners within our framework. Table VI shows the accuracy of the models (RF, DT, GB, KNN, NB, SVM) across four landmarks: Lift, Stairs, Rooms, and Corners. RF shows strong performance with an average accuracy of 90.15%. DT and GB also perform well, with average accuracies of 87.5% and 87.7% respectively. KNN has a moderate average accuracy of 81%. NB performs adequately with an average accuracy of 70.8%, but has a notably low accuracy for Lift (61.4%). SVM has the lowest average accuracy of 50.2%, with particularly poor performance for Rooms (54.7%) and Corners (34.3%).

TABLE VI: Accuracy of activity classifier models

Classifier	Lift	Stairs	Rooms	Corners
RF	98.5	96.2	86.2	79.7
DT	96.7	93.1	77.9	82.3
GB	96.6	93.0	78.3	82.8
KNN	91.8	84.4	72.9	74.9
NB	61.4	80.7	77.8	63.2
SVM	61.6	57.8	54.7	34.3

VIII. EXPLOITATION BY ANDROID APPLICATION

By exploiting the `ACCESS_FINE_LOCATION` permission, apps can covertly log semi-processed GPS data, which as we have seen poses privacy risks. While users may agree to precise location access for legitimate services, apps could misuse data for targeted ads or privacy breaches. Android 12 introduced a privacy option for selecting between fine and coarse location permissions. Before this, there was only a single location permission. This enhanced the control that users had over location data access [55].

Countermeasures: Android 12 (and beyond) users can limit permissions to *coarse location*, thus restricting app access to the semi-processed GPS data. Android mandates developers to explicitly outline the required permissions in the app description [56] file. Android 11 and earlier versions users are often advised to check descriptions for fine location requests. They

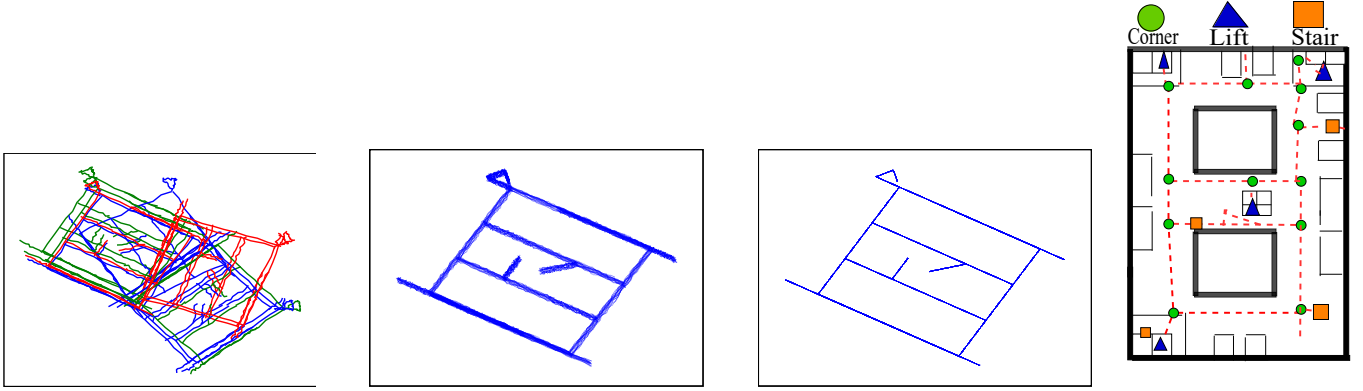


Fig. 17: Result of the indoor floor mapping experiments: (a) Raw trajectories, (b) After trajectory alignment, (c) After graph optimization, and (d) Final result after aligning with a coordinate system

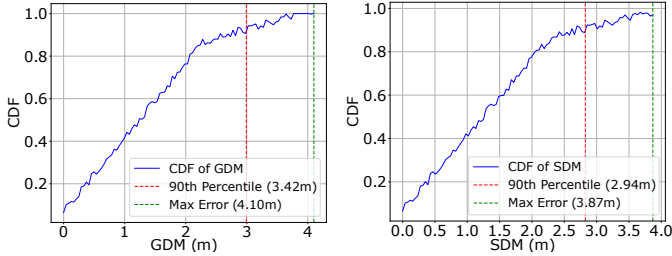


Fig. 18: CDF of (a) GDM and (b) SDM

often overlook such details during installation [57] risking privacy breaches.

