On the Same Track: Privacy Leaks in Electric Rail Transport via Magnetic Fields

Stefan Czybik*
BIFOLD & TU Berlin
Berlin, Germany

Pia Hanfeld*
BIFOLD & TU Berlin
Berlin, Germany

Konrad Rieck
BIFOLD & TU Berlin
Berlin, Germany

Abstract

Inertial sensors of smartphones are a well-known privacy risk, as they allow inferring sensitive user movements and activity without explicit permissions. Among these sensors, the magnetometer has received little attention, as it is assumed to measure mainly the Earth's magnetic field. In this paper, we show that electric trains generate distinctive magnetic fields that can be captured by a smartphone magnetometer. Apps deployed on the device can access these signals without user permission, enabling two novel attacks on passenger privacy: First, we are able to determine whether the user is riding an electric train (transport mode detection). Second, we can infer whether other victims are aboard the same train due to shared patterns in the magnetic field (colocation prediction). In an empirical evaluation across 217 train rides, we demonstrate that both attacks are effective, confirming a real-world threat. We discuss countermeasures and argue that magnetometer access should require explicit user permission in mobile operating systems.

CCS Concepts

• **Security and privacy** → *Social aspects of security and privacy*;

Keywords

Mobile Privacy, Magnetometer, Colocation, Transport Mode Detection

ACM Reference Format:
Stefan Czybik, Pia Hanfeld, and Konrad Rieck. 2026. On the Same Track: Privacy Leaks in Electric Rail Transport via Magnetic Fields. In *Proceedings of the 2026 ACM SIGSAC Conference on Computer and Communications Security (CCS '26), November 15–19, 2026, The Hague, Netherlands.* ACM, New York, NY, USA, 15 pages. *https://doi.org/10.1145/3830454.3846554*

1 Introduction

Inertial sensors are an integral component of modern smartphones. Composed of an accelerometer, gyroscope, and magnetometer, these sensors continuously track the orientation and motion of the device during use. Such measurements are essential for a wide range of everyday functions, including screen rotation, step counting, gesture recognition, navigation assistance, and image stabilization [24]. Because these functions are pervasive across the mobile ecosystem, neither Android nor iOS protects inertial sensors behind explicit permissions, allowing any installed application to access their readings without notice [8]. This unrestricted access has repeatedly been shown to pose privacy risks.

Prior work has demonstrated that inertial sensors can reveal various sensitive information about the user [8], ranging from traveled routes [17, 23, 31] and transportation mode [11] to typed text [27, 34] and even spoken words [16, 28]. Within this body of work, however, the magnetometer has received comparatively little attention so far. It is commonly assumed to measure mainly the Earth's static magnetic field, providing a compass heading of limited use to an adversary. This is because, at first glance, other magnetic fields strong enough to be captured by a smartphone magnetometer seem rare in everyday environments.

In this paper, we challenge this assumption. We show that electric trains, powered by alternating current, generate distinctive magnetic fields that are measurable inside the passenger cabin. As electric current flows through the overhead power line and traction motors, it produces a magnetic emission that penetrates the train's metal shell into the passenger cabin. The emission varies with acceleration, route, and train load, forming a fingerprint of the ride. We find that this fingerprint can be captured by common smartphone magnetometers, opening a new channel for inferring sensitive information about passengers.

Building on this observation, we introduce two attacks on privacy. First, we present a new variant of *transport mode detection*: we determine whether the user is riding an electric train from the magnetometer alone. This is possible because the electric current oscillates at the rail grid frequency, producing a distinctive magnetic signature. Second, we present *colocation prediction*: by comparing magnetic field traces between smartphones, we are able to infer whether two users are traveling on the same train. The attack rests on the fact that passengers in the same train are immersed in the same magnetic field, so their traces share a synchronous pattern. Both attacks can be conducted covertly on Android and iOS, since magnetometer access is unrestricted on both platforms.

We empirically evaluate the attacks on a dataset of 217 train rides collected across diverse routes, covering long-distance and regional services. For transport mode detection, we achieve a Matthews correlation coefficient (MCC) score of 0.93 (F1: 0.98), on par with the state of the art. Our method, however, is notably simpler than prior work, as detecting the distinct frequency of the magnetic field requires little feature engineering. For colocation inference, we achieve an MCC score of 0.79 (F1: 0.90) using a learning-based approach, outperforming prior methods based on other sensors. Our results demonstrate that magnetometer-based inference is a real-world privacy threat, complementing existing attack vectors.



This work is licensed under a Creative Commons Attribution 4.0 International License.
This is a corrected version (v2) of the paper originally published in:
CCS '26, The Hague, Netherlands.
© 2026 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-2871-6/2026/11
https://doi.org/10.1145/3830454.3846554

*Authors contributed equally.

Both attacks apply specifically to electric trains operating on alternating current (AC) and thus not to diesel routes or DC-powered subway lines. However, AC traction dominates long-distance and regional rail across Europe and Asia, exposing a substantial passenger population to the threats we describe. More broadly, our findings add to the growing evidence that unrestricted access to inertial sensors continues to expose users to privacy threats. We discuss possible countermeasures for the magnetometer in this work, but ultimately join recent calls to systematically restrict access to these sensors in mobile operating systems.

In summary, we make the following contributions in this work:

- **Magnetic fields of electric trains.** We identify the magnetic field of electric trains as a previously overlooked privacy-relevant signal, readily captured by common smartphone magnetometers inside the passenger cabin.
- **Novel privacy attacks.** We present two novel attacks based on magnetic fields in trains that enable transport mode detection and colocation inference of passengers from the magnetometer alone.
- **Empirical evaluation and dataset.** We demonstrate the efficacy of both attacks in real-world conditions across 217 train rides, showing that electric train rides and jointly traveling passengers can be identified with high accuracy.

To foster research on privacy in rail transport, we publicly release our implementation along with our dataset, which to the best of our knowledge constitutes the largest collection of train rides with recorded inertial sensor data.

2 Magnetic Fields in Trains

Our attacks build on three pieces of background: the physics of magnetic fields, how electric trains produce distinctive magnetic emissions through their traction systems, and how smartphone magnetometers capture these signals.

2.1 Magnetic Fields

Magnetic fields arise from two fundamental sources: electric currents and the intrinsic magnetic moments of materials such as permanent magnets. The most familiar example in everyday life is the Earth's magnetic field, generated by electric currents in the planet's molten outer core. Near the surface, its magnitude is on the order of tens of microtesla, and it is typically treated as a static background field that supports compass-based navigation.

Beyond natural sources, magnetic fields are generated by electric currents, since every current-carrying conductor produces a field around it. Such fields are commonly classified as either static or alternating. Static fields are produced by direct current (DC) and remain constant over time. Alternating fields, in contrast, are generated by alternating current (AC) and change periodically with the current's frequency. In practice, magnetic fields produced by alternating currents are thus associated with a characteristic frequency, which simplifies their detection and analysis, as we discuss later.

Strong magnetic fields occur in systems with high operating currents. Examples include the electric power grid as well as electric railway and subway systems with overhead lines or conductor rails. Figure 1 illustrates a train powered by an overhead line. The traction current flowing through the line produces a magnetic field in the surrounding space, which can be measured by smartphone magnetometers inside the carriage.

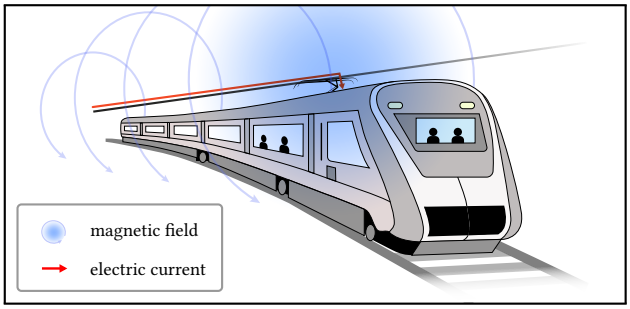


Figure 1: Train illustration with the magnetic field introduced by the current through the overhead line.

2.2 Electric Trains

On electrified railways, energy is usually supplied to trains via overhead contact lines and a pantograph mounted on the roof of the train. The pantograph maintains continuous mechanical contact with the overhead wire while the train is in motion. In a simplified model the electric current flows from the overhead line through the pantograph to the onboard traction units. As the current is typically on the order of hundreds of amperes, it produces significant magnetic fields in the vicinity. The same applies to alternative electrical power transmission schemes, such as the third rail system.

AC and DC traction systems. Electrified railway networks are widespread across Europe and Asia, while electrification in North America is restricted to selected commuter and intercity corridors. Several voltage and frequency standards coexist worldwide. Common AC systems are 15 kV at 16.7 Hz and 25 kV at 50 Hz. In addition, North America has 25 Hz systems, which are typically used for specific railways. DC systems operate primarily at 1.5 kV or 3 kV and are common on subway and tram networks.

For our work, we focus on 16.7 Hz AC systems for two reasons. First, this standard is used in Austria, Germany, Norway, Sweden, and Switzerland, covering most railways in central and northern Europe. Second, the resulting magnetic field carries a clear 16.7 Hz signature that can be identified directly from the grid frequency. While DC systems also produce magnetic fields, these are static and therefore difficult to distinguish from the Earth's static background.

Magnetic field from traction current. The traction current flowing through the overhead line produces a magnetic field that can be described using Ampere's law. In the simplified case of a long straight conductor, the magnetic flux density $B(t)$ is proportional to the electric current $I(t)$:

$$B(t) = \frac{\mu_0}{2\pi r} I(t) = \frac{\mu_0}{2\pi r} \frac{P(t)}{V} \tag{1}$$

where μ_0 is the magnetic permeability of free space and r is the distance to the current. Since the supply voltage V is constant within the system, the current I is proportional to the train's power

consumption P. Consequently, if we place a measurement device at a fixed distance from the conductor, the measured magnetic field amplitude directly reflects the train's instantaneous power consumption. Variations such as acceleration or changes in track gradient therefore produce characteristic time-dependent patterns in the magnetic field. This even holds on busy tracks: when other trains share the victim's overhead line, the train's movement still produces its own distinctive time-dependent pattern.

2.3 Smartphone Magnetometers

Since our attacks rely on measurements collected from consumer devices, we briefly review the magnetic sensing capabilities of modern smartphones and the constraints that apply to them. Smartphones are commonly equipped with three-axis micro-electromechanical system (MEMS) magnetometers, which capture the magnetic field vector along all spatial directions. These sensors are primarily intended for compass functionality and device orientation, but within their accuracy limits they can also be used to record other magnetic signals of interest, including those generated by power systems and electric railway infrastructure.

