A Survey on Fingerprinting Technologies for Smartphones Based on Embedded Transducers

Adriana Berdich, Bogdan Groza, *Member, IEEE*, and René Mayrhofer

***Abstract*—Smartphones are a vital technology, they improve our social interactions, provide us a great deal of information, and bring forth the means to control various emerging technologies, like the numerous IoT devices that are controlled via smartphone apps. In this context, smartphone fingerprinting from sensor characteristics is a topic of high interest not only due to privacy implications or potential use in forensics investigations but also because of various applications in device authentication. In this work we review existing approaches for smartphone fingerprinting based on internal components, focusing mostly on camera sensors, microphones, loudspeakers, and accelerometers. Other sensors, i.e., gyroscopes and magnetometers, are also accounted, but they correspond to a smaller body of works. The output of these transducers, which convert one type of energy into another, e.g., mechanical into electrical, leaks through various channels such as mobile apps and cloud services, while there is little user awareness on the privacy risks. Needless to say, miniature physical imperfections from the manufacturing process make each such transducer unique. One of the main intentions of our study is to rank these sensors according to the accuracy they provide in identifying smartphones and to give a clear overview on the amount of research that each of these components triggered so far. We review the features which can be extracted from each type of data and the classification algorithms that have been used. Last but not least, we also point out publicly available data sets which can serve for future investigations.**

***Index Terms*—Accelerometer, data sets, gyroscope, loudspeaker, magnetometer, microphone, smartphone fingerprinting.**

I. Introduction and Motivation

SMARTPHONE usage is on a continuously increasing slope, as proved by many recent industry reports. More and more people are using smartphones for video calls, digital health, education services, financial services, agriculture services, etc. Not least, the Covid-19 crisis from the recent years, imposing lockdown restrictions and social distancing, had a severe social and economic impact but seems to have also led to an increase in smartphone usage [1]. The online market and consumer data platform Statista places the number of mobile devices in 2022 at 15.96 billion, expecting 18.22 billion by 2025, out of which 20% will have 5G connectivity [2]. As expected, in this context, smartphone security and user privacy are continuously gaining importance. Last but not least, smartphones are a key technology in controlling various Internet of Things (IoT) devices that improve the quality of our life and productivity in smart homes or offices.

Nowadays, smartphones have overwhelming computational power and memory resources, they are equipped with many sensors, such as camera sensors, microphones, accelerometers, magnetometers, gyroscopes or radio frequency sensors (e.g., NFC, UWB, GPS, etc.) but also with actuators such as loudspeakers. Generally speaking, these can be referred to as transducers, i.e., devices that convert from one type of energy to another, electrical into mechanical (in the case of loudspeakers) or the reverse (in the case of microphones), etc. Each transducer has unique characteristics, caused by imperfections in the manufacturing process, which can be used for fingerprinting the mobile device. However, device fingerprinting based on the unique features of the embedded transducers is not always straightforward due to various environmental conditions such as noise, temperature, etc., which can affect the fingerprint. This makes the deployment of noninteractive device authentication mechanisms, based on such fingerprints, more challenging. Consequently, there are a lot papers addressing smartphone fingerprinting. In this survey we analyze existing works targeting each type of transducer and we outline various features of the signals that are used, the clustering methodology and the results, also pointing on the number of devices that were used and the publicly released data sets.

Brief Depiction of Smartphone Transducers: In Fig. 1, we show a disassembled Samsung Galaxy J5 which is a commonly used mid-range smartphone. We used this device to illustrate various sensors, i.e., front/back camera, microphone, and accelerometer and also the loudspeaker, which is technically an actuator that converts electrical energy into sound. As mentioned, both sensors or actuators, as devices that convert one form of energy into another can be referred to as transducers. The Samsung Galaxy J5 was also used to extract data for the specific needs of this article in order to give a more accurate depiction on the statistical properties of the fingerprints. We extracted data from its camera sensors, loudspeakers, and accelerometers and we were forced to use a Samsung Galaxy S6 for microphone data since the J5 did not have a replaceable microphone (the microphone could be replaced only with

Manuscript received 21 December 2022; revised 26 March 2023 and 29 April 2023; accepted 15 May 2023. Date of publication 19 May 2023; date of current version 8 August 2023. This work was supported by the Project "Network of Excellence in Applied Research and Innovation for Doctoral and Postdoctoral Programs/InoHubDoc," Project co-funded by the European Social Fund Financing under Agreement POCU/993/6/13/153437. *(Corresponding author: Bogdan Groza.)*

Adriana Berdich and Bogdan Groza are with the Faculty of Automatics and Computers, Politehnica University of Timisoara, 300006 Timisoara, Romania (e-mail: adriana.berdich@aut.upt.ro; bogdan.groza@aut.upt.ro).