A. Impact

Disabling fine location permissions isn't practical, as developers may justify its need for location-based services. Even vigilant users regardless of whether they check app descriptions for requested permissions or choose between permissions are still vulnerable to privacy breaches. This vulnerability persists as users are unaware of the privacy risks associated with logging semi-processed GPS data, owing to the *absence* of relevant disclosures in the official Android documentation [55].

Additionally, devices running Android 10 or later must support semi-processed GPS measurements, while for Android 9 and earlier, this is mandatory for devices manufactured in 2016 or later [34]. Currently, **over 90%** of Android phones support semi-processed measurements [34], exposing a substantial portion of the user base to this kind of a covert attack.

IX. RELATED WORK

Given that there is an overlap in the prior work on activity and ambience recognition, we combine them and discuss the combined related work in Section IX-A. Subsequently, we shall discuss the related work on layout detection in Section IX-B.

A. Ambient Sensing and HAR

We present a brief comparison of the related work in Table VII. Image-based solutions are prominent due to their high accuracy [58]. However, their widespread adoption is limited by the high cost of processing image data and security concerns.

In response, wireless signal patterns: RSS and channel state information (CSI) is used for ambient and activity recognition. Wang et al. [59] introduced a deep learning technique, a sparse autoencoder (SAE), which recognizes events using WiFi CSI signals. Gao et al. [60] developed another CSI-based system for event classification using deep learning, where they converted CSI measurements into radio images, extracted features, and then applied an SAE network for better accuracy. Muaaz et al. [61] proposed WiWeHAR, a multimodal HAR system that combines WiFi CSI data and wearable inertial measurement unit (IMU) sensor data. These systems demand precise feature engineering and efficient signal processing and noise removal.

RSS-based solutions are comparatively more noise-tolerant. One of the earliest works is PAWS [17], which utilizes ambient WiFi signals to create RSSI fingerprints. Bhat et al. [62] refined existing techniques to develop a recognition system using the RSS of a single communication channel. Shuaib et al. [63] introduced an RFID-based indoor HAR system that uses RSS from passive RFID tags to track activity in real-time by mapping the analyzed data to reference datasets. Such solutions rely heavily on the availability of good-quality WiFi signals, which may not always be feasible especially in outdoor settings. GPS, on the other hand, is more ubiquitously available.

Sekiguchi et al. [18] categorize events based on GPS coordinates along with cell tower and WiFi signal data. Our approach does not use GPS coordinates or any other kind of location information. We solely focus on semi-processed GPS signal parameters, which are almost always available.

Year	Work	Approach	WiFi	IMU	Cell Tower	GPS	RFID	Camera
2021	Ramirez et al. [58]	Video-based trajectory	X	X	X	X	X	✓
2016	Wang et al. [59]	Wireless signal CSI patterns	✓	X	X	X	X	X
2017	Gao et al. [60]	Wireless signal CSI patterns	✓	X	X	X	X	X
2020	Muaaz et al. [61]	WiFi CSI + IMU sensors	✓	✓	X	X	X	X
2015	PAWS [17]	RSS-Based	✓	X	X	X	X	X
2020	Bhat et al. [62]	WiFi and RSS-based	✓	X	X	X	X	X
2020	Shuaib et al. [63]	RFID-Based	X	X	X	X	✓	X
2021	Sekiguchi et al. [18]	GPS coordinates + Other wireless signals	✓	X	✓	✓	X	X
2020	Bui et al. [64]	processed GPS signals	X	X	X	✓	X	X
2024	Zhu et al. [65]	processed GPS signals	X	X	X	✓	X	X
2024	<i>AndroCon</i>	Semi-processed GPS signal parameters	X	X	X	✓	X	X

TABLE VII: Comparison of Event Classification Approaches

We do not use any WiFi or assisted-GPS signals (signals from cellphone towers).