Sensor access and sampling rates. The sampling rate of smartphone magnetometers is constrained by both hardware limits and operating system policies. On Android, as of April 2026, access to magnetometer data does not require explicit runtime permissions. Since API level 31, however, the sampling rate for all inertial sensors is capped at 200 Hz[1]. This limit exceeds the maximum sampling capabilities of most commercially available magnetometer hardware. On iOS, currently sensor access is also possible but with an enforced maximum sampling rate of 100 Hz. Despite these differences, both platforms provide sufficient temporal resolution to capture slowly varying magnetic fields. This includes signals generated by power systems and electric railway infrastructure, such as the 16.7 Hz frequency used in overhead lines.

3 Privacy Leaks in Rail Transport

Having established the background, we now turn to the privacy leak resulting from magnetic fields in electric trains and show how usable signals can be extracted from the magnetometer.

3.1 Magnetic Field Signals

As discussed in Section 2, smartphone magnetometers can measure magnetic fields induced by electric currents. In particular, the data acquired from the magnetometer consists of three orthogonal components aligned with the device coordinate system. To obtain an orientation-independent representation of this data, we compute the Euclidean norm of the three axes. This resulting scalar signal reflects the total magnetic field magnitude and is independent of changes in device orientation. This normalization is necessary because the device can be moved arbitrarily during a train ride.

To illustrate the spectral characteristics of the measured magnetic fields, let us consider a spectrogram of the data. Figure 2 shows a spectrogram of the magnetic field magnitude for an electric train ride. The Earth's magnetic field appears as a low-frequency component at the bottom of the plot, while a distinct signal at 16.7 Hz corresponds to the electric current supplied via the overhead line. For other railway standards, this signal would appear at the respective grid frequency of 25 Hz or 50 Hz. Although this signal varies over time, it remains clearly visible throughout the entire recording. To work with it further, we apply a notch filter centered at the grid frequency. The filtered signal yields a time-domain trace for investigating changes over time.

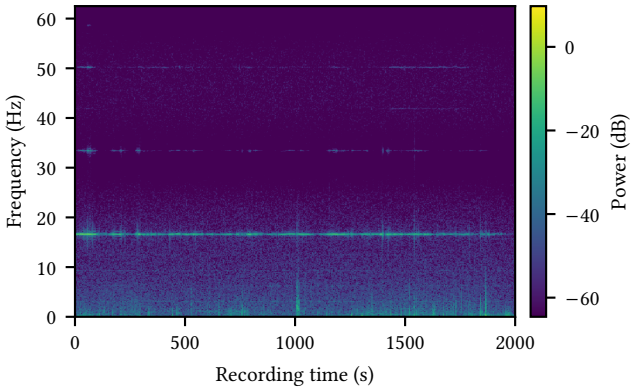


Figure 2: Spectrogram of the magnetometer of a train ride.

3.2 Magnetic Field Traces

Figure 3 shows the extracted trace from the train ride given in Figure 2 after filtering at 16.7 Hz and isolating the amplitude of the magnetic field. We observe distinct features of the train's movement: At the start, the signal fluctuates as the train moves out of the station. After around 300 s the train passes a switch between two tracks and follows one track until around 1300 s, where the next switch occurs. On the following straight section, the signal remains relatively stable, as train speed is constant and field variations are mainly caused by other trains drawing current from the same overhead line. After around 1800 s the train enters a busy rail network, where the signal shows more fluctuation, driven by the high density of other trains on the same overhead line.

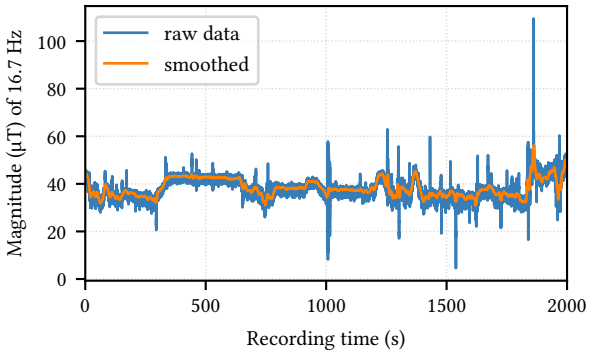


Figure 3: Signal of the magnetic field during a train ride.

[1] *https://developer.android.com/develop/sensors-and-location/sensors/sensors_overview?#sensors-rate-limiting*

To prepare a trace for comparison with others, we apply a high-pass filter to remove the static component and retain only the temporal changes in the signal. Since the amplitude depends on the distance to the overhead line, we normalize each trace by subtracting its mean and dividing by its standard deviation, making it independent of source distance and other attenuation factors. Finally, a smoothing step using a sliding window suppresses short-term noise, yielding a normalized trace that captures the characteristic current pattern of a train ride. This smoothed signal is shown as orange line in Figure 3.

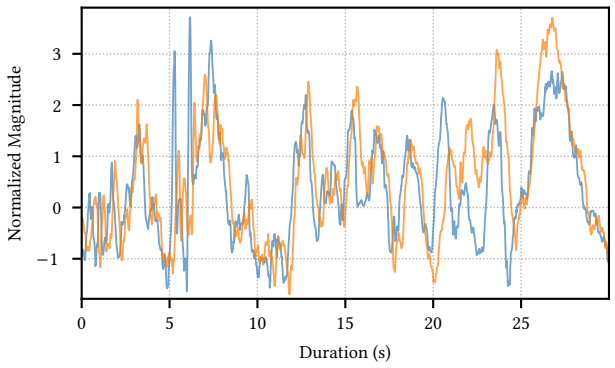


Figure 4: Signals of magnetic fields captured by magnetometers of two smartphones in the same train.

To illustrate the utility of this trace, let us examine the signals of two smartphones placed in different railcars of the same electric train. Figure 4 shows the respective traces. The recordings exhibit a strong similarity in their magnitude changes, despite minor differences. This shared structure arises because both devices are exposed to the same magnetic field during the train ride. We leverage these shared patterns to determine whether two smartphones are aboard the same train, that is, whether they are *colocated*.

At first glance, the signals shown in Figures 3 and 4 may seem to also indicate location-dependent features, such as particular switches on the route or the entering and exiting of rail networks. However, the trace does not primarily reflect the position of the recording train. The observed variations in the signal are influenced by the current train as well as all other trains connected to the same overhead line. Each of them draws or recuperates power depending on changes in speed, driving style, and traffic situation, so a share of the peaks stems from other trains accelerating or braking somewhere on the same feeding section. These events are time-dependent and vary from ride to ride, which means that the same route does not produce a stable and reproducible trace. Genuinely location-anchored features, such as section insulators or feeding points, do exist, but they are too sparse to compensate for the dominating temporal component. Consequently, the trace hinders precise localization or the identification of a particular train line. At the same time, it provides an accurate fingerprint of temporal events that all passengers of a train observe simultaneously, which is exactly what colocation inference requires.

4 Attack Scenarios

Equipped with an understanding of how information leaks from the observed magnetic fields in electric trains, we are now ready to derive privacy attacks and the underlying threat model.

4.1 Threat Model

The attacker's goal is to infer whether a victim is traveling by electric train and which victims are traveling together, without access to any location-related permission. We consider an attacker who is able to retrieve data recordings from applications running on a user's smartphone. As the magnetometer is not protected by permissions on the Android or iOS platform, any application can record magnetometer data without raising user suspicion. This covert monitoring can occur in different scenarios. First, the attacker may compromise a popular smartphone application through a third-party library and observe user movements on a large scale. For example, an advertisement provider might use the attack to track user habits unnoticed. Second, the adversary may target specific users through a compromised application tailored to their interests. For instance, an oppressive regime could monitor user movements via a tampered messaging application.

Attacker Capabilities. We assume that the attacker-controlled application can access magnetometer data without requesting runtime permissions. The application can record sensor data continuously in the background and transmit it to a server if required. We do not impose particular constraints on the computing costs for this; however, we later show that our attacks are efficient enough not to raise the user's suspicion.

In the following, we propose two attacks that can be combined into a two-stage approach. In the first stage, the attacker analyzes magnetometer data to detect characteristic frequency components induced by the train's power system, enabling transport mode detection (Section 4.2). This step can be performed locally on the device and filters the victim population down to users currently on an electric train. In the second stage, the attacker compares magnetic field traces across devices to identify shared patterns over time. These shared patterns enable colocation inference between users (Section 4.3), even in the absence of explicit location information. If a reference trace with known location is available, the inferred colocation directly exposes the location of additional users.

4.2 Transport Mode Detection

For transport mode detection (TMD), we do not require a complete amplitude trace of the magnetic field. Instead, it is sufficient to determine whether the frequency component of the overhead line is present in the magnetometer signal. As established in Section 3, this frequency is characteristic of electrified railway infrastructure and typically occurs only inside or close to a train. Detecting its presence therefore allows us to infer whether a victim user is currently traveling by an electric train.

Detection approach. To detect the presence of the target frequency, we require a method for identifying specific frequency components in a signal. Several techniques are available for this task, such as the discrete Fourier transform (DFT), power spectral density (PSD) estimation via Welch's method [44], and the Goertzel

algorithm [13]. Each method offers a different trade-off between implementation complexity, computational cost, frequency resolution, and robustness to noise. In our scenario, the main decision criterion is the signal-to-noise ratio (SNR) at the target frequency. The SNR compares the signal at the target frequency to an estimate of the background noise level derived from the surrounding spectrum. If the SNR exceeds a defined threshold, we can consider the frequency component *present* and the data is classified as a train ride.

When experimenting with different approaches, we found that the DFT-based method consistently achieves high detection performance while remaining computationally lightweight. We therefore adopt it as our method of choice. In particular, the normalized magnetometer signal is first detrended to remove its static component, then multiplied by a Hann window to reduce spectral leakage at bin boundaries. A DFT is applied to the windowed signal, and an SNR measure around the 16.7 Hz target frequency is computed. If the SNR exceeds a threshold τ, the window is classified as a train ride.

To further stabilize detections over time, we apply a majority vote over a sliding window of consecutive chunks of data, exploiting the fact that spontaneous alternating changes in transport mode are physically unlikely over short periods.

4.3 Colocation Prediction

For the prediction of colocation, we use the processed magnetometer signal trace described in Section 3. We consider two approaches for realizing the attack: a distance-based method that measures trace similarity directly, and a machine learning-based method that frames colocation as a classification problem.