René Mayrhofer is with the Institute of Networks and Security and the LIT Secure and Correct Systems Lab, Johannes Kepler University Linz, 4040 Linz, Austria (e-mail: rm@ins.jku.at).

This article has supplementary downloadable material available at https://doi.org/10.1109/JIOT.2023.3277883, provided by the authors.

Digital Object Identifier 10.1109/JIOT.2023.3277883

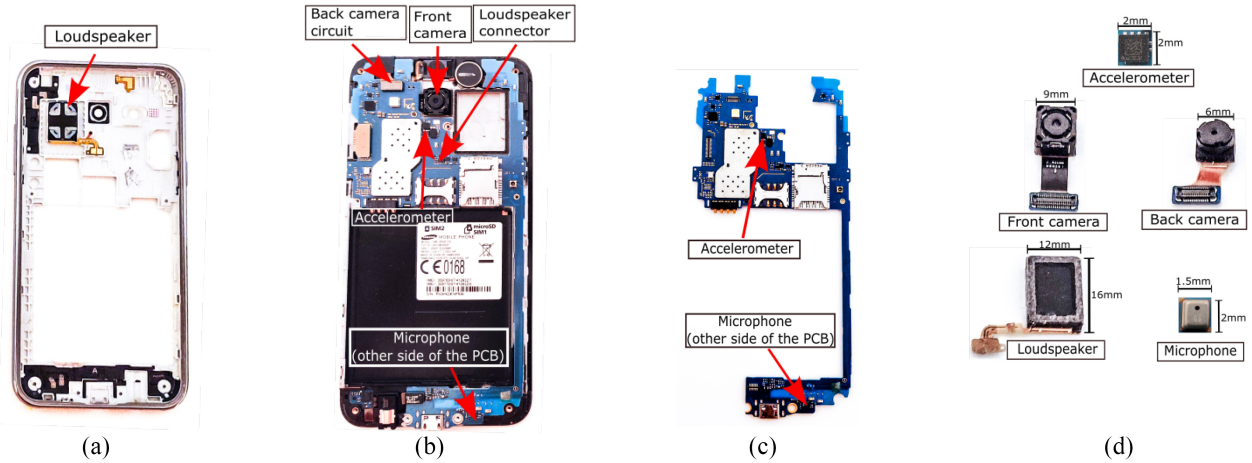


Fig. 1. Disassembled Samsung Galaxy J5: (a) back-case with loudspeaker, (b) display and circuit board, and (c) main circuit board and (d) five main transducers (accelerometer, front camera, back camera, loudspeakers, microphone).

TABLE I
NUMBER OF THE WORKS BY TOPIC IN THIS SURVEY

no.	Transducer	no. of papers (smartphones/other devices)
1.	camera	27/40
2.	microphone	21/6
3.	loudspeaker	5
4.	accelerometer	13
5.	gyroscope	7
6.	magnetometer	5
7.	other sensors or characteristics	14

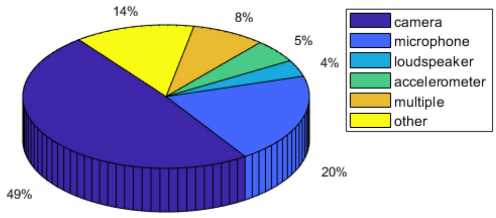


Fig. 2. Distribution of the works we survey by topic.

the smartphone mainboard). In the following sections, as a practical example, to determine the distance between fingerprints collected from identical devices (also referred to as the intradistance), we use either five identical Samsung J5s phones or, alternatively, we couple different transducers to the same device. Further, to determine the distance between fingerprints collected from different devices (also referred o as the interdistance), we use several smartphones from different manufacturers.

Distribution of Works by Topic: Generally speaking, there are two main types of fingerprints: 1) software-based fingerprints and 2) hardware-based fingerprints. In this work, we are concerned with the latter, i.e., hardware-based fingerprints. This is because they use characteristics of the transducers embedded on the circuit board that are more difficult to replace—on the one hand, making the fingerprint harder to forge, but on the other hand also creating higher privacy risks as such fingerprints can carry over between different mobile applications, use cases, and even operating system reinstalls.