Bui et al. [64] utilized the magnitude of the received GPS signal for distinguishing between indoor and outdoor environments. Zhu et al. [65] classify ambient conditions based on visible satellites, GNSS distribution, C/N0, and multipath effects. However, both studies utilize processed signal data, which yields less accurate results compared to our research. Our work also encompasses diverse settings that were not accounted for in either study.

B. Indoor Layout Mapping

We show a comparison of related studies in Table VIII. Most approaches utilize Simultaneous Localization and Mapping (SLAM), a computer vision technique designed to localize a robot in an unknown environment while concurrently generating a map. Conventional SLAM techniques typically depend on visual cues such as landmarks, camera-detected obstacles, sonar data or laser-range sensors (Kamar et al. [66]) [10], [70], [71]. However, these approaches may incur high costs or maybe quite intrusive in terms of privacy.

Alzantot et al. [67] propose to use crowd-sourced user trajectories to construct indoor maps. However, this method was susceptible to errors due to the non-alignment of trajectories. Addressing the issue, Shen et al. [54] present an indoor pathway mapping system using WiFi RSS fingerprints to align trajectories. Gu et al. [68], [72] further improved the performance by employing Bluetooth and WiFi data for trajectory alignment, but sparse WiFi access point deployment poses practical challenges. Philipp et al. [69] propose Mapgenie – generating maps from pedestrian movement traces and utilizing external building data and applies grammatical rules to model and interpret pedestrian movement patterns; however, it requires a foot-mounted IMU that is typically not found on smartphones. Zhou et al. [53] use a link-node approach (landmarks as node and pathways and links). This method utilizes smartphone sensors and WiFi RSS fingerprinting, complemented by geometric scaling methods. However, challenges arise in accurately representing curved features within the maps. Zhou et al. [50] address these challenges through graph optimization methods.

Our approach draws inspiration from Zhou et al. [50] that leverages WiFi RSS and MAC address fingerprinting

for trajectory alignment and landmark identification. Sadly, Zhou et al.’s method relies heavily on mobile sensors such as gyroscopes and compass readings, which may introduce errors and reduce reliability. Moreover, it requires extensive user permissions and depends on the availability of WiFi access points. In contrast, our proposed method utilizes semi-processed GPS data and does not need any other information.

For both event recognition and indoor floor mapping, image-based solutions [10], [58] captured using cameras are highly effective; however, they raise significant privacy concerns. This has limited their widespread adoption.

In contrast, methods utilizing EM waves such as WiFi, Bluetooth or GPS are preferred due to their covert nature. Many of these methods have matured over time and they provide solutions with a comparable quality as camera-based designs do. Our solution, AndroCon, also provides similar accuracies (if not better).

X. CONCLUSION

The fact that semi-processed GPS data can be used to sense the ambient, recognize human activity and figure out floor layouts was hitherto unknown. We successfully showed that all of the above can be achieved, and that too with a high accuracy that ranges from roughly 99.5% in controlled conditions to 87% in absolutely uncontrolled conditions. We conducted an extensive set of experiments with tens of volunteers, many phones of different makes and brands, diverse set of scenarios and thousands of sample points. We collected data for a year across a large geographical area – some of the collection points were 1000 kms away from the place where this research was carried out. We also collected data on flights, cruise ships and high-altitude locations. Our results are thus robust.

Furthermore, our approach can construct floor maps with a maximum error margin of 4.1m as compared to the ground truth. We can classify points of interest within an indoor layout such as elevators, stairs, corridors, empty corners and rooms with a roughly 90.15% accuracy, while just relying on GPS data. Currently, Android does not address this vulnerability, which leaves approximately 90% of users exposed.