Distance-based approach. A natural fit for inferring colocation of two devices is to quantify the similarity between two traces using a distance metric and apply a threshold: pairs with a distance below the threshold are classified as colocated, while pairs above it are not. Simple metrics such as Euclidean distance are limited in this setting, as they are not robust to minor temporal misalignments between traces. Alignment-aware distances are therefore preferred. Dynamic time warping (DTW) warps the time axes of both traces to find the non-linear alignment that minimizes the cumulative distance between matched points. Derivative dynamic time warping (DDTW) is a variant that runs DTW on the first-order derivatives of the traces, making the comparison sensitive to the shape rather than the level of the signal. Since our traces are already normalized and reduced to the 16.7 Hz component, the additional shape sensitivity of DDTW offers little benefit, while it adds a further step to every comparison.

We therefore adopt DTW as the trade-off between detection performance and computational cost. We extract segments of equal length from both traces, aligned to the same point in time using the device clocks. This ensures that both traces capture the same events on the train at the same time, maximizing the expected similarity. The comparison threshold is optimized on a set of recorded magnetometer data. Besides the threshold, we optimize the number of data points per second retained in the trace and temporal smoothing. While this approach is conceptually simple, the computational cost scales linearly with trace length and the number of comparisons—a bottleneck for 1:n inference scenarios.

Learning-based approach. For a learning-based approach to infer colocation, we draw on recent work in deep learning for time series analysis. In particular, we employ convolutional neural networks, which are well suited for our task as they can effectively capture local patterns and temporal dependencies [47]. We intentionally employ a lightweight network architecture to ensure that the inference process remains computationally lightweight, fulfilling a crucial requirement for seamless, real-time deployment.

During the training phase, the parameters of the convolutional neural network (CNN) are optimized using a comprehensive corpus of prerecorded signal pairs. We determine them in two steps. First, we derive the admissible range of each parameter from the physics of the signal, so that the resulting configurations remain able to represent the temporal variations of the traction current. Second, we search within these ranges for the values that perform best on the training data. Through this optimization, the network ultimately functions as a powerful, non-linear approximator, capable of directly classifying whether a given pair of recordings indicates device colocation.

Adopting this learning-based methodology yields two fundamental advantages over traditional distance-based comparisons. First, it completely shifts the burden of feature extraction from rigid mathematical metrics to the network itself. By learning directly from a training dataset, the model automatically identifies and isolates the specific, subtle signal variations that definitively indicate shared exposure to the same localized magnetic field. Second, the computational complexity of the evaluation is drastically reduced. Because classifying a pair of signal chunks requires only a single, efficient forward pass through the trained network, the comparison cost scales strictly linearly with the length of the data. This circumvents the quadratic scaling bottlenecks inherent to techniques like dynamic time warping, making the CNN approach exceptionally scalable for widespread inference tasks.

5 Evaluation Data

To evaluate both attacks, we collected a comprehensive dataset of smartphone sensor recordings from electric train rides, comprising detailed readings from all inertial sensors. While several public datasets exist, such as US-TMD [6], Collecty [10], MobilitApp [1], NOR-TMD [10], HTC [48], and SHL [12], none is directly suitable for our setting, as they do not distinguish between electric and non-electric trains.

5.1 Data Collection

For data collection, we recruited participants who regularly commute by train and provided them with a short set of instructions for collecting and labeling data. Recordings were made with the Sensor Logger app [7] at the highest sampling rate the device offered, typically 100 Hz. We recorded the magnetometer along with the accelerometer, gyroscope, gravity, linear acceleration, orientation and pressure whenever they were available. This broader sensor set was deliberately captured to make the data useful beyond the present work.

To create labels, participants noted each stop and departure together with the station name, the train line number, and the train type. The resulting labels divide each recording into trips and into

station-to-station segments, with the train type allowing us to separate electrified from non-electrified trains. We placed no restrictions on smartphone usage during the recordings. Participants used their devices as they normally would, including phone calls, messaging, and web browsing, as well as moving the device between pocket, hand, and table. The dataset therefore reflects realistic in-the-wild conditions rather than a controlled placement protocol.

5.2 Dataset Composition

The collected data contains 482.6 hours of labeled recordings. The data is organized hierarchically: a *journey* is a continuous recording session of a participant's travel, a *trip* is a single train ride between two changes within a journey, and a *segment* is a station-to-station segment within a trip. The released dataset comprises 217 journeys, 342 trips, and 1 979 segments, covering 246 unique stations. There are in total 357 segments with at least two colocated recordings in the same train.

Table 1: Distribution of train types within the collected data. # is the number of segments for the train type.

Type	All		16.7 Hz electric			
	hours	#	hours		#	
Long distance	414.7	844	402.6	97.1 %	822	97.4 %
Regional	44.1	342	37.0	83.9 %	279	81.6 %
Urban	19.7	586	3.7	18.9 %	88	15.0 %
Metro	4.2	201	0.0	0.0 %	0	0.0 %
Tram	0.1	6	0.0	0.0 %	0	0.0 %
Total	482.6	1 979	443.2	91.8 %	1 189	60.1 %

Table 1 shows the distribution of train types, both in recording time and in number of station-to-station segments. Long-distance trips dominate the dataset by recording time, while urban transport contributes a comparable number of shorter segments. The 16.7 Hz signature exploited by our attacks is present mainly in regional and long-distance trains; metros and trams typically use other power supply systems.

This dataset is comparable in size to other public datasets, yet it contains, to the best of our knowledge, the largest number of train rides available to date. Moreover, it is the only one collected with electrification labels.

5.3 Dataset Availability and Privacy

The original collected dataset comprises 137 GB and is available at: *https://doi.org/10.5281/zenodo.22206115*.

To protect participant privacy, the dataset is anonymized. Station names are replaced with random identifiers, with the same physical station mapped consistently across journeys. Absolute timestamps are removed and replaced with times relative to the start of each journey; the relative timing between journeys recorded simultaneously is preserved, so that colocated trips remain identifiable. Finally, in line with privacy regulations, all participants provided informed consent for the release of the dataset. Our research institution does not operate an Institutional Review Board (IRB), as is common at European institutions; we therefore followed the principles of the Menlo Report, as detailed in Section 11.

6 Evaluating Transport Mode Detection

We now evaluate our first attack: transport mode detection (TMD) in electric trains. In particular, after describing the experimental setup, we compare our approach against existing baselines and report its detection performance.

6.1 Experimental Setup

To evaluate our approach, we derive an evaluation dataset from our recordings; the preprocessing pipeline and implementation details are described below.

Data preprocessing. For this experiment, we consider all electric train rides operating on a 16.7 Hz grid frequency from our dataset. This includes all long-distance and regional trips. As negative samples, we additionally consider recordings collected during daily activities in which participants carried their smartphones outside of train rides. For these recordings, we leverage the Google Activity Recognition API[2] to infer the current activity, including classes such as Stationary, Walking, Cycling and Automotive. An overview of the data distribution is shown in Table 2. We perform the split between training and testing data at the journey level, assigning 30 % of the journeys to the test set.

Table 2: Overview of the data for transport mode detection. Activities are from Google's Activity Recognition API.

Activity	Time	Activity	Time
Stationary	107 925 s	Automotive	29 598 s
Walking	33 048 s	Tilting	38 s
Cycling	21 983 s	Unknown	89 935 s
No Train	282 527 s	Train	1 737 481 s

Implementation details. For signal processing an fast Fourier transform (FFT) is applied to the windowed signal, yielding a single spectrum over the full observation window with frequency resolution $\Delta f = f_s/N$, where f_s is the sampling rate and N is the number of samples. We then identify the peak amplitude within a narrow band of $\pm\delta_s$ Hz around the target frequency of 16.7 Hz and compute the SNR against the estimated noise floor. If the SNR exceeds a threshold τ, the window is classified as a train ride. Both δ_s and τ are treated as hyperparameters and optimized on the training set.

These free parameters are optimized with Optuna [2], a black-box hyperparameter optimization framework. The tuned parameters are the width of the signal band, δ_s, the SNR threshold τ, and the width of the noise band δ_n. The search ranges for these parameters are derived from the physics of the signal: the signal band δ_s has to be wide enough for the traction frequency and the resolution Δf of the transform, yet narrow enough to exclude neighboring spectral content, and the noise band δ_n has to cover enough bins for a stable noise floor estimate without reaching into the signal band. For noise floor estimation, we evaluate three strategies: the median of non-target bins, linear interpolation between neighboring bands, and a $1/f$ spline fit. We run the search for 5 000 trials, and each trial repeats the evaluation across 5 random journey-level splits of the

[2] *https://developers.google.com/location-context/activity-recognition*

training and test set to obtain a robust estimate of the objective. For each resulting detector, we then evaluate majority-vote aggregation over chunk lengths of $[1, 3, 6, 9, 12]$ seconds. We use the Matthews correlation coefficient (MCC) as our primary evaluation metric, as it is robust to class imbalance, and additionally report the F1 score for comparability with prior work on TMD.

6.2 Baseline Approaches

Over the past decades, TMD has been extensively studied, as discussed in detail in Section 9. In Table 3, we summarize all studies that, to the best of our knowledge, rely exclusively on permissionless smartphone sensors and include at least one class corresponding to train journeys.

Table 3: Related approaches and datasets. Sensors are Accelerometer (A), Gravity (G), Gyroscope (Y), Magnetometer (M) and Pressure (P). C is the number of TMD classes.

Study	Year	Dataset	C	Sensors
Hemminki et al. [15]	2013		7	A
Iskanderov and Guvensan [19]	2020	HTC	13	A, Y, M
Wang and Jiang [43]	2022	SHL, HTC	8	A, Y, M, P
Wang and Jiang [42]	2022	SHL, HTC	8	A, Y, M, P
Taherinavid et al. [41]	2024	HTC	8	A, G, Y, M
Xu [46]	2024	HTC	8	A, G, Y, M
Ibañez et al. [18]	2025	MobilitApp	11	A, Y, M
Skretting et al. [40]	2025	NOR-TMD	8	A, G, Y, M
Our study	2026	Ours	2	M

Previous work on TMD typically addresses multi-class classification, where the goal is to distinguish between several transport modes. This setting is inherently more challenging than our binary classification task. Nevertheless, these approaches provide a useful basis for comparison. To enable comparability, we extract the reported F1 and MCC scores from the respective studies. In cases where multiple train-related classes are present, such as *Train* and *High-Speed-Train*, we merge them into a single class. Although the approaches in Table 4 are evaluated on different datasets under varying conditions, they serve as a useful reference point for contextualizing the performance of existing methods.

6.3 Results

Table 4 shows the performance of our attack in comparison to results reported by prior work. Existing approaches for detecting train rides achieve F1 scores ranging from 0.70 to 0.93 on their respective datasets. All studies except Ibañez et al. [18] use 10 s chunks for classification; the latter does not report sufficient information to compute the MCC score. To reproduce the results of Skretting et al. [40] we reimplemented their approach, outlined in Appendix A.1. Despite relying on a single sensor and a single spectral feature, our TMD attack achieves an F1 score of 0.94 and an MCC of 0.83, placing it on par with state-of-the-art approaches. Applying a majority vote further improves performance to an F1 score of 0.98 and an MCC of 0.93, outperforming prior work in our evaluation setting.

