Tiny Sensors, Big Threats: Assessing Motion Sensor-based Fingerprinting in Mobile Systems

Carlos Sulbaran Fandino, Anne Josiane Kouam, Konrad Rieck
BIFOLD & TU Berlin, Germany
c.sulbaranfandino@campus.tu-berlin.de, kouam.djuigne@tu-berlin.de, rieck@tu-berlin.de

***Abstract*—Motion sensors in mobile devices enable device fingerprinting through hardware-induced variations in sensed data. Although the feasibility of this identification technique has been demonstrated across numerous studies, the literature remains fragmented in terms of experimental setups and evaluation metrics—hindering a comprehensive understanding of its effectiveness and limitations. In this work, we provide the first systematization of the motion sensor fingerprinting landscape, structuring the pipeline into distinct stages and identifying key design parameters and countermeasures. Building on this, we develop a unified evaluation framework to assess each parameter in isolation under realistic conditions. Our results show that motion-based fingerprinting remains effective across diverse settings and classifier architectures, yet current countermeasures fail to provide reliable protection and often degrade data utility. We release our dataset to foster reproducibility and future work in this underexplored yet persistent privacy threat.**

***Index Terms*—Mobile Sensors, User fingerprinting, Privacy Countermeasures**

I. Introduction

Mobile devices have become deeply embedded in our daily routines—carried constantly, they serve as extensions of ourselves and continuously reflect our movements, interactions, and surrounding environments. The sensors embedded in these devices not only power user-facing features like step counters and orientation detection, but also produce rich, continuous data streams that intimately capture our physical behaviors. This pervasive presence makes mobile devices a powerful lens into users' real-world activities—and an increasingly attractive target for profiling and surveillance.

Among the growing class of privacy threats, *device fingerprinting* has emerged as a potent method for identifying and tracking individual devices without relying on traditional identifiers such as IP addresses or user accounts. Fingerprinting works by exploiting the subtle, often device-specific differences in how hardware components behave or produce data. These differences, though unintended, are persistent enough to create a unique "fingerprint" that can be recognized across time and applications. While device fingerprinting techniques have primarily focused on browser-level features (e.g., canvas rendering, user-agent strings [1]) or device configurations [2], *a stealthy and still poorly mitigated threat remains underexplored: motion sensor-based fingerprinting [3]–[5].*

Fingerprinting based on sensor data operates at a *low level, beyond the reach of conventional browser protections and user controls*. Motion sensors—such as accelerometers and gyroscopes—are embedded in virtually all smartphones and are routinely accessed by applications for tasks like screen orientation or gesture recognition. Crucially, many platforms, including modern web browsers, allow access to these sensors *without requiring any explicit user permission, making them a silent and convenient target for adversaries.*

These sensors are inherently sensitive to micro-level manufacturing tolerances and calibration discrepancies—a property readily observable in practice. Illustrating the potential of this threat, Fig. 1 shows that *two smartphones of the same make and model* (Iphone 13 Mini), placed side by side on a flat surface *produce distinguishable accelerometer readings*. Such variations form the foundation for extracting persistent, device-specific fingerprints.

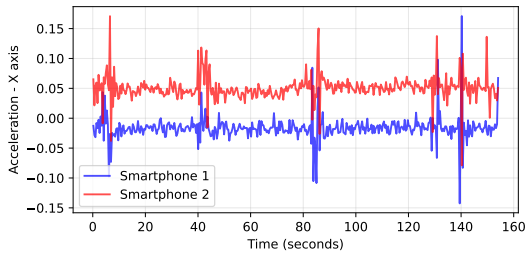
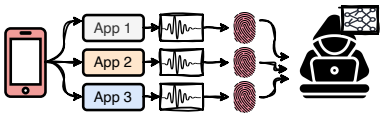


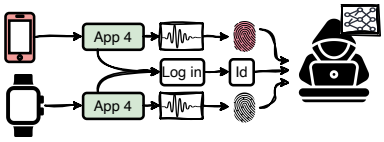
Fig. 1: Accelerometer signals from two identical devices

A typical Motion Sensor-based Mobile Device Fingerprinting (MSMDF) attack involves an adversary embedding malicious code into web platforms or mobile applications, each of which silently collects motion sensor data whenever accessed by a device. From this data, a unique fingerprint is extracted and used to train a classifier capable of recognizing the same device elsewhere. The consequences are profound: attackers can re-identify users across websites and apps, even after they clear cookies or use private browsing modes. Over time, this enables the reconstruction of timelines of app visits and behavioral patterns across services and locations, paving the way for long-term surveillance (cf. Fig. 2a). Additionally, by correlating motion fingerprints across platforms—such as during login events or through behavioral cues—an attacker can link multiple devices that belong to the same user (cf. Fig. 2b). This enables cross-device tracking, deep behavioral profiling and circumvention of anonymization techniques.

Research on MSMDF has already demonstrated its technical feasibility and potential for large-scale application. Early works explored accelerometer and gyroscope fingerprints under controlled settings [6], [7], while later studies evaluated



(a) Cross-platform tracking without identifiers.



(b) Cross-device tracking via correlated fingerprints.

Fig. 2: Illustrative threat models for MSMDF.

their robustness in real-world usage scenarios across a broader set of devices [8], [9]. Recent contributions have employed advanced deep learning techniques to optimize both attacks and defenses [10]. However, existing studies tend to focus on specific threat models, narrow experimental conditions, or isolated defense strategies. This fragmented landscape makes it difficult to compare results, understand generalizability, or assess MSMDF real-world impact.

In this paper, we aim to *systematize and broaden the study of motion sensor-based fingerprinting* by consolidating existing approaches and evaluating them under realistic, diverse, and unified conditions. Our contributions are the following:

- First in **§II**, we formalize the fingerprinting pipeline into distinct stages and provide a structured taxonomy of design parameters and countermeasures. We systematically map existing literature to this framework, identifying coverage gaps and overlooked aspects of MSMDF design and evaluation.
- Then, we design and implement a comprehensive, reproducible evaluation framework to isolate the impact of each design parameter under realistic conditions (**§III**). Beyond classifier performance, we introduce structural metrics that quantify the geometric properties of fingerprint distributions—capturing compactness and separation—to offer deeper insight into fingerprint uniqueness and resilience.
- Our large-scale experiments in **§IV-§VI** reveal key findings across the MSMDF pipeline. We show that motion sensors consistently yield robust fingerprints, even under diverse conditions, and that some fingerprinting enhancements (e.g., audio stimulation) offer only marginal gains. While existing countermeasures reduce fingerprintability slightly, they introduce significant signal distortion—underscoring the need for more effective and utility-preserving defenses.
- Finally, in **§VII**, we distill actionable insights from our findings to guide future research and defense strategies. We highlight the scalability strengths of tree-based attacks, the limits of current countermeasures, and advocate for context-aware protections tailored to realistic threat models.
- The dataset collected in this study—comprising 1,200 annotated motion recordings from 42 smartphones across five real-world collection conditions—is publicly released to support reproducibility and further research at [11].