XI. RESPONSIBLE DISCLOSURE

We reported the identified vulnerability and use-cases, excluding the floor mapping use-case to the Android security

Year	Work	Approach	SLAM	WiFi	Smartphone sensors	IMU	GPS
2020	Karam et al. [66]	IMU + LiDAR	✓	✗	✗	✓	✗
2012	Alzantot et al. [67]	User trajectories	✗	✗	✓	✗	✗
2013	Shen et al. [54]	WiFi RSS alignment	✗	✓	✗	✗	✗
2016	Gu et al. [68]	User trajectory +WiFi+Bluetooth	✗	✓	✓	✗	✗
2014	Philipp et al. [69]	Pedestrian movement+crowd-sourced structural (external) information of buildings	✗	✗	✗	✓	✗
2015	Zhou et al. [53]	Link-node approach	✗	✓	✓	✗	✗
2018	Zhou et al. [50]	Link-node approach+GO	✗	✓	✓	✗	✗
2024	AndroCon	GPS Semi-processed data + user trajectory	✗	✗	✗	✗	✓

TABLE VIII: Comparison of Indoor Floor Mapping Approaches

team. They were able to reproduce some of the results and they acknowledged our concern.

REFERENCES

- [1] H.-L. Yang and S.-L. Lin, "User adoption of location-based service," in *2018 Tenth International Conference on Advanced Computational Intelligence (ICACI)*, 2018, pp. 51–56.
- [2] Mappitall, "How Location-Based Services Function and Why They Are Essential," Accessed on: April 16 2024. [Online]. Available: <https://www.linkedin.com/pulse/how-location-based-services-function-why-essential-mappitall-vulbf/>
- [3] M. Jackiewicz, "Location-based app development: a complete 2022 guide for CEOs," Accessed on: April 16 2024. [Online]. Available: <https://www.rst.software/blog/location-based-app-development-a-complete-2022-guide-for-ceos>
- [4] A. M. Research, "Location Based Services Market Size, Share, Competitive Landscape and Trend Analysis Report by Component, by Technology, by Application, by Industry Vertical : Global Opportunity Analysis and Industry Forecast, 2021-2031," Accessed on: April 16 2024. [Online]. Available: <https://www.alliedmarketresearch.com/location-based-services-market>
- [5] J. Lindner, "Location based marketing statistics [fresh research]," *Gitnux*. <https://www.gitnux.org/location-based-marketing-statistics/>, Dec 2023, [Online; accessed on: April 16 2024].
- [6] S. Wang, Q. Hu, Y. Sun, and J. Huang, "Privacy preservation in location-based services," *IEEE Communications Magazine*, vol. 56, no. 3, pp. 134–140, 2018.
- [7] J. Lau, B. Zimmerman, and F. Schaub, "Alexa, are you listening? privacy perceptions, concerns and privacy-seeking behaviors with smart speakers," *Proceedings of the ACM on human-computer interaction*, vol. 2, no. CSCW, pp. 1–31, 2018.
- [8] L. Atallah, M. ElHelw, L. Wang, and B. Lo, "Behaviour profiling with ambient and wearable sensing," in *4th International Workshop on Wearable and Implantable Body Sensor Networks (BSN 2007)* March 26–28, 2007 RWTH Aachen University, Germany. Springer, 2007, pp. 133–138.
- [9] J. L. Carrera, Z. Zhao, T. Braun, and Z. Li, "A real-time indoor tracking system by fusing inertial sensor, radio signal and floor plan," in *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. IEEE, 2016, pp. 1–8.
- [10] Z. Kang, J. Yang, Z. Yang, and S. Cheng, "A review of techniques for 3d reconstruction of indoor environments," *ISPRS International Journal of Geo-Information*, vol. 9, no. 5, p. 330, 2020.
- [11] D. R. Beddiar, B. Nini, M. Sabokrou, and A. Hadid, "Vision-based human activity recognition: a survey," *Multimedia Tools and Applications*, vol. 79, no. 41, pp. 30 509–30 555, 2020.
- [12] S. K. Yadav, K. Tiwari, H. M. Pandey, and S. A. Akbar, "A review of multimodal human activity recognition with special emphasis on classification, applications, challenges and future directions," *Knowledge-Based Systems*, vol. 223, p. 106970, 2021.