Table 4: Performance of TMD for our attack and related work. Performance numbers are only shown for the "train" class of related approaches.

Study	Dataset	F1	MCC
Hemminki et al. [15]		0.70	0.68
Iskanderov and Guvensan [19]	HTC	0.83	0.79
Wang and Jiang [43]	SHL	0.74	0.72
Wang and Jiang [43]	HTC	0.79	0.77
Wang and Jiang [42]	SHL	0.74	0.73
Wang and Jiang [42]	HTC	0.83	0.82
Ibañez et al. [18]	MobilitApp	0.66	-
Skretting et al. [40]	NOR-TMD	0.93	0.93
Our attack w/o majority vote	Our dataset	0.94	0.83
Our attack w/ majority vote	Our dataset	**0.98**	**0.93**

Effect of data cardinality and granularity. Figure 5 shows how the total amount of data used for detection influences the MCC score of our attack. The x-axis represents the total data used, computed as the product of the number of chunks in the majority vote and the chunk duration. We observe that more data consistently improves detection across all post-departure windows. Whether this data is spread across a longer DFT window or across more majority-vote chunks has little influence on the result. With as little as 60 s of data, our detector already achieves an MCC of 0.91.

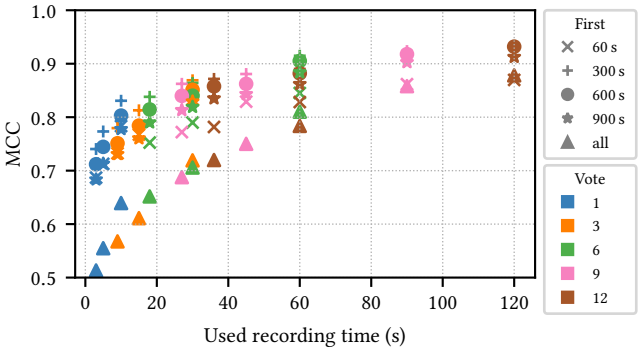


Figure 5: Influence of recording time on TMD performance with and without majority vote.

Effect of the post-departure window. The effect of recording length on performance is shown in Table 5. Across all majority-vote lengths, performance is highest when restricting the analysis to the start of the train ride, and degrades as longer segments than 600 s are included. The reason is physical: trains draw large traction currents during acceleration, which produce a strong 16.7 Hz component. Including later, low-signal segments dilutes the feature and reduces performance. Using all data further introduces a larger class imbalance, with fewer non-train chunks, which explains the gap between F1 and MCC in Table 4. This is why we adopt MCC as the primary metric throughout.

Table 5: Influence of post-departure window on TMD performance with majority voting as MCC score.

First (s)	Chunks for majority voting				
	1	3	6	9	12
60	0.78	0.83	0.85	0.86	0.87
300	0.83	0.87	0.91	0.92	**0.93**
600	0.80	0.85	0.91	0.92	**0.93**
900	0.78	0.83	0.88	0.90	0.91
All	0.64	0.72	0.81	0.86	0.88

Effect of majority voting. The majority vote provides a strong improvement at almost no additional cost. For 10 s chunks within the first 300 s after departure, a single decision yields 0.83, three votes yield 0.87, and twelve votes yield 0.93. The same trend holds in the 60 s window, where the MCC rises from 0.78 to 0.87 across the same vote lengths. Transport-mode changes occur on the order of minutes, so aggregating consecutive decisions removes spurious classifications without loss of temporal precision.

6.4 Summary

Our results show that electric train rides can be detected reliably from the smartphone magnetometer alone. The feature that carries the decision is the presence of the 16.7 Hz magnetic field generated by electrified railway infrastructure. Detecting this spectral component requires no learned features and no multi-sensor fusion. Our detector operates directly on the DFT and exposes only a few parameters.

The simplicity of the approach is matched by its data efficiency. A single 10 s chunk shortly after departure already reaches an MCC of 0.83, and a short majority vote over 120 s raises this to 0.93. This places our detector in the same performance range as prior multi-sensor pipelines, while keeping the computational footprint small enough for continuous background operation on a mobile device. The detector can also serve as a component within existing TMD systems. Many of these systems rely on machine-learning models trained on multiple sensors and struggle to separate trains from other smooth transport modes such as cars or buses. A dedicated indicator for the 16.7 Hz component can be added to such pipelines as an auxiliary feature, where it provides a physically grounded signal that is hard to confuse with non-rail transport.

7 Evaluation of Colocation Prediction

We continue to evaluate our attack for inferring the colocation of devices introduced in Section 3.2. After describing the experimental setup, we introduce our baseline and then evaluate the distance-based and learning-based variants of our approach. For both variants, we report detection performance as well as the amount of data required before a reliable decision can be made.

7.1 Experimental Setup

We evaluate the prediction of colocation on traces derived from our dataset, and describe the preprocessing, evaluation setup, and chunk aggregation below.

Data preprocessing. From the recordings dataset described in Section 5, we extract all trips that occur at least twice to form colocated pairs. For each trip, we derive the 16.7 Hz trace from the full sensor recording following the approach in Section 3. The resulting evaluation dataset consists of up to 348 trip segments with colocation. We preprocess each trace as described in Section 3.2, including normalization and downsampling to a fixed value of data points per second. This downsampling ensures consistent time alignment across all traces, reduces the input length to a constant size, and lowers the computational cost of the DTW distance calculation.

Evaluation Setup. The following setup applies to both the distance-based and the learning-based variants of our attack. We evaluate our approach using time windows from the train ride start of 60, 300, 600, and 900 seconds. For the learning-based variant, we additionally evaluate the full recordings. We omit this setting for the distance-based variant, due to the high computational costs and small interest. Our primary goal is to minimize the amount of data required for reliable inference, reducing both the time an attacker must wait before making a decision and the computational resources needed. In addition, we investigate smoothing the data using a sliding window to reduce noise and suppress short-term fluctuations in the signal.

To study the impact of different train types, we additionally construct separate datasets for short-distance and long-distance trains. We perform the train and test data split at the journey level, assigning 30 % of the journeys to the test set. Splitting at the journey level ensures that the traces from a journey are not split across training and evaluation data. To assess the stability of our results, we repeat each evaluation across 10 independent splits and report the mean. As evaluation metrics, we use MCC as our primary metric, since it is robust to the class imbalance in our dataset, and additionally report F1 for comparability with prior work.

Chunk Aggregation. The two variants process the chunks differently, which we briefly outline here and in detail in Sections 7.3 and 7.4. The distance-based variant concatenates all chunks within the post-departure window into a single trace and computes one distance value per trace pair. No per-chunk decision and no majority vote are applied. The learning-based variant operates on individual chunks during training and inference, producing a per-chunk decision. These per-chunk decisions are then aggregated via majority vote to obtain the final trace-level prediction.

7.2 Baseline

In contrast to TMD, only a limited number of studies address colocation prediction, most of which originate from pandemic tracing research, as discussed in Section 9. The work most closely related and comparable to ours is Nguyen et al. [32]. In that study, the authors use the magnitude of the magnetic field and the DDTW distance to infer colocation between devices.

Despite this conceptual similarity, there are important differences between their evaluation setting and ours. In our evaluation, we always compare traces of equal length, which removes trace length as a discriminative feature and makes the task substantially harder. When traces of different length are compared, the length itself already provides a strong cue, as colocated devices typically

share the same recording duration. By fixing the length across all pairs, the decision has to rely entirely on the shape of the signal rather than on trivial metadata.

Implementation Details. The baseline approach is conceptually simple: it operates on the magnitude of the magnetic field and applies DDTW as the distance measure. We therefore just calculated the Euclidean norm of the sensor's three axes as described in Section 3.1. The resulting traces are then directly applicable to the DDTW and we calculated the distance to all combinations of traces. On the training set we calculate a threshold, which then is used to evaluate the colocation on the testing set.

Results. We expect a noticeably lower performance than reported in the original study. The reason is twofold. First, the number of trace pairs in our evaluation grows quadratically with the size of the dataset, increasing the chance of spurious matches. Second, trace length, which is highly discriminative in their setting, is controlled for in ours. Under these conditions, the baseline has to rely solely on the signal shape, which is a considerably more demanding task. We report the resulting performance in Section 7.7 alongside our own approaches and a more detailed ablation study in Appendix B.1.

7.3 Distance-based Approach

As first approach to colocation prediction, we use a distance-based method that quantifies the similarity between two traces and applies a threshold to decide whether they originate from colocated devices.

Implementation Details. For each trace pair, we used two continuous traces, as described in Section 7.1. We then compute the DTW distance between these traces. We adopt a one-versus-all strategy in which each trace is compared against all other traces. A trace pair is classified as colocated if its distance falls below a decision threshold, and as non-colocated otherwise. The threshold is optimized on the training split together with the length of the post-departure window, the sampling rate of the underlying signal, and the temporal smoothing. The optimized threshold is then applied to the held-out test split to identify the smallest amount of data for a reliable colocation prediction.

Table 6: Distance-based colocation prediction performance as MCC score depending on data points per second and the length of the compared recording from the start.

First (s)	Data points per second			
	10	20	40	60
60	0.13±0.04	0.15±0.07	0.14±0.02	0.17±0.04
300	0.31±0.03	0.31±0.03	0.33±0.04	0.43±0.05
600	0.26±0.05	0.25±0.06	0.26±0.06	0.49±0.09
900	0.30±0.04	0.29±0.04	0.27±0.05	**0.60±0.11**

Results. Table 6 shows the MCC score of the distance-based approach using DTW. The best performance is achieved at a recording length of 900 s from the train ride start, 60 data points per second, and a sliding window of 150 samples for temporal smoothing, yielding an MCC score of 0.60. Shorter recordings reduce the amount of signal available for distance computation and lead to a noticeable drop in performance. At 60 s, the MCC decreases to 0.17, and even shorter recordings degrade further. Longer recordings have little influence on the MCC, except at 60 data points per second. A rate of 60 data points per second consistently outperforms lower rates, indicating that finer temporal resolution preserves discriminative details in the 16.7 Hz trace. However, the higher sampling rate produces a larger standard deviation across runs, indicating that the approach becomes less stable and reliable. Beyond the moderate detection performance, the distance-based approach is computationally more expensive, which we investigate in Section 7.6.