There are a lot of papers published in the recent years addressing mobile device identification based on their sensors characteristics. In this work, we survey more than 130 papers. To give an accurate figure, in Table I, we list all sensor fingerprints that have been exploited so far and the number of papers covered by this survey (papers using multiple sensors are counted once for each sensor). In Fig. 2, we give an overview of the analyzed papers. Almost half of them discuss device identification based on the camera sensor, 20% of them discuss smartphone identification based on their microphone and only 5% of them discuss smartphone fingerprinting based on their loudspeaker. About 4% of the works discuss fingerprinting based on accelerometer sensors and 8% discuss device fingerprinting based on multiple sensors, i.e., accelerometers, magnetometers, and gyroscopes. Last but not least, 14% of the analyzed papers discuss device fingerprinting based on other, less commonly used sensors, e.g., magnetometers and gyroscopes, or even battery consumption, etc.

Several surveys on smartphone fingerprinting have been already published. A study published in 2015, regarding mobile phone fingerprinting, discusses the use of the network layer, i.e., IP and Internet control message protocol (ICMP) packets, as well as the application layer, i.e., browsers or mobile apps [3]. The work also mentions some countermeasures against fingerprinting. A later work, from 2017, addresses smartphone identification based on physical fingerprints [4]. The authors survey distinct techniques for fingerprinting starting with techniques based on signals emitted by smartphone components and processed by external systems, i.e., radio frequency, medium access control (MAC), display, clock differences, then they pursue techniques based on sensor identification, i.e., camera sensors, microphones, magnetometers. Finally, the authors discuss some risks and countermeasures for smartphone fingerprinting. In the same year, i.e., 2017, a study regarding fingerprinting algorithms, e.g., ratio and relational distance, K-nearest neighbor (KNN), thresholding, Gabor filters, etc., was published in [5]. A short study from 2019 analyzes research papers which are focusing