- [13] L. Bibbò, R. Carotenuto, and F. Della Corte, "An overview of indoor localization system for human activity recognition (har) in healthcare," *Sensors*, vol. 22, no. 21, p. 8119, 2022.
- [14] "What Is Ambient Advertising? (Advantages and Examples)," 2022, [accessed on: April 16 2024]. [Online]. Available: <https://www.indeed.com/career-advice/career-development/ambient-advertising>
- [15] "Easy Ways to Fix the GPS Signal Not Found Error in Pokémon Go," 2024, [accessed on: July 2 2024]. [Online]. Available: <https://www.wikihow.com/Pokemon-Go-GPS-Signal-Not-Found>
- [16] Q. Pu, S. Gupta, S. Gollakota, and S. Patel, "Whole-home gesture recognition using wireless signals," in *Proceedings of the 19th annual international conference on Mobile computing & networking*, 2013, pp. 27–38.
- [17] Y. Gu, F. Ren, and J. Li, "Paws: Passive human activity recognition based on wifi ambient signals," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 796–805, 2015.
- [18] R. Sekiguchi, K. Abe, s. shogo, M. Kumano, D. Asakura, R. Okabe, T. Kariya, and M. Kawakatsu, "Phased human activity recognition based on gps," 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*, 2021, pp. 396–400.
- [19] Y. Yuan, F. Shen, and X. Li, "Gps multipath and nlos mitigation for relative positioning in urban environments," *Aerospace science and technology*, vol. 107, p. 106315, 2020.
- [20] G. Hu, B. Gao, Y. Zhong, and C. Gu, "Unscented kalman filter with process noise covariance estimation for vehicular ins/gps integration system," *Information Fusion*, vol. 64, pp. 194–204, 2020.
- [21] F. A. Administration, "Satellite navigation - global positioning system (gps)," 2022. [Online]. Available: https://www.faa.gov/about/office_org/headquarters_offices/ato/service_units/techops/navservices/gnss/gps
- [22] B. N. Vu and M. Andrie, "The code and carrier tracking loops for gps signal," in *Proceedings of the 16th International Conference on Mechatronics-Mechatronika 2014*. IEEE, 2014, pp. 569–574.
- [23] "What's the difference between precise and approximate location in Android 12?" 2021, [accessed on: July 1 2024]. [Online]. Available: <https://www.androidcentral.com/whats-difference-between-precise-and-approximate-location-android-12>
- [24] [Online]. Available: <https://developer.android.com/develop/sensors-and-location/location>
- [25] M. Sugiyama, "Local fisher discriminant analysis for supervised dimensionality reduction," in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 905–912.
- [26] A. Sarveniazi, "An actual survey of dimensionality reduction," *American Journal of Computational Mathematics*, vol. 4, no. 02, pp. 55–72, 2014.
- [27] M. Sugiyama, "Dimensionality reduction of multimodal labeled data by local fisher discriminant analysis," *Journal of machine learning research*, vol. 8, no. 5, 2007.
- [28] M. Verleysen and D. François, "The curse of dimensionality in data mining and time series prediction," in *International work-conference on artificial neural networks*. Springer, 2005, pp. 758–770.
- [29] F. Zangenehnejad and Y. Gao, "Gnss smartphones positioning: Advances, challenges, opportunities, and future perspectives," *Satellite navigation*, vol. 2, pp. 1–23, 2021.
- [30] M. Filić and R. Filjar, "Smartphone gnss positioning performance improvements through utilisation of google location api," in *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. IEEE, 2018, pp. 0458–0461.
- [31] S. Malkos, "Google to provide raw gnss measurements - gps world," Jul 2016. [Online]. Available: <https://www.gpsworld.com/google-to-provide-raw-gnss-measurements/>
- [32] "Gnssmeasurement." [Online]. Available: <https://developer.android.com/reference/android/location/GnssMeasurement>