7.4 Learning-based Approach

For our learning-based approach, as introduced in Section 4.3, we utilize a Siamese 1D CNN with convolutional blocks consisting of a 1D convolutional layer, batch normalization, a ReLU activation, and a max pooling layer. Both recordings pass through the same convolutional blocks, so their feature sequences stay time-aligned and can be compared at each time step via their absolute difference and element-wise product. A further convolutional block encodes this comparison, followed by global average and max pooling, allowing the network to learn cross-trace features while encoding both signals identically. The pooled features are followed by fully-connected layers, which perform the actual binary classification.

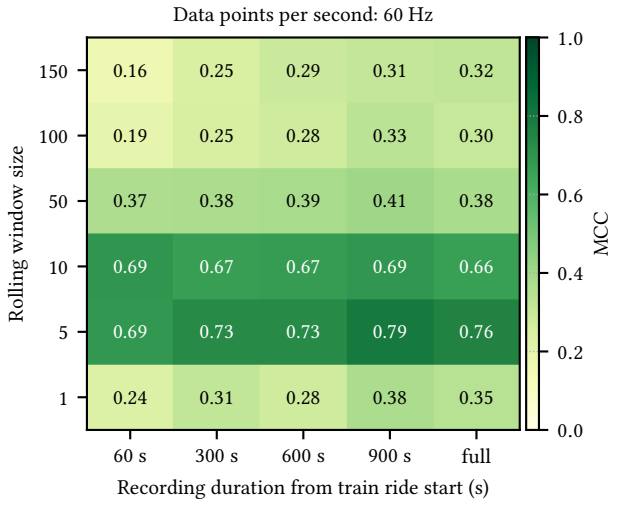


Figure 6: Learning-based colocation prediction performance as MCC score for different recording durations and rolling windows. The best MCC score is at a recording length of 900 s and a rolling window size of 5.

Implementation details. We train the CNN to predict the colocation of two chunks from two traces as described in Section 4.3. The model is trained using a binary cross-entropy loss function, which measures the discrepancy between the predicted value and the true label, namely if the pair is colocated or not. The optimization is performed using the Adam optimizer with a linear scheduler gradually increasing the learning rate for the first few epochs. The network

is implemented in PyTorch. Instead of drawing a single random pair from the training set as with the distance-based approach, we draw batches of pairs from the dataset uniformly at random. We trained our model on a NVIDIA A40 GPU for up to 1000 epochs on our training set, which took on average 30 minutes.

Hyperparameter search. In order to obtain an optimal configuration for the network architecture and subset of our dataset which yields the highest MCC for training, we used Optuna to search over 1 000 trials with a genetic sampling algorithm, i.e. the nondominated sorting genetic algorithm II (NSGA-II). We provide the results of this ablation study regarding network architecture and training hyperparameters in Appendix B.2. Training time varies slightly depending on the training set configuration, i.e. a higher sampling rate for longer recording durations leads to more combinations of pairs and, therefore, to increased training time.

Evaluation. Figure 6 presents the MCC scores across subsets of our evaluation dataset consisting of 60 s chunks, shown as multiple heatmaps over varying preprocessing parameters. The CNN is trained on one subset of our dataset, i.e. 60 s chunks recorded at 60 Hz with a rolling window size of 5 over the full recording length. As smoothing acts as a low-pass filter, each window size changes the frequency content of the trace, and the model performs well only on the frequency range it sees during training.

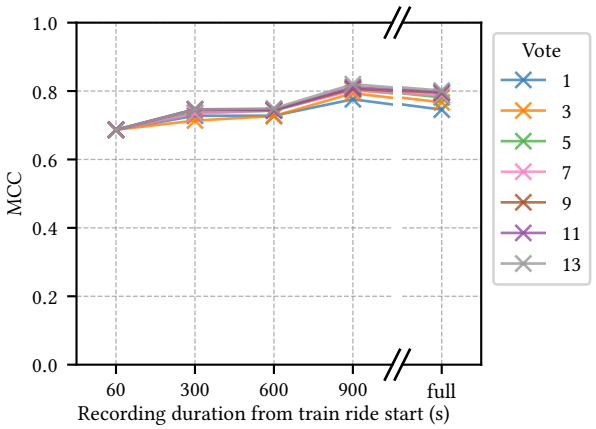


Figure 7: Learning-based colocation prediction evaluation with majority vote as MCC scores. Voting lengths k are odd numbers up to $k = 13$. From $k \geq 5$, the MCC score does not improve significantly.

Majority voting. We also analyze the impact of a majority vote (see Section 6.3) on colocated, consecutive k chunks from different devices. In Figure 7 we present the MCC scores for a chunk size of 60 s, a rolling window size of 5, and data points recorded at a sampling rate of 60 Hz. We can observe that the majority vote does not increase the MCC score significantly after $k \geq 5$. Nevertheless, given a larger recording duration, utilizing several consecutive chunks to predict colocation is beneficial. Especially due to the very fast inference time, which we discuss in the following.

7.5 Influence of the Train Type

Long-distance and regional services differ in speed, stopping frequency, and traction power. The magnetic trace can therefore carry a different amount of information for each of them. To test this, we repeat the evaluation of both variants on two subsets of our data, one with long-distance trips only and one with regional trips only. Table 7 reports the MCC of both variants.

The two subsets differ in size, and shrink with the length of the post-departure window. After 300 s, 197 colocated segments remain for long-distance trips, but only 39 for regional trips. Regional trains stop more frequently, so few of their segments run longer than five minutes. The regional subset falls to 10 segments after 600 s and to 2 after 900 s. We therefore report no regional scores for these two windows, as there are not enough pairs on which to base them.

Long-distance trips match the full dataset closely, as they dominate our dataset and thus drive its overall result. Regional trips reach higher scores in both variants, with 0.58±0.08 for the distance-based and 0.92 for the learning-based variant. We see two possible reasons for this gap. First, regional trains accelerate more often and more strongly, so each ride adds more characteristic events to the trace. Second, the regional subset is much smaller, and since we compare each trace against all others, it offers fewer candidate pairs and thus fewer chances for false matches. Given the large difference in subset size, we expect the second effect to dominate.

Table 7: Colocation inference performance (MCC score) of the distance-based and machine learning approaches across different train types and initial time windows. Best ML performances per duration are highlighted in bold.

Approach	First (s)	Train Type		
		All	Long distance	Regional
Distance-based	60	0.17±0.04	0.18±0.06	0.43±0.06
	300	0.43±0.05	0.40±0.11	0.58±0.08
	600	0.49±0.09	0.44±0.24	-
	900	0.60±0.11	0.58±0.14	-
Machine Learning	60	0.69	0.65	**0.92**
	300	0.73	0.70	**0.85**
	600	0.73	0.72	**0.91**
	900	0.79	**0.80**	0.74

Overall, colocation inference works for both service categories and is therefore of little concern for the attacker. The first five minutes of a ride are sufficient for both subsets to achieve a high success rate. Whether regional trains are in fact easier targets remains open, as the regional subset is too small to separate the effect of the train type from the effect of the subset size.

7.6 Runtime Analysis

To assess whether each attack is computationally feasible, we measure the runtime of a single comparison between two recordings. We evaluate all 114 colocation datasets, covering every combination of recording length and sampling rate, and measure 50 randomly selected recording pairs per dataset, repeated five times. Only the comparison itself is measured, and loading and preprocessing are excluded. To ensure a fair comparison, both approaches run on the same server using only the CPU, in our case an AMD EPYC 7713.

For the DTW-based approach a comparison is a single distance computation over the two recordings, whereas for the learning-based approach it is the classification of all chunk pairs of the two recordings followed by the majority vote, of which the vote accounts for at most 11 % of the runtime. The runtime of the learning-based approach is independent of the sampling rate, as the network resamples every chunk to a fixed input length. Classifying a single chunk pair with the CNN costs only 0.7 ms. The cost of a single comparison is therefore dominated by the fixed cost of invoking the network once, not by the number of chunks to classify. The classification of the whole trace grows with the recording length, from 3.2 ms to 26.3 ms. In contrast, the DTW-based approach scales linearly with both factors, ranging from 1.1 ms to 527.1 ms.

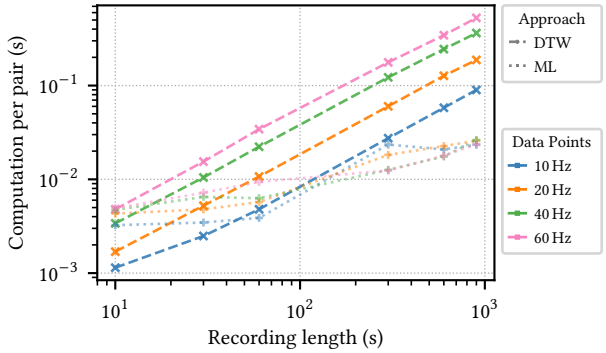


Figure 8: Runtime per comparison for the learning-based and distance-based colocation approaches.

Figure 8 illustrates the runtime of both approaches as a function of recording length. Averaged over the whole grid the learning-based approach is a factor of 9 faster than the DTW-based approach, rising to a factor of 22 for the longest recordings at the highest sampling rate. This advantage can be increased further by using GPUs or other AI accelerators for inference.

7.7 Summary

Table 8 summarizes the colocation prediction performance of the baseline, the distance-based approach, and the learning-based approach on our dataset. The baseline by Nguyen et al. [32] reaches an MCC of 0.09 on our dataset, which is barely above chance. This gap to the original study is consistent with the harder evaluation setting on traces sharing the same length introduced in Section 7.2. Our distance-based approach improves on the baseline by extracting the 16.7 Hz component before computing the DTW distance, reaching a moderate MCC of 0.60. The learning-based approach performs best, with an MCC of 0.79. Replacing the distance measurement with a CNN lifts the score by a clear margin, on both long-distance and regional services, as shown in Section 7.5. The model is trained on a single subset of our data, yet it holds this margin across all recording durations. It depends, however, on the smoothing used during training, so an attacker must preprocess new traces in the same way, which is easy to ensure in practice.

Both of our approaches reach their best score on the full length of the first 900 s of a ride. But already after five minutes of travel, the learning-based approach can link two devices with only 60 data points per second of the extracted trace. This rate refers to the trace, not to the raw magnetometer, which still has to be sampled fast enough to resolve the 16.7 Hz component.

Table 8: Colocation prediction performance of our attack and the related approach by Nguyen et al.