We conclude in **§VIII** and discuss potential limitations.

II. MSMDF LANDSCAPE SYSTEMATIZATION

Motion sensors play a central role in enabling rich user-device interaction, but they also open up subtle and often overlooked privacy risks. This section provides a structured foundation for understanding Motion Sensor-based Mobile Device Fingerprinting (MSMDF), paving the way for a unified evaluation framework. We begin with a brief technical overview of motion sensors in §II-A, then examine how MSMDF attacks are conducted and which experimental parameters shape their effectiveness in §II-B. A similar analysis is conducted for countermeasures in §II-C. Finally, in §II-D, we highlight the fragmented nature of existing work and position our study as a response to this gap.

A. Motion sensors in mobile devices

Modern smartphones are equipped with motion sensors that allow them to detect orientation, movement, and rotation. In this study, we focus on three of such sensors: the accelerometer, a software-derived gravity sensor, and the gyroscope. Below, we summarize their roles and operational principles.

Accelerometer measures linear acceleration along the x, y, and z axes using a microelectromechanical system. It captures both dynamic motion (e.g., shaking) and static forces like gravity. Values are reported in meters per second squared (m/s^2).

Gravity is derived from the accelerometer and isolates the gravitational component using low-pass filtering. It reflects the steady pull of gravity and helps estimate device orientation in static conditions.

Gyroscope measures angular velocity across three axes using the Coriolis effect. When the device rotates, a vibrating internal mass deflects proportionally to the angular speed, producing a measurable signal. The signal is expressed in radians per second (rad/s).

B. Systematizing MSMDF attacks

MSMDF exploits the inherent characteristics of a device's motion sensor data to create unique fingerprints. The fingerprinting process generally follows a three-stage pipeline:

1) **Data collection:** The ***selected sensors***' data is collected under varying data collection conditions (shortened as ***collection conditions***), such as lab setups, public environments, or with the device placed on a desk, held in hand, or exposed to external stimuli like audio signals or vibrations.
2) **Fingerprint extraction:** Devices typically output three sensor ***data streams*** corresponding to the x, y, and z axes. These streams can be transformed into derived streams such as the *magnitude*, which provides an orientation-invariant measure of motion intensity. Another example is the *inter-sample interval*, the time difference between consecutive readings, reflecting the signal's temporal regularity. From these streams, a ***set of features*** is computed and combined into a *fingerprint vector*.
3) **Fingerprint exploitation:** ***Classifier model(s)*** are trained using these fingerprint vectors to distinguish between devices. Each device acts as a class label, and the classifier learns to associate specific feature patterns with individual

devices. Classification performance is typically evaluated using metrics such as accuracy, precision, recall, or F1-score, which reflect how well the model can identify or re-identify a device from its fingerprint.

While the fingerprinting pipeline may appear straightforward, its actual performance is highly sensitive to a range of design and evaluation choices. These choices affect not only the discriminative power of the fingerprints, but also the scalability and real-world feasibility of MSMDF attacks. To better understand and compare prior work, we systematize the parameters that influence both fingerprint quality and evaluation outcomes.

We group these parameters into two categories. The first concerns the *design of the fingerprint itself*, which we refer to as *design-related parameters*. The second category, which we call *scalability-related parameters*, pertains to how the fingerprinting model is trained and evaluated during the exploitation phase, such as the number of devices involved or the data volume per device. These scalability dimensions are critical for understanding how fingerprinting systems behave beyond controlled scenarios. Table I overviews all existing studies, to the best of our knowledge, structured by these parameters. Columns 2–7 highlight design-related factors, while the last two columns, shaded in grey, reflect scalability-related aspects.

a) Design-related parameters: We identify seven core design decisions that may affect fingerprinting performance:

- ***Sensor selection*** refers to which motion sensors are used—most commonly the accelerometer, gyroscope, and gravity sensor.
- ***Collection conditions*** describe the physical environment during sensing, e.g., phone on a desk or under audio stimuli.
- ***Sampling rate*** is the frequency at which sensor values are recorded. It affects the resolution of the signal and the number of features derivable from it.
- ***Data streams*** captures how raw sensor axes (x, y, z) are transformed into derived representations such as magnitude, azimuth, inclination, or inter-sample intervals. These affect how orientation and temporal properties are encoded.
- ***Feature set*** defines which statistical features are extracted from the streams. *Time*-domain and *frequency*-domain *features* vary in their ability to capture signal characteristics.
- ***Window length*** is the time span over which features are extracted. Longer windows may provide more signal structure but reduce responsiveness.
- ***Classifier*** defines the machine learning model used to distinguish between devices based on fingerprint vectors.

b) Scalability-related parameters: Beyond design performance, the scalability of an MSMDF attack depends on:

- ***#Devices***: The size of the targeted device population.
- ***#Fingerprint vectors per device***: The volume of training data available per device.
- ***Train:test ratio***: The proportion of labeled data used to train the classifier.
- ***Known:unknown device ratio***: Whether the evaluation assumes a closed-world (all devices seen during training) or open-world (unknown devices present at test time) setting.

Despite shared objectives, prior work differs markedly in the parameters it explores. For instance, Dey et al. [6] focus on a single sensor in a controlled lab environment, using centrally managed devices and a minimal feature set. In contrast, studies like [7], [8] incorporate multiple sensors and richer data streams, but often limit classifier diversity or rely on entirely distinct feature sets. Some parameters such as *sampling rate* and *window length* are frequently omitted or only implicitly addressed. This uneven coverage hampers comparability across studies and obscures the specific impact of each design choice on fingerprinting performance.

The fragmentation extends to scalability considerations. For example, Das et al. [8] evaluates over 200 devices in open-world public settings, collecting data from users in real-life usage scenarios. On the other hand, [5] remains confined to a controlled lab setting and tests only 10 devices. Additionally, the number of fingerprints per device and the sampling density vary considerably across studies. In the absence of standardized benchmarks or consistent reporting practices, it becomes challenging to assess how well MSMDF techniques generalize across populations, devices, or conditions.