- [33] "GnssLogger App," Accessed on: April 16 2024. [Online]. Available: <https://play.google.com/store/apps/details?id=com.google.android.apps.location.gps.gnsslogger>
- [34] "Raw GNSS Measurements ." [Online]. Available: <https://developer.android.com/develop/sensors-and-location/sensors/gnss>
- [35] Google, "Google Smartphone Decimeter Challenge 2023-2024," 2024. [Online]. Available: <https://www.kaggle.com/competitions/smartphone-decimeter-2023/>
- [36] S. Dubey, R. Wahi, and A. Gwal, "Ionospheric effects on gps positioning," *Advances in Space Research*, vol. 38, no. 11, pp. 2478–2484, 2006.
- [37] [Online]. Available: <https://j3.rf-explorer.com/>
- [38] Mar 2023. [Online]. Available: <https://www.nist.gov/pml/time-and-frequency-division/popular-links/time-frequency-z/time-and-frequency-z-g>
- [39] S. J. Julier and J. K. Uhlmann, "New extension of the kalman filter to nonlinear systems," in *Signal processing, sensor fusion, and target recognition VI*, vol. 3068. Spie, 1997, pp. 182–193.
- [40] M. M. Ahsan, M. P. Mahmud, P. K. Saha, K. D. Gupta, and Z. Siddique, "Effect of data scaling methods on machine learning algorithms and model performance," *Technologies*, vol. 9, no. 3, p. 52, 2021.
- [41] L. Guo, L. Wang, C. Lin, J. Liu, B. Lu, J. Fang, Z. Liu, Z. Shan, J. Yang, and S. Guo, "Wiar: A public dataset for wifi-based activity recognition," *IEEE Access*, vol. 7, pp. 154 935–154 945, 2019.
- [42] B. Vidya and P. Sasikumar, "Wearable multi-sensor data fusion approach for human activity recognition using machine learning algorithms," *Sensors and Actuators A: Physical*, vol. 341, p. 113557, 2022.
- [43] W. Kong, L. He, and H. Wang, "Exploratory data analysis of human activity recognition based on smart phone," *IEEE Access*, vol. 9, pp. 73 355–73 364, 2021.
- [44] S. Daengduang and P. Vateekul, "Enhancing accuracy of multi-label classification by applying one-vs-one support vector machine," in *2016 13th International Joint Conference on Computer Science and Software Engineering (JCSSE)*. IEEE, 2016, pp. 1–6.
- [45] [Online]. Available: https://scikit-learn.org/stable/auto_examples/ensemble/plot_forest_importances.html
- [46] N. Ahmed, J. I. Rafiq, and M. R. Islam, "Enhanced human activity recognition based on smartphone sensor data using hybrid feature selection model," *Sensors*, vol. 20, no. 1, p. 317, 2020.
- [47] Y. Zheng, Y. Zhang, K. Qian, G. Zhang, Y. Liu, C. Wu, and Z. Yang, "Zero-effort cross-domain gesture recognition with wi-fi," in *Proceedings of the 17th annual international conference on mobile systems, applications, and services*, 2019, pp. 313–325.
- [48] O. Nafea, W. Abdul, G. Muhammad, and M. Alsulaiman, "Sensor-based human activity recognition with spatio-temporal deep learning," *Sensors*, vol. 21, no. 6, p. 2141, 2021.
- [49] "Location." [Online]. Available: <https://developer.android.com/reference/android/location/Location>
- [50] B. Zhou, Q. Li, G. Zhai, Q. Mao, J. Yang, W. Tu, W. Xue, and L. Chen, "A graph optimization-based indoor map construction method via crowdsourcing," *IEEE Access*, vol. 6, pp. 33 692–33 701, 2018.
- [51] [Online]. Available: https://en.wikipedia.org/wiki/Faraday_cage
- [52] A. Ranganathan, "The levenberg-marquardt algorithm," *Tutorial on LM algorithm*, vol. 11, no. 1, pp. 101–110, 2004.
- [53] B. Zhou, Q. Li, Q. Mao, W. Tu, X. Zhang, and L. Chen, "Alime: Activity landmark-based indoor mapping via crowdsourcing," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 5, pp. 2774–2785, 2015.
- [54] G. Shen, Z. Chen, P. Zhang, T. Moscibroda, and Y. Zhang, "{Walkie-Markie}: indoor pathway mapping made easy," in *10th USENIX symposium on networked systems design and implementation (NSDI 13)*, 2013, pp. 85–98.