Approach	First	F1	MCC
Nguyen et al. [32] reimplemented baseline DDTW on magnitude	60 s	0.02±0.01	0.09±0.05
	300 s	0.02±0.03	0.06±0.08
	600 s	0.02±0.01	0.01±0.02
	900 s	0.02±0.01	0.02±0.03
Our attack with DTW distance on full 16.7 Hz trace	60 s	0.13±0.05	0.17±0.04
	300 s	0.38±0.05	0.43±0.05
	600 s	0.42±0.11	0.49±0.09
	900 s	0.56±0.12	0.60±0.11
Our attack with CNN model on 60 s chunks of 16.7 Hz trace	60 s	0.84	0.69
	300 s	0.85	0.73
	600 s	0.86	0.73
	900 s	**0.90**	**0.79**

The two approaches also differ in computational cost. A single pairwise DTW computation in the best configuration of our distance-based approach takes about 200 ms on our system, while the learning-based approach needs below 10 ms for the same decision. The runtime of DTW additionally grows with both the rate and the length of the recording, while the runtime of the CNN is independent. Since the one-versus-all evaluation requires a quadratic number of such comparisons, this difference decides whether the attack scales to the 1:n setting of our threat model. Overall, the learning-based approach is the only variant that combines high detection performance, robustness across train types, and the computational efficiency required for realistic large-scale deployment.

8 Discussion

We now place our findings in context and discuss what they mean for smartphone users. The discussion is organized in four parts. We first describe the privacy implications of our attacks, then state the assumptions under which they hold. We then propose recommendations for mobile operating systems and close with a note on beneficial applications of the same signal.

8.1 Implications

Our results show that the smartphone magnetometer is a privacy-relevant sensor on electrified trains. With our TMD approach, an attacker can reliably tell whether a victim is on a train. Once a train ride is detected, the same attacker can compare magnetic traces between devices and infer whether two victims share the same train. Together, these two capabilities expose travel patterns and social proximity without any explicit location permission. This places the magnetometer alongside the accelerometer and gyroscope as a sensor that warrants careful treatment in the mobile threat model. The population most exposed by our specific instantiation are commuters in Central Europe, where the 16.7 Hz traction system is widespread and where train travel is part of daily routines.

8.2 Assumptions

Our attack rests on a small set of physical and technical assumptions. First, the attack only applies to electrified trains and not to other modes of transport. Second, it requires a railway network with a traction frequency that is extractable from a smartphone magnetometer. Most current smartphones sample the magnetometer at 100 Hz, and some at 125 Hz. Both support extraction of 25 Hz traction signals, and we see no physical barrier to transferability in such networks. As sensor hardware improves, reliable measurement of 50 Hz components may become possible, which would extend our approach to a large part of the world's rail network. However, 50 Hz often coincides with the public power grid frequency, so the signal is no longer characteristic of railways alone. In its current form, the geographic confinement of 16.7 Hz systems is itself an information leak, as the presence of this frequency in a trace reveals the continent on which the victim is located.

8.3 Recommendations

Two complementary mitigations are available at the operating system level. The first is a stricter access control model for motion and environmental sensors. At present, the magnetometer can typically be read by any application without explicit user permission, which enables continuous background recording. As already suggested by others [30, 35], introducing a dedicated permission for the magnetometer and similar sensors would increase transparency and let users decide which applications may collect such data. This would also make malicious applications easier to identify.

The second mitigation is a cap on the sampling rate exposed to applications. Our attack relies on a stable 16.7 Hz component, which requires a temporal resolution of at least 35 Hz in the recorded signal. If the operating system limits the sampling rate available to applications below this value, frequency extraction is no longer possible. Such a cap would directly remove the basis of both the TMD and the colocation inference, but may also influence other features like step counting or gesture detection.

The two mitigations can be combined, by tying the available sampling rate to a permission tier. A less restrictive variant is to require a special permission only when an application requests sampling rates that enable detailed spectral analysis. This keeps benign applications working while denying malicious ones the resolution needed for frequency-based inference.

8.4 Beneficial Applications

The same signal that enables the attack can also support useful features. Our frequency extraction provides a lightweight indicator for train rides that can be embedded into existing TMD pipelines on smartphones, as current implementations have difficulties to distinguish between trains and cars (see Section 9). Most modern operating systems already offer activity recognition, and a robust train indicator would complement these features in a privacy-preserving and responsible deployment. A second class of applications is colocation-based contact tracing. During the COVID-19 pandemic, many contact tracing approaches were deployed and evaluated [36]. The proposal of Nguyen et al. [32] aligns with our setting, which makes colocation inference based on magnetic fields a candidate signal for contact tracing within trains.

9 Related work

Having presented our two attacks, we now place them in the context of prior work along two axes: transport mode detection and colocation inference. We focus on studies that, like ours, rely on permissionless smartphone sensors, since this is the property that makes the attacks of practical concern.

Transport mode detection. Research on TMD has grown in step with the spread of smartphones and can be organized along three axes: the sensors used, the set of inferred classes, and the inference method. Since our approach uses permissionless sensors and considers train journeys as a transport mode, we restrict our discussion to related work that meets these requirements. Broader surveys are provided by Sadeghian et al. [39] and Gong et al. [14]. A range of further studies relies on protected sensor data such as GPS, Wi-Fi, sound or camera input and falls outside our scope, but we list them for completeness [3–5, 9, 11, 25, 26, 29, 33, 37, 38, 45].

Hemminki et al. [15] propose the first approach that relies solely on the accelerometer for TMD. They show the feasibility of distinguishing the modes *Still*, *Walking*, *Bus*, *Train*, *Metro* and *Tram*, and report an F1 score of 0.70 for the *Train* class on their own dataset. Iskanderov and Guvensan [19] extend the set of classes to *Running*, *Stair Up*, *Stair Down*, *Bus*, *Motorcycle*, *Car*, *Bike*, and *High-Speed-Train*, using a long short-term memory (LSTM) model that reaches an F1 score of 0.83 for the combined *Train* and *High-Speed-Train* class on the HTC dataset. As a follow-up, Wang and Jiang [42, 43] propose two machine learning approaches with a smaller set of 8 classes, reaching F1 scores of 0.79 and 0.83 on HTC and of 0.74 on SHL. Similarly, Xu [46] and Taherinavid et al. [41] apply machine learning on HTC to infer 11 transport modes. Ibañez et al. [18] combine CNNs with an LSTM on their own dataset to infer 12 classes, achieving an F1 score of 0.66 for train rides. On the NOR-TMD dataset, Skretting et al. [40] evaluate feature importance across multiple sensors and statistical descriptors with several machine learning methods up to an F1 score of 0.93.

In contrast to these approaches, we address a binary decision rather than a multi-class problem and rely on the magnetometer alone. This narrower setting is enabled by the 16.7 Hz trace of electric trains and yields the F1 scores reported in Section 6.3 with a much simpler model and shorter recordings.

Inferring colocation. In contrast to TMD, only a few studies on colocation rely on permissionless smartphone sensors, and most of them are motivated by contact tracing during epidemics. The existing work can be divided into approaches that use machine learning and approaches that do not. Without machine learning, Nguyen et al. [32] propose an approach that uses the smartphone magnetometer to record the amplitude of the ambient magnetic field on London's public transport network. Pairs of recordings are compared with the DDTW distance, and a threshold on this distance decides whether two devices were on the same vehicle. Jeong et al. [20] address indoor colocation with the same sensor and also use traces of the magnetic field magnitude, but compare them with the Pearson correlation coefficient. Among learning-based approaches, Kosasih et al. [21, 22] propose a convolutional autoencoder model evaluated on a dataset of about 24 minutes [22], reporting F1 scores between 0.65 and 0.89 [21].

All of these approaches operate on the raw amplitude or magnitude of the magnetic field, in either general transport or indoor settings. Our work differs in two respects: we isolate the 16.7 Hz traction signal that is specific to electric rail transport, and we evaluate colocation on hundreds of real-world train rides rather than on minutes of recordings.

Across both threads of research, prior work has not treated permissionless magnetometer access as a privacy attack surface in the rail setting. Our paper closes this gap by showing that a single, narrow frequency component of the magnetometer signal is sufficient for both transport mode detection and colocation prediction under realistic conditions.

10 Conclusion

In this paper, we show that the magnetometer in smartphones leaks information about a user's train ride. The traction current in the overhead line of an electric train produces a distinctive magnetic field at the grid frequency, which is measurable in the carriage and is captured by common smartphone sensors. From this signal, we derive two attacks on passenger privacy that operate without any explicit permission on Android and iOS: transport mode detection and colocation inference.

We evaluate both attacks on a dataset of 217 train rides collected under realistic, in-the-wild conditions. For transport mode detection, a single spectral feature at 16.7 Hz is sufficient to reach an F1 score of 0.98 and an MCC of 0.93, on par with multi-sensor pipelines from prior work. For colocation inference, a lightweight convolutional model on the extracted magnetic trace reaches an F1 score of 0.90 and an MCC of 0.79, outperforming prior magnetometer-based approaches by a clear margin. These results add to the growing body of evidence that unrestricted access to inertial sensors exposes smartphone users to privacy leaks. We argue that these sensors should no longer be treated as harmless and mobile operating systems should place their access behind an explicit permission.

Acknowledgments

We would like to thank all contributors to the dataset, Elias Burggraef for reimplementing the NOR-TMD baseline and Stef Lenk for creating Figure 1. This work was supported by the European Research Council (ERC) under the consolidator grant MALFOY (101043410) and the German Federal Ministry of Research, Technology and Space (BMFTR) under the grant "AIgenCY" (16KIS2012).