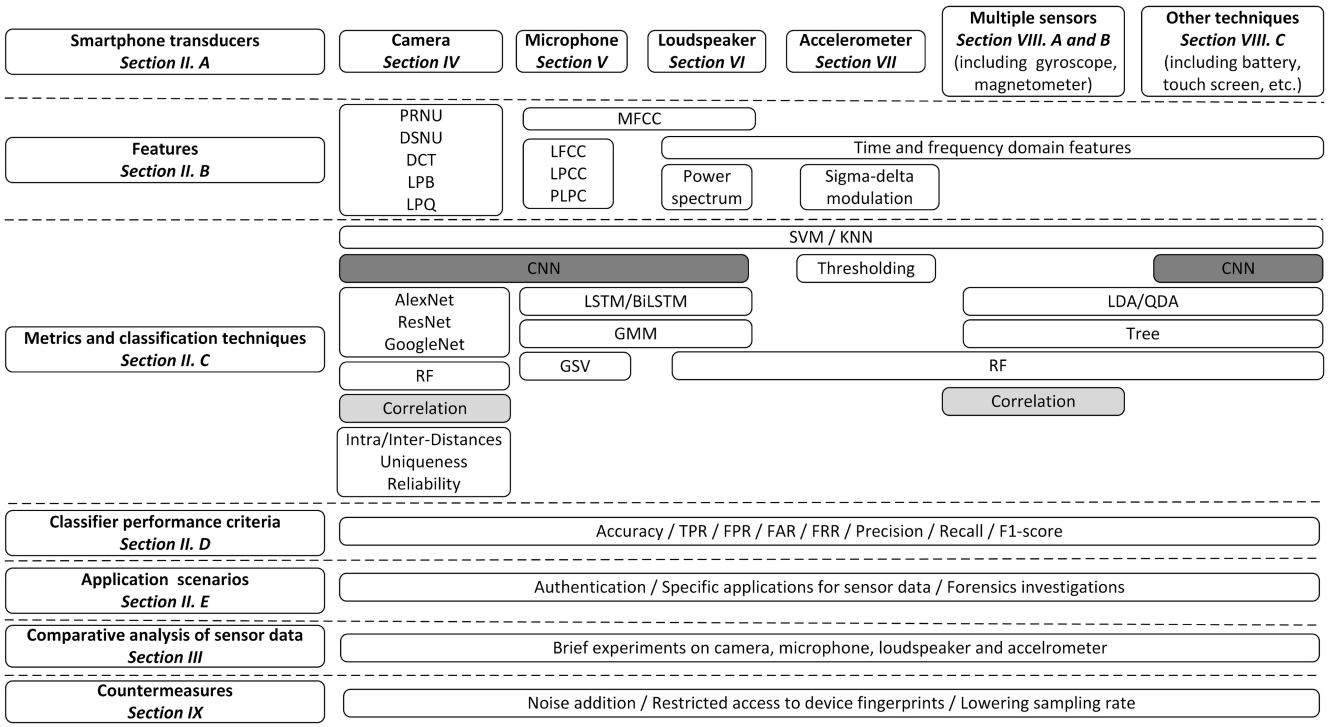


Fig. 3. Smartphone fingerprinting technologies and the structure of our work.

TABLE II
COMPARISON OF EXISTING SURVEYS ON DEVICE FINGERPRINTING

no.	work	year	target device	use-cases	counter measures	dataset	experimental comparison
1.	[3]	2015	smartphone	n	y	n	n
2.	[4]	2017	smartphone	y	y	n	n
3.	[5]	2017	any	n	n	n	n
4.	[6]	2019	smartphone	n	n	n	n
5.	[7]	2020	smartphone	n	n	n	n
6.	[8]	2021	IoT	y	y	y	n
7.	this work	2022	smartphone	y	y	y	y

on smartphone identification based on their accelerometers, cameras, loudspeakers and wireless transmitters [6]. One year later, in 2020, another study dedicated to smartphone fingerprinting was published in [7], investigating device identification based on various fingerprints, i.e., IMEI, MAC, serial numbers, or based on internal circuits, i.e., sensors and memory defects. Several techniques used for identification, machine learning, physical unclonable function (PUFs) and sensor calibration are discussed. More recently, in 2021, a survey of device fingerprinting focusing on IoT devices was published in [8]. The authors discuss data sources, techniques for device identification, application scenarios, and data sets. In Table II, we briefly compare the previous surveys. Compared to these, our work is more focused on smartphones fingerprinting and also adds the existing data sets into discussion. We also provide a brief experimental analysis to outline the differences between the most commonly employed sensors.

Roadmap to Our Work: In Fig. 3, we provide a graphical overview of smartphone fingerprinting technologies which can be regarded as a roadmap for the current survey. Our work is organized as follows. In Section II, we discuss the operation principles for smartphone transducers, the most commonly used features and classification techniques, performance metrics, and some application scenarios. In Section III, we briefly present some concrete experimental data for cameras, microphones, loudspeakers, and accelerometers. These topics can be retrieved from the subsections on the left side of Fig. 3. Then, the upper side of Fig. 3 shows the structure of our work with respect to smartphone transducers: Section IV addresses cameras, Section V addresses microphones, Section VI addresses loudspeakers, and Section VII addresses accelerometers. Next, in Section VIII, we survey some papers which propose device identification based on the mixed use of the previous sensors, possibly with other sensors as well. In Section IX, we discuss some countermeasures and the stability of fingerprints in front of external factors. Finally, in Section X, we conclude our work.

II. BACKGROUND

In this section, we present the sensor fingerprinting procedure, starting from the operation principles of sensors, then discuss the most common techniques for feature extraction, the classification algorithms, and metrics. Last but not least, we present some application scenarios.

A. Operation Principles for Smartphone Transducers

In what follows, we briefly discuss the operation principle for the aforementioned smartphone transducers, i.e., camera sensors, microphones, loudspeakers, and accelerometers.

1) Operation Principle of Camera Sensors: There are two commonly used types of sensors: 1) charge-coupled device

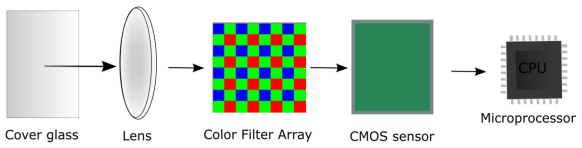


Fig. 4. Operation principle of camera sensor.

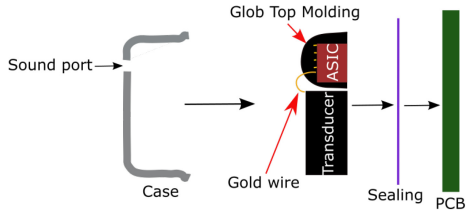


Fig. 5. Operation principle of MEMS microphone (redrawn based on https://www.digikey.be/nl/articles/how-mems-microphones-aid-sound-detection).