- [55] "Request location permissions." [Online]. Available: <https://developer.android.com/develop/sensors-and-location/location/permissions>
- [56] "Declare permissions for your app." [Online]. Available: <https://support.google.com/googleplay/android-developer/answer/9214102?hl=en>
- [57] Appradar, "How to write perfect short & long descriptions in play store?" Jun 2023. [Online]. Available: <https://appradar.com/academy/short-long-app-description>
- [58] H. Ramirez, S. A. Velastin, I. Meza, E. Fabregas, D. Makris, and G. Farias, "Fall detection and activity recognition using human skeleton features," *Ieee Access*, vol. 9, pp. 33 532–33 542, 2021.
- [59] J. Wang, X. Zhang, Q. Gao, H. Yue, and H. Wang, "Device-free wireless localization and activity recognition: A deep learning approach," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 7, pp. 6258–6267, 2016.
- [60] Q. Gao, J. Wang, X. Ma, X. Feng, and H. Wang, "Csi-based device-free wireless localization and activity recognition using radio image features," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 11, pp. 10 346–10 356, 2017.
- [61] J. A. Chelli, A. A. Abdelgawwad, A. C. Mallofré, and M. Pätzold, "Wiwehar: Multimodal human activity recognition using wi-fi and wearable sensing modalities," *IEEE access*, vol. 8, pp. 164 453–164 470, 2020.
- [62] S. A. Bhat, A. Mehbodniya, A. E. Alwakeel, J. Webber, and K. Al-Begain, "Human motion patterns recognition based on rss and support vector machines," in *2020 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2020, pp. 1–6.
- [63] W. Shuaieb, G. Oguntala, A. AlAbdullah, H. Obeidat, R. Asif, R. A. Abd-Alhameed, M. S. Bin-Melha, and C. Kara-Zaitri, "Rfid rss fingerprinting system for wearable human activity recognition," *Future Internet*, vol. 12, no. 2, p. 33, 2020.
- [64] V. Bui, N. T. Le, T. L. Vu, V. H. Nguyen, and Y. M. Jang, "Gps-based indoor/outdoor detection scheme using machine learning techniques," *Applied Sciences*, vol. 10, no. 2, p. 500, 2020.
- [65] F. Zhu, K. Luo, X. Tao, and X. Zhang, "Deep learning based vehicle-mounted environmental context awareness via gnss signal," *IEEE Transactions on Intelligent Transportation Systems*, 2024.
- [66] S. Karam, G. Vosselman, and V. Lehtola, "Strategies to integrate imu and lidar slam for indoor mapping," in *XXIVth ISPRS Congress 2020*. Copernicus, 2020, pp. 223–230.
- [67] M. Alzantot and M. Youssef, "Crowdinside: Automatic construction of indoor floorplans," in *Proceedings of the 20th international conference on advances in geographic information systems*, 2012, pp. 99–108.
- [68] Y. Gu, Q. Song, M. Ma, Y. Li, and Z. Zhou, "Using ibeacons for trajectory initialization and calibration in foot-mounted inertial pedestrian positioning systems," in *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. IEEE, 2016, pp. 1–7.
- [69] D. Philipp, P. Baier, C. Dibak, F. Dürr, K. Rothermel, S. Becker, M. Peter, and D. Fritsch, "Mapgenie: Grammar-enhanced indoor map construction from crowd-sourced data," in *2014 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. IEEE, 2014, pp. 139–147.
- [70] C. X. Lu, S. Rosa, P. Zhao, B. Wang, C. Chen, J. A. Stankovic, N. Trigoni, and A. Markham, "See through smoke: robust indoor mapping with low-cost mmwave radar," in *Proceedings of the 18th International Conference on Mobile Systems, Applications, and Services*, 2020, pp. 14–27.
- [71] D. Plikynas, A. Žvironas, A. Budrionis, and M. Gudauskis, "Indoor navigation systems for visually impaired persons: Mapping the features of existing technologies to user needs," *Sensors*, vol. 20, no. 3, p. 636, 2020.
- [72] Y. Gu, C. Zhou, A. Wieser, and Z. Zhou, "Wifi based trajectory alignment, calibration and crowdsourced site survey using smart phones and foot-mounted imus," in *2017 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. IEEE, 2017, pp. 1–6.