References

[1] Mónica Aguilar Igartua, Gerard Caravaca Ibáñez, and Miquel Gotanegra Estañol. 2024. MobilitApp - Mobility sensor dataset. doi:*10.34810/DATA1948*

[2] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A Next-generation Hyperparameter Optimization Framework. In *Proc. of the 25th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD) (KDD '19)*. ACM, New York, NY, USA, 2623–2631. doi:*10.1145/3292500.3330701*

[3] Zahra Ansarilari and Amir Golroo. 2015. Automated Transportation Mode Detection Using Smart Phone Applications via Machine Learning. In *Proc. of the 94th Transportation Research Board Annual Meeting (TRB)*. Transportation Research Board, Washington, DC, USA. doi:*10.3929/ethz-b-000722728*

[4] Huthaifa I. Ashqar, Mohammed H. Almannaa, Mohammed Elhenawy, Hesham A. Rakha, and Leanna House. 2019. Smartphone Transportation Mode Recognition Using a Hierarchical Machine Learning Classifier and Pooled Features From Time and Frequency Domains. *IEEE Transactions on Intelligent Transportation Systems* 20, 1 (2019), 244–252. doi:*10.1109/tits.2018.2817658*

[5] Azwad Aziz, Nafisa Khan Youkee, Fariha Shams Ahmed, Sandia Tasnim, Md. Fahim-Ul-Islam, and Amitabha Chakrabarty. 2024. Enhancing Transparency in Transport Mode Detection: An Interpretable Ensemble Model Classifier. In *Proc. of the 2024 International Conference on Advances in Computing, Communication, Electrical, and Smart Systems (iCACCESS)*. IEEE, Piscataway, NJ, USA, 1–6. doi:*10.1109/iCACCESS61735.2024.10499511*

[6] Claudia Carpineti, Vincenzo Lomonaco, Luca Bedogni, Marco Di Felice, and Luciano Bononi. 2018. Custom Dual Transportation Mode Detection By Smartphone Devices Exploiting Sensor Diversity. In *Proc. of the 2018 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. IEEE, Piscataway, NJ, USA, 367–372. doi:*10.1109/PERCOMW.2018.8480119*

[7] Kelvin Tsz Hei Choi. 2024. *Sensor Logger*. doi:*10.5281/zenodo.20277979*

[8] Paula Delgado-Santos, Giuseppe Stragapede, Ruben Tolosana, Richard Guest, Farzin Deravi, and Ruben Vera-Rodriguez. 2022. A Survey of Privacy Vulnerabilities of Mobile Device Sensors. *Comput. Surveys* 54, 11s (Jan. 2022), 1–30. doi:*10.1145/3510579*

[9] Ifigenia Drosouli, Athanasios Voulodimos, Georgios Miaoulis, Paris Mastorocostas, and Djamchid Ghazanfarpour. 2021. Transportation Mode Detection Using an Optimized Long Short-Term Memory Model on Multimodal Sensor Data. *Entropy* 23, 11 (Nov. 2021), 1457. doi:*10.3390/e23111457*

[10] Martina Erdelić, Tomislav Erdelić, and Tonči Carić. 2023. Dataset for multimodal transport analytics of smartphone users - Collecty. *Data in Brief* 50 (2023), 109481. doi:*10.1016/j.dib.2023.109481*

[11] Tao Feng and Harry J.P. Timmermans. 2013. Transportation Mode Recognition Using GPS and Accelerometer Data. *Transportation Research Part C: Emerging Technologies* 37 (Dec. 2013), 118–130. doi:*10.1016/j.trc.2013.09.014*

[12] Hristijan Gjoreski, Mathias Ciliberto, Lin Wang, Francisco Javier Ordoñez Morales, Sami Mekki, Stefan Valentin, and Daniel Roggen. 2018. Sussex-Huawei Locomotion and Transportation Dataset. doi:*10.21227/7vtt-8c19*

[13] Gerald Goertzel. 1958. An Algorithm for the Evaluation of Finite Trigonometric Series. *The American Mathematical Monthly* 65, 1 (Jan. 1958), 34. jstor:2310304 doi:*10.2307/2310304*

[14] Lei Gong, Takayuki Morikawa, Toshiyuki Yamamoto, and Hitomi Sato. 2014. Deriving Personal Trip Data from GPS Data: A Literature Review on the Existing Methodologies. *Procedia - Social and Behavioral Sciences* 138 (July 2014), 557–565. doi:*10.1016/j.sbspro.2014.07.239*

[15] Samuli Hemminki, Petteri Nurmi, and Sasu Tarkoma. 2013. Accelerometer-Based Transportation Mode Detection on Smartphones. In *Proc. of the 11th ACM Conference on Embedded Networked Sensor Systems (SenSys)*. ACM, New York, NY, USA, 1–14. doi:*10.1145/2517351.2517367*

[16] Pengfei Hu, Hui Zhuang, Panneer Selvam Santhalingam, Riccardo Spolaor, Parth Pathak, Guoming Zhang, and Xiuzhen Cheng. 2022. AccEar: Accelerometer Acoustic Eavesdropping with Unconstrained Vocabulary. In *Proc. of the 2022 IEEE Symposium on Security and Privacy (SP)*. IEEE, Piscataway, NJ, USA, 1757–1773. doi:*10.1109/SP46214.2022.9833716*

[17] Jingyu Hua, Zhenyu Shen, and Sheng Zhong. 2017. We Can Track You If You Take the Metro: Tracking Metro Riders Using Accelerometers on Smartphones. *IEEE Transactions on Information Forensics and Security* 12, 2 (Feb. 2017), 286–297. doi:*10.1109/tifs.2016.2611489*

[18] Gerard Caravaca Ibañez, Luis J. De La Cruz Llopis, Adrian Catalín Diaconeasa, Alberto Bazán Guillén, and Mónica Aguilar Igartua. 2025. MobilitApp: A Deep Learning-Based Tool for Transport Mode Detection to Support Sustainable Urban Mobility. *IEEE Access* 13 (2025), 68439–68461. doi:*10.1109/access.2025.3561238*

[19] Jemshit Iskanderov and M. Amac Guvensan. 2020. Breaking the Limits of Transportation Mode Detection: Applying Deep Learning Approach With Knowledge-Based Features. *IEEE Sensors Journal* 20, 21 (Nov. 2020), 12871–12884. doi:*10.1109/jsen.2020.3001803*

[20] Seungyeon Jeong, Seungho Kuk, and Hyogon Kim. 2019. A Smartphone Magnetometer-Based Diagnostic Test for Automatic Contact Tracing in Infectious Disease Epidemics. *IEEE Access* 7 (2019), 20734–20747. doi:*10.1109/access.2019.2895075*

[21] David Ishak Kosasih, Byung-Gook Lee, Hyotaek Lim, and Mohammed Atiquzzaman. 2021. An Unsupervised Learning-Based Spatial Co-Location Detection System from Low-Power Consumption Sensor. *Sensors* 21, 14 (July 2021), 4773. doi:*10.3390/s21144773*

[22] David Ishak Kosasih, Hyotaek Lim, and Lee Byung Gook. 2020. Co-location Detection System Using Mobile Magnetometer Data. In *Proc. of the 2020 International Conference on Information and Communication Technology Convergence (ICTC)*. IEEE, Piscataway, NJ, USA, 1833–1838. doi:*10.1109/ICTC49870.2020.9289619*

[23] Anne Josiane Kouam, Stefan Czybik, and Konrad Rieck. 2026. On the Feasibility of Reference-Free Tracking via Motion Sensors in Public Transport. In *Proc. of the 28th International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM)*. ACM, New York, NY, USA.

[24] Nicholas D. Lane, Emiliano Miluzzo, Hong Lu, Daniel Peebles, Tanzeem Choudhury, and Andrew T. Campbell. 2010. A survey of mobile phone sensing. *IEEE Communications Magazine* 48, 9 (2010), 140–150. doi:*10.1109/mcom.2010.5560598*

[25] Ajaykumar Manivannan, Elias J. Willemse, Balamurali B. T, Wei Chien Benny Chin, Yuren Zhou, Bige Tuncer, Alain Barrat, and Roland Bouffanais. 2022. A

Framework for the Identification of Human Vertical Displacement Activity Based on Multi-Sensor Data. *IEEE Sensors Journal* 22, 8 (April 2022), 8011–8029. doi:*10.1109/jsen.2022.3157806*

[26] Vincenzo Manzoni, Diego Maniloff, Kristian Kloeckl, and Carlo Ratti. 2011. *Transportation Mode Identification and Real-Time CO2 Emission Estimation Using Smartphones.* Technical Report.

[27] Philip Marquardt, Arunabh Verma, Henry Carter, and Patrick Traynor. 2011. (sp)iPhone: decoding vibrations from nearby keyboards using mobile phone accelerometers. In *Proc. of the 18th ACM SIGSAC Conference on Computer and Communications Security (CCS) (CCS '11).* ACM, New York, NY, USA, 551–562. doi:*10.1145/2046707.2046771*

[28] Yan Michalevsky, Dan Boneh, and Gabi Nakibly. 2014. Gyrophone: recognizing speech from gyroscope signals. In *Proc. of the 23rd USENIX Conference on Security Symposium (SEC'14).* USENIX Association, USA, 1053–1067.

[29] Emiliano Miluzzo, Nicholas D. Lane, Kristóf Fodor, Ronald Peterson, Hong Lu, Mirco Musolesi, Shane B. Eisenman, Xiao Zheng, and Andrew T. Campbell. 2008. Sensing Meets Mobile Social Networks: The Design, Implementation and Evaluation of the CenceMe Application. In *Proc. of the 6th ACM Conference on Embedded Networked Sensor Systems (SenSys).* ACM, New York, NY, USA, 337–350. doi:*10.1145/1460412.1460445*

[30] Reham Mohamed, Habiba Farrukh, Yidong Lu, He Wang, and Z. Berkay Celik. 2023. iSTELAN: Disclosing Sensitive User Information by Mobile Magnetometer from Finger Touches. *Proceedings on Privacy Enhancing Technologies* 2023, 2 (April 2023), 79–96. doi:*10.56553/popets-2023-0042*

[31] Sashank Narain, Triet D. Vo-Huu, Kenneth Block, and Guevara Noubir. 2016. Inferring User Routes and Locations Using Zero-Permission Mobile Sensors. In *Proc. of the 2016 IEEE Symposium on Security and Privacy (SP).* IEEE, Piscataway, NJ, USA, 397–413. doi:*10.1109/SP.2016.31*

[32] Khuong An Nguyen, Chris Watkins, and Zhiyuan Luo. 2017. Co-location epidemic tracking on London public transports using low power mobile magnetometer. In *Proc. of the 2017 International Conference on Indoor Positioning and Indoor Navigation (IPIN).* IEEE, Piscataway, NJ, USA, 1–8. doi:*10.1109/IPIN.2017.8115963*

[33] Philippe Nitsche, Peter Widhalm, Simon Breuss, Norbert Brändle, and Peter Maurer. 2014. Supporting Large-Scale Travel Surveys with Smartphones – A Practical Approach. *Transportation Research Part C: Emerging Technologies* 43 (June 2014), 212–221. doi:*10.1016/j.trc.2013.11.005*

[34] Emmanuel Owusu, Jun Han, Sauvik Das, Adrian Perrig, and Joy Zhang. 2012. ACCessory: password inference using accelerometers on smartphones. In *Proc. of the Workshop on Mobile Computing Systems and Applications (HotMobile) (HotMobile '12).* ACM, New York, NY, USA, Article 9, 6 pages. doi:*10.1145/2162081.2162095*

[35] Giuseppe Petracca, Yuqiong Sun, Ahmad-Atamli Reineh, Patrick McDaniel, Jens Grossklags, and Trent Jaeger. 2019. Entrust: regulating sensor access by cooperating programs via delegation graphs. In *Proc. of the 28th USENIX Conference on Security Symposium (SEC'19).* USENIX Association, USA, 567–584.