C. Systematizing countermeasures

To mitigate the privacy risks posed by MSMDF attacks, researchers have proposed countermeasures that perturb motion sensor data before it reaches applications or web scripts. We identify four of such methods in the literature, all injecting controlled distortions into raw data streams to degrade fingerprint stability in the feature space.

a) Uniform Noise Addition (UNA) [7]: This technique perturbs each sensor axis using random *scaling* and *offset* values drawn from uniform distributions centered at 1 and 0, respectively. The transformation follows: $\tilde{\text{data}} = \frac{\text{data}_o - \text{offset}}{\text{scaling}}$.

b) Laplace Noise Addition (LNA) [7]: This variant draws the *offset* and *scaling* values from Laplace distributions rather than uniform ones. It is inspired by the differential privacy framework [12], with noise calibrated based on a predefined privacy budget ϵ.

c) Quantization (Q) [13]: This method discretizes sensor readings into fixed-size bins, treating all values within a bin as identical. This effectively reduces precision and blurs the subtle variations exploited by fingerprinting algorithms: $\tilde{\text{data}} = \left\lfloor \frac{\text{data}_o}{\text{bin\_size}} \right\rfloor \cdot \text{bin\_size}$. For accelerometer data, quantization is typically applied after transforming the raw cartesian signals into polar coordinates. Separate bin sizes are then used for the derived *magnitude* and *angle* components to control the granularity of anonymization. In contrast, gyroscope data is left in its original form with only angle quantization applied.

d) Generative Models [10]: Li et al. explore a DL-based approach using generative models to rewrite sensor data in a way that obfuscates the original device identity. Though promising in performance, these methods demand computational resources that exceed current mobile hardware capabilities, limiting their practical deployment and justifying their exclusion from the scope of this study.

TABLE I: Overview of MSMDF attacks in the literature. $FingerprintSize = \#Streams \times \#features$

Ref.	Sensors	Data Streams	Time Features	Frequency Features	Collection conditions	Classifiers	#Devices	#Fingerprints
[6]	Accelerometer	Inter sample time, Magnitude	Avg. Dev. [6]–[8] Kurtosis [5]–[9] Max [5]–[9] Mean [6]–[9]	DC Offset [9] Irregularity J and K [6] Low-Energy-Rate [7], [8] Smoothness [6]	Lab: on desk with vibrations	Bagged Decision Trees [Bagged DT] [6], [7] Bagged KNN [Bagged k-NN] [8] Decision Tree [DT] [5], [7] Ensemble Subspace Discriminant [5] eXtreme Gradient Boosting [9] Extra Trees [ET] [8], [9] Gaussian Naive Bayes [GNB] [7], [8] K-Nearest Neighbors [k-NN] [5], [7]–[9] Linear Discriminant Analysis [LDA] [8] Logistic Regression [8], [9] Multilayer Perceptron [MLP] [8] Random Forest [RF] [8], [9] Stochastic Gradient Descent [SGD] [8] Support Vector Machine [SVM] [5], [7]–[9] Quadratic Discriminant Analysis [7] Wide Neural Network [wide_NN] [5]	80 accelerometers, 25 smartphones, 2 tablets (9 models)	50 per device (size 36)
[7]	Accelerometer, Gyroscope	Accelerometer: Magnitude Gyroscope: X, Y, Z	Min [6]–[9] Mode [9] Non-Negative Count [7], [8] Range [9] RMS [5]–[9] Signal to Noise Ratio [5] Skewness [5]–[9] Std. Dev. [5]–[9] Var [9] Zero Crossing Rate [7], [8]	Sepc. Attack Time [7], [8] Sepc. Attack Slope [7], [8] Spec. Brightness [7], [8] Spec. Centroid [6]–[9] Spec. Crest [6] Spec. Entropy [7]–[9] Spec. Flatness [6]–[9] Spec. Flux [7], [8] Spec. Irregularity [7], [8] Spec. Kurtosis [6]–[9] Spec. Mean [9] Spec. RMS [7], [8] Spec. Roll-off [6]–[8] Spec. Roughness [7], [8] Spec. Skewness [6]–[9] Spec. Spread [7]–[9] Spec. Std. Dev. [6], [9] Spec. Var [9]	Lab: (i) on desk, (ii) on hand (a) no audio, (b) inaudible, (c) popular song Public: (i) on desk (a) no audio, (b) inaudible		Lab: 30 smartphones Public: 63 smartphones (24 models)	Lab: 60 per device Public: 20 per device (size 70)
[8]	Accelerometer (with & without gravity), Gyroscope	Accelerometer: X, Y, Z, Magnitude, Azimuth, Inclination Gyroscope: X, Y, Z, Magnitude			Public: (i) on desk, (ii) on hand		On desk: 294 smartphones On hand: 256 smartphones (45 models)	20 per device (size 400)
[9]	Accelerometer	Magnitude			Public: (i) on desk, (ii) on hand, (iii) moving		7 smartphones (4 models)	180 per device (size 20)
[5]	Accelerometer	Magnitude			Lab: on desk with vibrations		10 smartphones (5 of same model)	200 per device (size 7)

TABLE II: Overview of countermeasures in the literature.

Countermeasure	Strength (param)	Strength (value)	Resampling frequency
Uniform NA [7]	offset $\sim$ scaling $\sim$	$\mathcal{U}(-0.5, 0.5)$ for Acc. $\mathcal{U}(-0.1, 0.1)$ for Gyr. $\mathcal{U}(0.95, 1.05)$	Once per fingerprint
Laplace NA [7]	offset $\sim$ scaling $\sim$	$Lap(0, 0.5)$ $Lap(1, 0.5)$	Once per fingerprint
Quantization [13]	bin size =	1 for *Magnitude* 6 for *Angle*	No resampling

We systematize these defenses based on two implementation parameters that can significantly affect their effectiveness and utility. Table II summarizes existing countermeasures:

- ***Countermeasure strength***: the magnitude of perturbation applied to the signal. For noise-based methods, this corresponds to the range or scale of the sampling distribution; for quantization, to the granularity of bins. Stronger perturbations offer greater privacy but may compromise utility.
- ***Resampling frequency***: the rate at which new perturbation parameters (e.g., noise offset and gain) are drawn and applied to sensor data. In prior works, this frequency is typically defined relative to the fingerprinting duration as parameters are refreshed once per fingerprint extraction window. Lower resampling frequencies lead to more consistent transformations, which may preserve identifiable patterns. In contrast, higher frequencies inject more variability, enhancing obfuscation at the risk of distorting the signal.

D. Positioning and research questions

As shown in our systematization, existing work on MSMDF—both attacks and defenses—remains fragmented. Fingerprinting studies vary widely in design parameters such as sensor choice, data stream structure, and feature sets, while often omitting key details like window length or sampling rate. Countermeasures are similarly underexplored with limited understanding of how parameter tuning affects performance.

This lack of comparability creates uncertainty around the real-world feasibility and robustness of MSMDF techniques. It remains unclear which fingerprinting configurations are most effective across conditions, how well the attacks scale, and which defense strategies offer meaningful protection without degrading utility.

This leads us to three core research questions:

- **RQ1:** *How do different fingerprint design parameters impact MSMDF performance?*
- **RQ2:** *How does the attack scale with the number of devices, the amount of training data, and the evaluation setting?*
- **RQ3:** *How effective and utility-preserving are existing countermeasures when varied in strength and frequency?*

To answer these questions, we conduct a broad evaluation of MSMDF using a unified experimental framework. Our goal is to provide a more complete and reproducible understanding of MSMDF capabilities and limitations.

III. Empirical evaluation framework

To evaluate MSMDF feasibility, scalability, and mitigation, we design a modular empirical evaluation framework that balances realistic attacker goals with controlled, parameter-wise analysis across the fingerprinting pipeline. This section outlines our methodology, metrics, and experimental setup.

A. Evaluation metrics

To evaluate both the effectiveness of motion sensor-based fingerprinting and the impact of proposed countermeasures, we adopt a dual-perspective evaluation framework. This combines structural analysis of the fingerprint space with classifier-based performance metrics, offering insights into how well devices can be distinguished, but also what makes a fingerprint inherently robust or fragile. In parallel, we quantify how countermeasures degrade fingerprinting performance and affect the utility of the underlying sensor data.

To this end, we rely on three complementary families of metrics: *structural metrics*, which characterize fingerprint distributions using unsupervised clustering scores; *attack effectiveness metrics*, which quantify device classification performance in realistic attack scenarios; and *countermeasure impact metrics*, which assess both fingerprint obfuscation and preservation of sensor utility.

1) Structural metrics: We compute three metrics that describe fingerprint structure in the high-dimensional feature space: *compactness*, *separation*, and *entanglement*. The first two are computed per device and reflect intra- and inter-class spread; the last one is computed per fingerprint and give a global sense of quality. For each metric, we report its distribution across the dataset.

a) Compactness: Measures how tightly grouped the fingerprints of a given device are. A lower value indicates that fingerprints are consistent and easier to learn: Compactness $= \frac{1}{|D|}\sum_{i,j\in D} d(x_i, x_j)$ where D is the set of fingerprints from the same device, and d is the Euclidean distance.

b) Separation: Measures how well fingerprints of different devices are separated. High values indicate inter-device distinguishability: Separation $= \frac{1}{|G|}\sum_{i,j\in G, i\neq j} d(\mu_i, \mu_j)$ where G is the set of devices and μ_i the centroid of device i.

c) Entanglement: Quantifies the degree of overlap between neighboring clusters. Lower values suggest cleaner cluster boundaries and stronger fingerprint uniqueness: Entanglement$(x_i) = \frac{\text{\#same-device neighbors}}{\text{k}}$, where k is the number of nearest neighbors, heuristically set to $\sqrt{N}$ with N denoting the total number of fingerprints in the dataset.

2) Attack effectiveness metrics: To assess fingerprinting from an attacker's perspective, we use classical classification metrics—namely, *accuracy* and *F1-score*—to evaluate how well fingerprints enable device identification. Accuracy reflects overall prediction correctness, while F1-score balances precision and recall, making it more informative in the presence of class imbalance. As our task involves multi-class classification across numerous devices (i.e., users), we report both metrics using the *macro-average* approach. This means each class (device) contributes equally to the final score, regardless of its number of samples—providing a fair view of per-device fingerprinting effectiveness.

3) Countermeasure impact metrics: To assess the effectiveness of known MSMDF defenses, we apply each countermeasure directly to the raw motion data before fingerprint construction. Their impact is evaluated along two key dimensions:

- Fingerprinting performance degradation: We quantify the reduction in MSMDF attack performance by measuring the change in accuracy (Δ Accuracy) of classifiers before and after applying the countermeasure.
- Sensor data utility: We assess the impact of the transformation on sensor data by comparing the distribution of anonymized vs. raw readings. For this, we compute the Hellinger distance, providing a score within [0-1] range, where lower values indicate minimal distortion and higher values signal reduced utility.

B. Study scope and parameter design

Our evaluation aims to quantify the individual impact of each parameter across fingerprint design, scalability, and mitigation within the MSMDF pipeline. To this end, we adopt a controlled experimental methodology: for each parameter, we define a representative range of values and assess its impact in isolation, while keeping other parameters fixed to a common default. These default values strike a practical balance between fingerprint distinctiveness and realistic attacker capabilities, and are reused consistently in the remaining experiments.

TABLE III: Evaluation settings for each attack and countermeasure parameter.

Evaluation setting	Possible Values	Used Values	Default Value
Sensor selection (Figs. 3a, 3b)	Accelerometer, Gravity, Gyroscope	Each individually, All	Accelerometer, Gyroscope
Collection conditions (Fig. 3c)	On hand, On desk, On hand audio, On desk audio, Walking	Each individually, All	On hand, On desk, On hand audio, On desk audio
Data streams (Figs. 3e, 3d)	X, Y, Z, Magnitude, Inter-Sample time (IST), Azimuth, Inclination	{Magnitude}, {IST}, {Magnitude, IST}, {X, Y, Z, Magnitude}, {X, Y, Z, Magnitude, IST}	All
Feature set (Fig. 3f)	34 different (cf. Table I)	Time Domain, Frequency Domain	All
Window length (Fig. 4a)	1 to 10 seconds	2, 4, 6, 8, 10 seconds	6 seconds
Sampling rate (Fig. 4b)	1 to 200 Hz	20, 40, 60, 80, 100 Hz	100 Hz
Classifier (Fig. 3b)	12 different (cf. Figure 3b)	Each individually	The best
#Devices (Fig. 5a)	$[1, 42]$	5 to 42 $(step = 10)$	42
#Fingerprints per device (Fig. 5b)	$[6, 60]$	6 to 60 (step = 6)	$2 \times 6 = 12$
Train:Test ratio (Fig. 5c)	$(0.0, 1.0)$	0.1 to 0.9 $(step = 0.1)$	0.5
Known:Unknown device ratio (Fig. 5d)	$(0.0, 1.0)$	0.1 to 1.0 $(step = 0.1)$	1.0
Strength (Fig. 6a)	Scaling factor	$\times\frac{1}{3}$, $\times\frac{1}{2}$, $\times1$, $\times2$, $\times3$	$\times1$
Resampling freq. (Fig. 6b)	Number of resamplings per fingerprint	$\times5$, $\times4$, $\times3$, $\times2$, $\times1$	$\times1$

Table III summarizes the parameter space considered in our study. Rows 1–7 cover fingerprint design choices, Rows 8–11 address scalability, and Rows 12–13 describe countermeasure configurations. This systematic and labeled exploration reflects the broad and nuanced scope of our analysis.

To ensure clarity and ease of navigation, we organize our findings into three overarching categories: fingerprint design (**FD**, §IV), scalability (**FS**, §V), and countermeasures (**FC**, §VI). Each finding is sequentially labeled (e.g., FD1, FD2, FS1…), allowing readers to follow the empirical narrative and directly link each result to its corresponding evaluation setup.

C. Implementation details

We implement the full MSMDF pipeline, encompassing data collection, feature extraction, classification, and countermeasure evaluation. We adopt a real-world public setting, where each participant completed multiple recording sessions on different days, using our web-guided interface[1] and the Sensor Logger app [14]. In each session, around 2 min of sensor data were recorded while the participant sequentially followed five predefined collection conditions. The raw recordings were then segmented into 10-second motion intervals and labeled according to the corresponding collection condition. Labeling was performed using heuristic-based detection methods, followed by manual refinement to ensure accuracy. This resulted

[1]https://carlossulba.github.io/MSMDF-Study-website/

in a dataset of 1,200 labeled 10-second recordings from 42 smartphones, serving as the foundation for our evaluation.