[36] Ha-Linh Quach, Erin I. Walsh, Thi Ngoc Anh Hoang, Richard Norman Leslie Terrett, and Florian Vogt. 2025. Effectiveness of Digital Contact Tracing Interventions for COVID-19: A Systematic Scoping Review. *Public Health* 242 (May 2025), 146–156. doi:*10.1016/j.puhe.2025.02.019*

[37] Sasank Reddy, Min Mun, Jeff Burke, Deborah Estrin, Mark Hansen, and Mani Srivastava. 2010. Using Mobile Phones to Determine Transportation Modes. *ACM Transactions on Sensor Networks* 6, 2 (Feb. 2010), 1–27. doi:*10.1145/1689239.1689243*

[38] Sebastien Richoz, Lin Wang, Philip Birch, and Daniel Roggen. 2020. Transportation Mode Recognition Fusing Wearable Motion, Sound, and Vision Sensors. *IEEE Sensors Journal* 20, 16 (2020), 9314–9328. doi:*10.1109/JSEN.2020.2987306*

[39] Paria Sadeghian, Johan Håkansson, and Xiaoyun Zhao. 2021. Review and Evaluation of Methods in Transport Mode Detection Based on GPS Tracking Data. *Journal of Traffic and Transportation Engineering (English Edition)* 8, 4 (Aug. 2021), 467–482. doi:*10.1016/j.jtte.2021.04.004*

[40] Anders Skretting, Tor-Morten Grønli, and Raghava Rao Mukkamala. 2025. Feature Importance from Novel Dataset for On-Device Transport Mode Detection. *Journal of Intelligent Transportation Systems* 30, 4 (June 2025), 1–22. doi:*10.1080/15472450.2025.2515567*

[41] Siavash Taherinavid, Seyed Vahid Moravvej, Yen-Lin Chen, Jing Yang, Chin Soon Ku, and Por Lip Yee. 2024. Automatic Transportation Mode Classification Using a Deep Reinforcement Learning Approach With Smartphone Sensors. *IEEE Access* 12 (2024), 514–533. doi:*10.1109/access.2023.3346875*

[42] Pu Wang and Yongguo Jiang. 2022. DFTrans: Dual Frequency Temporal Attention Mechanism-Based Transportation Mode Detection. *Sensors* 22, 21 (Nov. 2022), 8499. doi:*10.3390/s22218499*

[43] Pu Wang and Yongguo Jiang. 2022. Transportation Mode Detection Using Temporal Convolutional Networks Based on Sensors Integrated into Smartphones. *Sensors* 22, 17 (Sept. 2022), 6712. doi:*10.3390/s22176712*

[44] P. Welch. 1967. The Use of Fast Fourier Transform for the Estimation of Power Spectra: A Method Based on Time Averaging over Short, Modified Periodograms. *IEEE Transactions on Audio and Electroacoustics* 15, 2 (June 1967), 70–73. doi:*10.1109/tau.1967.1161901*

[45] Peter Widhalm, Philippe Nitsche, and Norbert Brändie. 2012. Transport mode detection with realistic Smartphone sensor data. In *Proc. of the 21st International Conference on Pattern Recognition (ICPR).* 573–576.

[46] Xiaoyu Xu. 2024. Automatic Classification of Transportation Modes Using Smartphone Sensors: Addressing Imbalanced Data and Enhancing Training with Focal Loss and Artificial Bee Colony Algorithm. *Journal of Optics* 53, 5 (Nov. 2024), 4656–4670. doi:*10.1007/s12596-024-01703-6*

[47] Kun Yi, Qi Zhang, Wei Fan, Longbing Cao, Shoujin Wang, Hui He, Guodong Long, Liang Hu, Qingsong Wen, and Hui Xiong. 2025. A Survey on Deep Learning based Time Series Analysis with Frequency Transformation. In *Proc. of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD) (KDD '25).* ACM, New York, NY, USA, 6206–6215. doi:*10.1145/3711896.3736571*

[48] Meng-Chieh Yu, Tong Yu, Shao-Chen Wang, Chih-Jen Lin, and Edward Y. Chang. 2014. Big Data Small Footprint: The Design of a Low-Power Classifier for Detecting Transportation Modes. *Proceedings of the VLDB Endowment* 7, 13 (Aug. 2014), 1429–1440. doi:*10.14778/2733004.2733015*

11 Ethical Considerations

Research into attacks carries the potential for misuse, which we weigh against the need to understand emerging threats and to improve the security of widely deployed systems. We follow the principles of the Menlo Report, in particular beneficence, respect for persons, and justice, and design our study to minimize harm while maximizing societal benefit. Our research institute does not operate an Institutional Review Board (IRB), as is common in Europe, so we instead follow established best practices to avoid harm to stakeholders and to protect personal data throughout the study.

Stakeholders. Our work affects several groups with varying degrees of exposure and risk. Participants of our data collection are affected directly, as their recordings reflect travel behavior and movement patterns. Smartphone users at large are affected indirectly, since we show that permissionless magnetometer data reveals transport mode and colocation. The research community and industry stakeholders, such as smartphone manufacturers and operating system developers, benefit from our findings, as these insights help to design improved privacy protections and sensor access controls. Adversaries, in contrast, could misuse the presented techniques to infer user behavior without consent.

Privacy of participants. Participation is voluntary and based on informed consent. All data is processed and analyzed in a controlled environment. For the public release, we pseudonymize stations and segments and convert absolute timestamps into relative ones. These measures reduce, but do not eliminate, the risk of deanonymization, and we explicitly inform participants about this residual risk during the consent process. Together, they mitigate harms such as the privacy loss of participants.

Decision. Our work reveals a security and privacy threat posed by magnetometer data from smartphones, and its publication may allow adversaries to replicate or extend our techniques. At the same time, it enables countermeasures such as permission-based sensor access and reduced sampling rates, protecting a large population of smartphone users. From a respect-for-persons perspective, participant rights are preserved through informed consent and privacy-preserving data handling. While the possibility of misuse cannot be fully eliminated, we believe that disclosure of the threat is necessary to enable effective safeguards. We therefore conclude that the expected benefits outweigh the risks and that both conducting and publishing this research is ethically justified. An extended version of this section is available in our public artifact repository: *https://zenodo.org/records/22662413?preview_file=ETHICS.md*

12 Open Science

We support transparency, reproducibility, and reuse of our work by providing all relevant artifacts whenever possible. Our work is based on three main artifacts required to evaluate our core contributions. These are the dataset, the transport mode detection pipeline, and the colocation approach. For more details refer to the Artifact Appendix.

Dataset. We provide the dataset, which contains raw sensor recordings collected during our study. This includes magnetometer data as well as additional sensors recorded on the smartphones. A detailed description of the dataset is given in the Artifact Appendix.

Transport mode detection. We provide all code required to reproduce our transport mode detection pipeline. This includes scripts for dataset generation, processing and evaluation. We also provide our implementation of the method by Skretting et al. [40].

Colocation. We provide all code required to reproduce our colocation approach. This includes dataset construction, processing of the DTW, model training, and evaluation. We also provide a trained model used in our experiments, the code for training and the optimized hyperparameters of the network.

13 Generative AI Usage

GitHub Copilot and Anthropic Claude were used to assist in transforming research code originally developed in Jupyter notebooks into a structured, release-ready code library. All AI-assisted code was reviewed, tested, and validated by the authors through manual inspection. To improve the understandability of the core library, GitHub Copilot was additionally used for documentation of the published code library. All generated documentation was reviewed and corrected by the authors to ensure accuracy and consistency with the actual implementation. OpenAI ChatGPT and Anthropic Claude were further used to check and improve the grammar, spelling and fluency of the submission. All changes were reviewed, edited, and verified afterward by the authors.

A Transport Mode Detection

A.1 NOR-TMD implementation

To implement the transport mode detection model originally proposed in the reference study, we re-created the preprocessing strategy described in the paper. We applied this strategy to the dataset provided by the authors and chose the XGBoost algorithm for classification as the paper listed it to be the best-performing algorithm on the data by means of classification accuracy.

In our implementation, sensor recordings from mobile devices were first segmented into 10s time windows with an overlap of 5s. Within each time window, we computed the described set of statistical aggregation features that summarize the behavior of each available sensor. These features served as the input variables for the machine learning model. For the model itself, we used the extreme gradient boosting (XGBoost) classifier from the XGBoost library. The gradient boosting process sequentially constructs an ensemble of decision trees, where each new tree corrects the residual errors of the previous ones, leading to a robust classifier capable of capturing complex non-linear relationships between features and transport modes. Our reimplementation closely mirrors the methodology described in the original work.

A.2 Detailed Results

In addition to the highlighted results in Section 6, the released artifacts provide the full list of evaluated parameter combinations with the results as CSV file.

B Colocation inference

B.1 Baseline Implementation

The implementation of the approach by Nguyen et al. [32] is straightforward, as described in Section 7.2. To establish this approach as a strong baseline, we evaluate multiple configurations of sliding window sizes and data point densities. As with our approach, higher data point density leads to a slightly better performance, but in contrast, longer recordings degrade performance. Furthermore, the consistently high standard deviation indicates that the DDTW-based approach lacks robustness across configurations. In our released artifacts we provide the full list of evaluated parameter combinations with the results as CSV file.

B.2 Hyperparameter Search and Network Architecture

Table 9: Best hyperparameters from the Optuna search after 200 epochs. A dash marks layers that are not present.

Trial	940	991	792	952	693
MCC	0.728	0.728	0.716	0.715	0.714
Conv. layers	5	5	5	6	5
Base channels	8	16	16	8	32
Kernel size	7	7	7	7	5
Stride	1	1	1	1	1
Pool kernel	3	3	3	2	3
Embedding dim	32	64	64	64	32
FC layers	3	3	3	1	3
FC width 1	512	128	512	128	256
FC width 2	64	128	128	–	256
FC width 3	64	256	128	–	64
Dropout	0.269	0.269	0.315	0.374	0.269
Learning rate	0.001	0.003	0.001	0.003	0.001
Batch size	128	128	128	128	128

We tune both the network architecture and the training-related hyperparameters jointly using Optuna with the NSGA-II sampler over 1 000 trials, as described in Section 7.4. The full search space and the corresponding objective values for all trials are available in our published artifacts (see Section 12). The five best trials and their results are reported in Table 9.

The architecture of the 1D CNN used for colocation inference is the result of the joint hyperparameter search, balancing detection performance against model complexity to keep inference cost low. The complete layer-by-layer specification, including kernel sizes, number of filters, activation functions, and output dimensions, is provided as part of our published artifacts (see Section 12).