From the labeled raw data, we generated up to 19 derived streams per sensor and computed 34 statistical features per stream. This modular setup enables to generate dedicated datasets for each fingerprint design configuration. For performance evaluation, we reproduce 12 classifier models from the MSMDF literature, optimizing each using grid search and 2-fold cross-validation. To evaluate countermeasures, we apply the same fingerprint extraction pipeline to sensor traces transformed using the three anonymization techniqueseach tested under varying parameter strengths and resampling frequencies.

IV. Fingerprint design analysis

We first evaluate how different fingerprint design parameters impact MSMDF performance. We analyze the fingerprints' structural properties and their classification effectiveness.

A. Sensor selection and Classifier

a) Sensor selection: We evaluate the fingerprinting contribution of three motion sensors: the accelerometer, gyroscope, and gravity. The accelerometer and gyroscope are standard in smartphones, while the gravity sensor applies internal filtering to emphasize low-frequency signals. We test each sensor independently and in combination to assess complementary value. Our default configuration combines accelerometer and gyroscope inputs reflecting common app-level access.

b) Classifier choice: We reproduce prior work classifiers, spanning tree-based, distance-based, statistical, and neural models. This diversity ensures broad comparability and captures distinct generalization behaviors. The best-performing model is used as default value, representing a strong attacker.

c) Results: Structurally, as shown in Fig. 3a, individual sensor fingerprints are compact but not well-separated—especially for the gyroscope, which shows high entanglement. Combining sensors enhances inter-device separation with minimal compactness loss. Classifier-wise, Fig. 3b reveals two performance groups. Tree-based models (e.g., Random Forest, Extra Trees) outperform others and benefit most from sensor fusion. Distance-based models (KNN, GNB) and neural networks underperform, likely due to their limited ability to model complex, axis-level feature interactions.

FD1: *Combining multiple motion sensors significantly improves fingerprint separability. The accelerometer alone provides the best single-sensor performance.*
FD2: *Tree-based classifiers are the most effective for MSMDF, consistently outperforming other models and scaling well with increased sensor input. Distance-based models are more sensitive to signal complexity.*

B. Collection conditions

We define five environmental conditions to simulate realistic smartphone usage: (i) held in hand, (ii) placed on desk, (iii) held in hand with audio stimulation, (iv) on desk with audio, and (v) while walking. These scenarios represent different levels of motion noise and device stability, and are recorded in *public settings*—where participants collect data in everyday environments following predefined instructions. This choice reflects realistic adversarial contexts and ensures our findings generalize beyond controlled laboratory setups. To isolate each condition's impact, we train and test classifiers on each scenario independently. Our default is the *on desk with audio* condition, selected for its reproducibility and its ability to capture hardware-level motion patterns.

a) Results: As shown in Fig. 3c, classifiers perform best when both training and test data originate from the same condition. Audio stimulation, however, has minimal effect—models trained with and without it yield similar results. In contrast, walking introduces notable signal variability: fingerprints remain effective when training and testing both use walking data, but generalize poorly in cross-condition scenarios. Finally, combining data from all five conditions yields the highest overall accuracy, even outperforming static-only settings. This suggests that environmental diversity enhances fingerprint generalizability by enriching signal variability.

FD3: *Fingerprints collected under stable conditions (with or without audio) are highly consistent. Including diverse conditions, especially walking, improves overall MSMDF performance by capturing richer signal variability.*

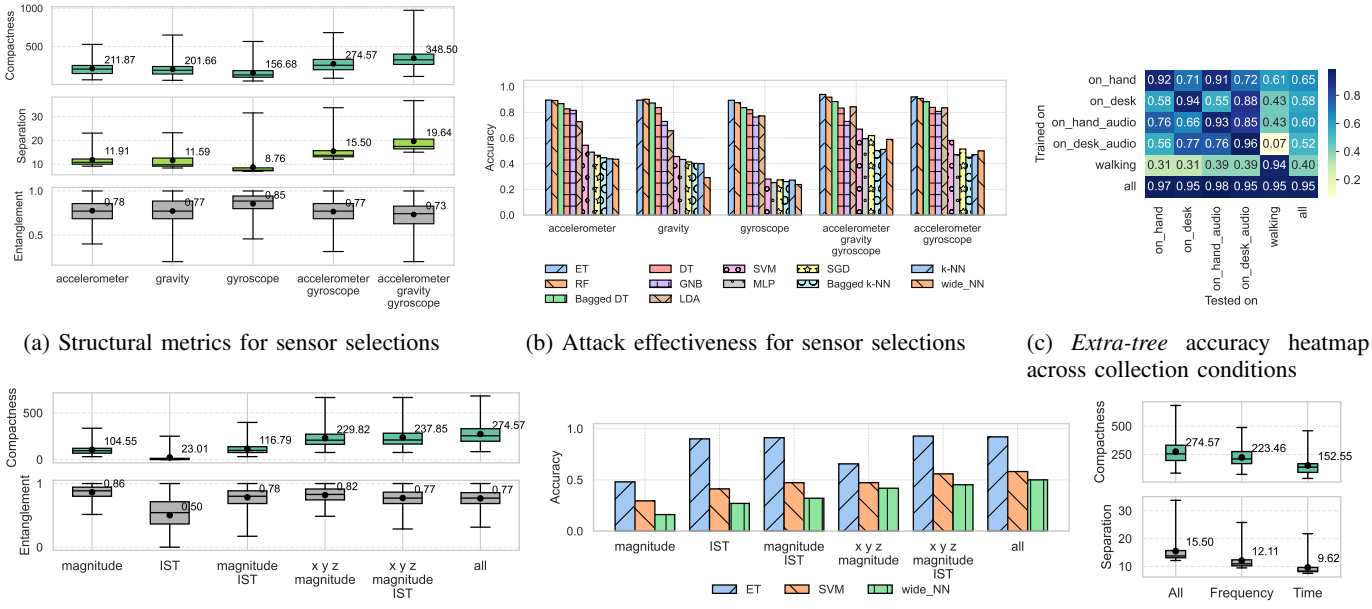
C. Data streams

Each sensor outputs three raw streams (x, y, z). From these, we derive additional representations to enrich the fingerprint: (i) *Magnitude*, which provides an orientation-invariant norm; (ii) *Inter-sample time* (IST), capturing temporal irregularities; and (iii) *Azimuth* and *Inclination*, reflecting device orientation.

We incrementally evaluate the contribution of each stream. Our default setup includes all data streams to maximize representational richness without requiring hardware modifications.

a) Results: Fig. 3d shows the structural fingerprint distribution across streams. Using only magnitude leads to low inter-device separation and high entanglement—indicating overlapping fingerprints. In contrast, IST produces compact and well-separated clusters, revealing its potential to exploit subtle, device-specific timing artifacts. Combining multiple streams increases separation with minimal change in entanglement, benefiting from a higher-dimensional feature space. Attack effectiveness results in Fig. 3e confirm these trends for three classifiers: Extra Trees (best), SVM (moderate), and Wide NN (worst). All models consistently rank IST above magnitude. Notably, IST alone achieves high accuracy, especially for top-performing models, underscoring its discriminative strength. However, performance improves further when additional streams are combined, highlighting the value of a richer feature set.

FD4: *Inter-sample time is a highly discriminative stream for fingerprinting. Combining it with other data streams enhances both fingerprint separability and attack success.*



(a) Structural metrics for sensor selections

(b) Attack effectiveness for sensor selections

(c) *Extra-tree* accuracy heatmap across collection conditions

(d) Structural metrics for data streams

(e) Attack effectiveness for data streams

(f) Structural metrics for feature sets

Fig. 3: MSMDF performance across fingerprinting design-related parameters

D. Feature set

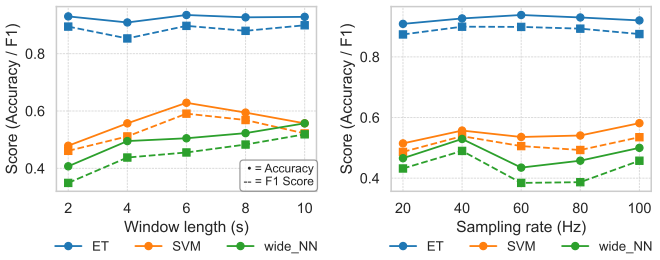
We evaluate statistical features extracted from both the time and frequency domains. From prior work, we select 13 time-domain and 21 frequency-domain features. Our evaluation compares using alone time-domain and frequency-domain features and the full set. The default uses both domains to maximize fingerprint separability, as reflected in past studies.

a) Results: Structural analysis (Fig. 3f) focuses on cluster compactness and separation, as entanglement remains largely unaffected. Time-domain features result in more compact clusters—indicating intra-device consistency—while frequency-domain features yield higher inter-device separation, likely due to their greater number and spectral sensitivity. Classifier performance (not shown) is nearly identical across time- and frequency-only configurations, with the full feature set offering only marginal improvement. This suggests that both domains capture overlapping device-specific traits and points to possible redundancy in the combined set.

FD5: *Time- and frequency-domain features yield comparable fingerprinting performance. Their combination offers limited gain, indicating redundancy and potential for feature reduction.*

E. Window length

Fingerprint extraction is performed over fixed-length time windows. We evaluate window lengths of 2, 4, 6, 8, and 10 seconds. Shorter windows enable faster and more frequent fingerprinting but may produce less stable features. In contrast, longer windows capture more behavioral information but require uninterrupted data, which may be harder to obtain



(a) Window length

(b) Sampling rate

Fig. 4: Attack effectiveness across temporal parameters.

in real-world conditions. We use 6 seconds as the default, balancing stability and data availability.

a) Results: As shown in Fig. 4a, classification accuracy improves slightly with increasing window length, but the gain quickly plateaus around 6 seconds. In some cases, performance even decreases beyond this point, suggesting diminishing returns with longer recordings. These findings indicate that extended recordings do not consistently enhance fingerprint quality. Notably, top-performing classifiers already achieve high accuracy with windows as short as 2–4s.

FD6: *Longer recordings offer limited benefit. High fingerprinting performance can be achieved with windows as short as 2–4s.*

F. Sampling rate

Motion sensor sampling frequency varies across devices. To simulate this, we evaluate fingerprinting performance at 20, 40, 60, 80, and 100 Hz. This allows us to assess robustness

under bandwidth-limited or resource-constrained conditions. We adopt 100 Hz as our default, as it preserves signal fidelity and represents the upper bound typically available on consumer smartphones.

a) Results: As in Fig. 4b, reducing the sampling rate from 100 Hz to 20 Hz has minimal effect on classification performance. MSMDF remains effective even at low sampling frequencies, showing attacks feasibility on low-power devices with limited sensor access.

FD7: *MSMDF remains robust at lower sampling rates, reducing overhead without loss in performance.*

V. Fingerprint scalability analysis

We now analyze how MSMDF attack performance evolves as the scenario scales in complexity. All experiments use a fixed, high-performing fingerprint design reported in Table III. Fig. 5 summarizes results for each scalability parameter.

A. Number of devices

Fingerprinting performance often degrades as the number of target devices grows, due to growing inter-device similarity and classifier confusion. We evaluate this by incrementally expanding the dataset from the top 5 devices (with the most fingerprints) to the full set of 42 devices, in steps of 5. This setup captures how fingerprint discriminability scales with population size. Our default configuration includes all 42 devices, reflecting realistic attacker goals.

a) Results: As in Fig. 5a, the top-performing classifier (Extra Trees) shows only a modest decline in accuracy and F1-score, dropping from 100% to 88% as the number of devices increases. In contrast, second-tier models such as SVM and Wide NN experience sharper performance drops, indicating weaker generalization under scale.

FS1: *Tree-based classifiers maintain strong fingerprinting accuracy at scale, while other models degrade sharply.*

B. Minimum fingerprints per device

An attacker's success depends on how many labeled fingerprints they can collect per target device. Since fingerprint availability varies widely across users, we simulate this by varying the min. number of fingerprints required for a device to be included in the training set. With a 6-second window, this threshold ranges from 6 to 60 fingerprints. This captures both opportunistic attacks based on sparse observations and long-term surveillance with abundant data. By default, we include only devices with at least two independent recording sessions, which yield 12 fingerprints—striking a practical balance between data availability and evaluation robustness.

a) Results: Fig. 5b shows that classifier performance improves as more fingerprints are available per device, particularly for weaker models. Extra Trees reaches near-perfect accuracy beyond 36 samples and plateaus, while other classifiers continue slowly improving.

TABLE IV: Effect of countermeasure strength on Hellinger distance (accelerometer magnitude)

Countermeasure	Collection setting	$\times\frac{1}{3}$	$\times\frac{1}{2}$	$\times1$	$\times2$	$\times3$
Uniform NA	*On desk*	0.57	0.64	0.78	0.84	0.88
	On hand	0.26	0.37	0.56	0.72	0.75
	On desk audio	0.45	0.50	0.58	0.67	0.77
	On hand audio	0.22	0.26	0.47	0.57	0.65
	Walking	0.02	0.02	0.04	0.05	0.09
Laplace NA	*On desk*	0.62	0.69	0.76	0.80	0.81
	On hand	0.35	0.43	0.61	0.64	0.67
	On desk audio	0.45	0.56	0.64	0.69	0.69
	On hand audio	0.32	0.36	0.50	0.58	0.58
	Walking	0.07	0.10	0.17	0.26	0.23
Quantization	*On desk*	0.99	0.99	0.99	0.99	0.99
	On hand	0.97	0.99	0.99	0.99	0.99
	On desk audio	0.98	0.98	0.99	0.99	0.99
	On hand audio	0.96	0.98	0.99	0.99	0.99
	Walking	0.68	0.75	0.83	0.89	0.93

FS2: *Increasing the number of fingerprints per device enhances identification. Strong models achieve stable performance from 36 samples (≈3.5 min at 6s per sample).*

C. Train-Test ratio

The amount of labeled data available during training significantly impacts classifier generalization. We vary the train-test split from 10:90 to 90:10 to simulate attackers with different levels of data access. Our default setting, 50:50, offers a balanced view of training capacity and test robustness.

a) Results: As shown in Fig. 5c, tree-based classifiers reach near-optimal performance with just 40% of the data used for training. In contrast, weaker models such as neural networks or distance-based classifiers benefit more from additional data, yet still plateau below top-performing models.

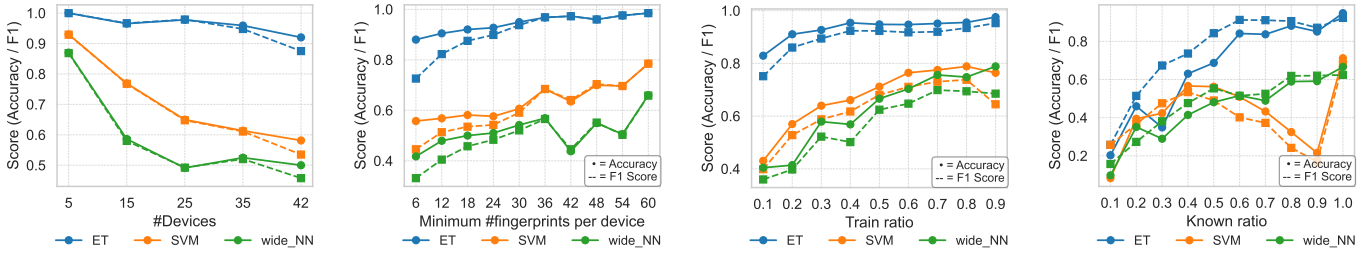
FS3: *Top classifiers reach peak performance with little training data, enabling attacks even in short periods.*

D. Known-Unknown ratio

Real-world attackers often face unknown devices not seen during training. To simulate this, we evaluate an open-world setting where only a subset of devices is known at training time. We vary the proportion of known devices from 10% to 100%, while testing always includes the full device set. A confidence threshold of 0.5 is used to determine whether a fingerprint belongs to a known device.

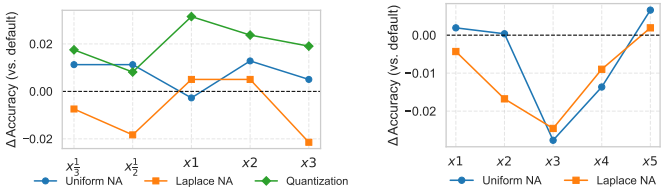
a) Results: As in Fig. 5d, classifier performance drops sharply when fewer than 30% of devices are known. Tree-based models improve quickly as the proportion of known devices increases but remain unreliable below the 40% mark. Notably, the Wide NN maintains steady—but modest—accuracy across all settings, indicating limited adaptability.

FS4: *Open-world settings significantly limit classifier performance. MSMDF attacks are effective only when a sufficient share of devices is known during training.*



(a) Number of devices (b) Min. #fingerprints per device (c) Train:test ratio (d) Known:unknown ratio

Fig. 5: Overview of MSMDF performance across fingerprinting scalability-related parameters.



(a) Δ accuracy w.r.t. strength (b) Δ accuracy w.r.t. freq.

Fig. 6: Change in fingerprinting performance (after – before countermeasure) with the Extra Trees classifier.

TABLE V: Effect of resampling freq. on Hellinger distance (accelerometer magnitude)

Countermeasure	Collection setting	×1	×2	×3	×4	×5
Uniform NA	*On desk*	0.78	0.77	0.77	0.77	0.76
	On hand	0.53	0.56	0.55	0.55	0.54
	On desk audio	0.60	0.59	0.58	0.59	0.60
	On hand audio	0.44	0.45	0.45	0.45	0.46
	Walking	0.03	0.03	0.03	0.03	0.04
Laplace NA	*On desk*	0.77	0.79	0.76	0.77	0.76
	On hand	0.59	0.59	0.56	0.57	0.59
	On desk audio	0.59	0.61	0.60	0.59	0.61
	On hand audio	0.52	0.50	0.49	0.48	0.52
	Walking	0.20	0.20	0.21	0.19	0.18

VI. Evaluating Countermeasures

We finalize our empirical study by examining how variations in countermeasure strength and resampling frequency impact fingerprint robustness and the utility of motion sensor data.

A. Countermeasure strength

This parameter controls how strongly the sensor data is perturbed. For noise-based countermeasures, strength determines the range from which random offsets and scaling factors are drawn. For quantization, it defines the bin size used to discretize sensor readings. We simulate varying strengths by scaling the original parameter settings using multiplicative factors: $\times\frac{1}{3}$, $\times\frac{1}{2}$, $\times 1$ (baseline), $\times 2$, and $\times 3$. This approach enables consistent evaluation of the privacy–utility trade-off across all techniques.

a) Results: Fig. 6a shows that classifier accuracy remains largely unaffected across strength levels. For the Extra Trees classifier, the maximum accuracy drop is under 2%, despite a high baseline of 92%. In some cases, stronger perturbation even improves accuracy (up to +3%), possibly due to the introduction of consistent artifacts that aid classification. This suggests that simply increasing countermeasure strength does not reliably weaken fingerprinting. Table IV reports the Hellinger distance between the original and perturbed accelerometer magnitude distributions. Most values exceed 0.4 and reach up to 0.9 for quantization, indicating substantial signal alteration. As expected, stronger countermeasures lead to greater distribution shifts. The walking condition shows less divergence due to its naturally high signal variability.

FC1: *All countermeasures cause significant signal distortion, yet fail to meaningfully reduce fingerprinting performance. Stronger perturbation does not guarantee better protection and often harms data utility.*

B. Resampling frequency

This parameter controls how often new countermeasure values (e.g., noise offset and gain) are applied within a fingerprint window. While prior work typically uses a single transformation per window, we evaluate higher resampling frequencies—up to 5 times per window—to test whether more dynamic perturbations reduce fingerprint consistency.

a) Results: As shown in Fig. 6b, fingerprinting accuracy remains largely unaffected, with only a modest F1-score and accuracy drop (up to 3%) at intermediate frequencies ($\times 3$–$\times 4$). At higher rates, performance stabilizes or slightly improves, suggesting that frequent perturbation may introduce new learnable patterns rather than eliminating them. Table V shows consistently high Hellinger distances across all resampling levels, confirming significant signal distortion. However, this distortion does not translate into meaningful protection, and the "walking" condition again shows the least divergence.

FC2: *Increasing resampling freq. has limited impact on attack performance but continues to degrade signal utility. More dynamic pertubations alone are insufficient for effective mitigation.*

VII. Insights and Implications for Future Work

Following MSMDF systematic study, this section revisits our research questions to synthesize key insights and outline avenues for future research and practical defense strategies.

A. RQ1: What enables a strong fingerprint?

Fingerprint strength stems from combining sensor modalities, stream inputs, and timing features—especially inter-sample timing (IST), a highly discriminative signal. Tree-based classifiers dominate due to their ability to partition high-dimensional spaces and scale with many classes, unlike SVMs or neural models. These insights call for defenses tailored to attacker model capabilities, especially ensembles.

B. RQ2: How far can the attack scale?

MSMDF attacks scale effectively: Extra Trees achieves >90% accuracy across 42 devices with only 36 samples per device (≈4 min of motion). Performance stays strong with limited training, though generalization in open-world settings drops below 40% known population. This signals the privacy risk of passive data collection [3] and motivates tighter restrictions on motion sensor access at the platform level.

C. RQ3: Can we defend without breaking utility?

Despite the diversity of countermeasure configurations tested, none succeeded in substantially degrading attack performance—even under aggressive parameter tuning and dynamic perturbation schemes. While transformations such as quantization and noise injection visibly altered the sensor distribution (with Hellinger distances up to 0.9), fingerprint accuracy dropped by at most 3–4%.

This ineffectiveness stems from a fundamental limitation: existing countermeasures operate at the signal level without understanding or targeting the actual fingerprint source—the hardware manufacturing imperfections. This is analogous to camera sensor fingerprinting, where robust defenses have only emerged once models of sensor noise and lens artifacts were properly formalized [15]. In the case of MSMDF, we lack such a theoretical foundation. The perturbations applied (noise, quantization, resampling) are not guaranteed to erase the unique signal patterns that leak through the sensing pipeline. True anonymization implies a formal privacy model or guarantee—neither of which is satisfied here.

D. Where do we go from here?

Given the dual limitations of attacks (e.g., open-world generalization) and countermeasures (e.g., utility degradation), the path forward requires a shift in perspective:

- **Rethinking the threat model.** As attackers are bounded by realistic data access and generalization constraints, defenses should prioritize protection under *limited observation settings*, where privacy risk is highest.
- **Targeting the source.** We need a deeper understanding of the *physical origin* of MSMDF traits—how manufacturing variations or firmware processing shape the fingerprint, as for denoising filters in camera fingerprinting [15].
- **Context-aware and platform-level defenses.** Future countermeasures should combine *context-aware privacy filters* that adapt signal fidelity to app needs with platform-enforced constraints—such as coarser readings or minimum noise levels—to limit attack vectors.

VIII. Conclusion and limitations

Motion-based fingerprinting poses a persistent privacy risk. Our study consolidates prior work and shows that strong fingerprints stem from specific sensor-stream choices, with tree-based classifiers scaling best. Current countermeasures offer limited protection, often harming utility more than privacy. Future work should explore defenses rooted in adaptive mechanisms responsive to context and threat level.

As a systematization paper, we acknowledge potential limitations in our methodology that align with known pitfalls in ML evaluations [16]. Our dataset, though diverse, may carry sampling bias due to participant variability and uneven device representation. We mitigate data snooping by enforcing non-overlapping temporal splits, but further work could explore stricter session-level separation. Lastly, while our analysis focuses on device-specific traits, the risk of spurious correlations with user behavior cannot be fully excluded—underscoring the need for more principled approaches to distinguish physical-layer fingerprints from behavioral artifacts.

Acknowledgments

This work was supported by the European Research Council (ERC) under the consolidator grant MALFOY (101043410).

References

[1] T. Hupperich, D. Maiorca, M. Kührer, T. Holz, and G. Giacinto, "On the robustness of mobile device fingerprinting: Can mobile users escape modern web-tracking mechanisms?" in *ACM ACSAC*, 2015.

[2] A. Kurtz, H. Gascon, T. Becker, K. Rieck, and F. Freiling, "Fingerprinting Mobile Devices Using Personalized Configurations," *PETS*, 2016.

[3] J. Zhang, A. R. Beresford, and I. Sheret, "Factory calibration fingerprinting of sensors," *IEEE Transactions on Information Forensics and Security*, 2021.

[4] A. Das, G. Acar, N. Borisov, and A. Pradeep, "The web's sixth sense: A study of scripts accessing smartphone sensors," in *ACM CCS*, 2018. [Online]. Available: https://doi.org/10.1145/3243734.3243860

[5] A. Berdich, P. Iosif, C. Burlacu, A. Anistoroaei, and B. Groza, "Fingerprinting smartphone accelerometers with traditional classifiers and deep learning networks," in *IEEE SACI*, 2023.

[6] S. Dey, N. Roy, W. Xu, R. R. Choudhury, and S. Nelakuditi, "Accelprint: Imperfections of accelerometers make smartphones trackable." in *NDSS*, 2014.

[7] A. Das, N. Borisov, and M. Caesar, "Tracking mobile web users through motion sensors: Attacks and defenses." in *NDSS*, 2016.

[8] A. Das, N. Borisov, and E. Chou, "Every move you make: Exploring practical issues in smartphone motion sensor fingerprinting and countermeasures," *PETS*, 2018.

[9] Z. Ding and M. Ming, "Accelerometer-based mobile device identification system for the realistic environment," *IEEE Access*, 2019.

[10] X. Li, H. Liu, L. Zhang, Z. Wu, Y. Xie, G. Chen, C. Wan, and Z. Liang, "Finding the stars in the fireworks: Deep understanding of motion sensor fingerprint," *IEEE/ACM Transactions on Networking*, 2019.

[11] C. Sulbaran Fandino, A. J. Kouam, and K. Rieck, ""msmdf: Motion sensor fingerprinting dataset with 1,200 annotated samples from 42 smartphones across diverse conditions," May 2025. [Online]. Available: https://doi.org/10.5281/zenodo.15554712

[12] C. Dwork, "Differential privacy," in *Automata, Languages and Programming*. Springer Berlin Heidelberg, 2006.

[13] A. Das, N. Borisov, E. Chou, and M. H. Mughees, "Smartphone fingerprinting via motion sensors: Analyzing feasibility at large-scale and studying real usage patterns," 2016.

[14] K. Choi, "Sensor logger," 2024, version 1.36.1, Accessed: 2024-08-02.

[15] M. Kirchner, *Sensor Fingerprints: Camera Identification and Beyond*. Springer Singapore, 2022.

[16] D. Arp, E. Quiring, F. Pendlebury, A. Warnecke, F. Pierazzi, C. Wressnegger, L. Cavallaro, and K. Rieck, "Dos and don'ts of machine learning in computer security," in *USENIX Security*, Aug. 2022, pp. 3971–3988.