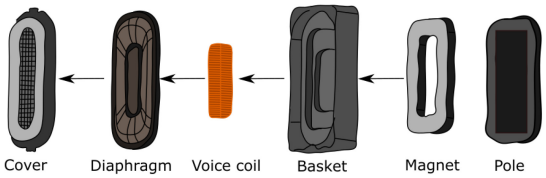


Fig. 6. Operation principle of a MEMS loudspeaker (redrawn based on [9]).

(CCD) and 2) complementary metal–oxide–semiconductor (CMOS) sensors. CCD sensors are used for digital cameras and systems which need to acquire high-quality images. CMOS sensors are smaller and consume less power, so they are typically used in small-size devices, e.g., smartphones, laptops, IoT devices, etc. [10]. In Fig. 4, we depict the operation principle of a CMOS sensor. The light captured by the lens goes into a Bayer filter array which parses the light into three components red, green, and blue. Half of the filter elements are green because the human eye is more sensitive to green, the other two elements are for red and blue. Finally, the light is transformed into an electrical signal by the CMOS sensor.

2) Operation Principle of Microphones: Smartphones are equipped with microelectromechanical systems (MEMSs) microphones due to their low power consumption, low costs, and small dimensions. In Fig. 5, we show the components of a MEMS microphone. The microphone is enclosed in a case with a small opening that facilitates the reception of sound. Inside the case, there are two main components: 1) a transducer used to convert the acoustic signal into an electrical signal and 2) an application-specific integrated circuit (ASIC) which amplifies the signal received from the transducer and implements the analog digital converter (ADC) functionalities. The transducer is connected to the ASIC with a golden wire. To improve the quality of the received sound, a special sealing material is used to hermetically isolate the microphone. The printed circuit board (PCB) of the phone is depicted on the back of the sealing material.

3) Operation Principle of Loudspeakers: In Fig. 6, we depict the main components of a smartphone MEMS loudspeaker. The loudspeaker is covered by a sieve which protects

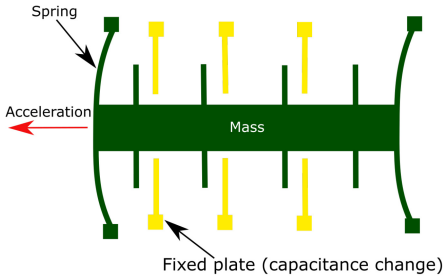


Fig. 7. Operation principle of MEMS accelerometer.

the diaphragm. The diaphragm is usually built from plastic (alternatively, it can be built from paper or aluminium) and allowed to move by the suspension, which is made from a flexible material and anchors it to the case (also called basket). After the diaphragm, a voice coil is present, which is fixed in the loudspeaker's case. Behind it, there is a pole and a magnet which make the voice coil vibrate, driven by the electromagnetic force, and so the diaphragm generates sound.

4) Operation Principle of Accelerometer Sensors: In Fig. 7, we depict the operation principle of MEMS accelerometers. The accelerometer contains a moving beam structure which has a fixed solid plane and a mass on springs. When an acceleration is applied, the mass is moving and the capacitance between the fixed plane and the moving beam changes.

B. Frequently Used Features for Device Fingerprinting

We now give a brief summary of the most common techniques for feature extraction that facilitate smartphone identification from data produced by the aforementioned transducers.

1) Time and frequency-domain features can be extracted for all kinds of sensor data.
   a) The most commonly used statistical features of the *time-domain* representation are the following: mean, standard and average deviation, skewness (asymmetry), kurtosis (tailedness), root mean square (RMS), maximum and minimum values, zero-crossing rate (ZCR), nonnegative count, variance, mode and range, etc.
   b) The most commonly used features of the *frequency-domain* representation are the spectral centroid, spread, skewness, kurtosis, entropy, flatness, brightness, roll-off, roughness, irregularity, RMS, flux, attack time, attack slope, mean, variance, standard deviation, low energy rate, and dc component from discrete cosine transform (DCT).

   Various time and frequency-domain features are used in [11], [12], [13], [14], [15], [16], [17], [18], and [19]. An exhaustive list of the features would be out of scope.
2) *Features Extracted From Camera-Collected Images:*
   a) Fixed-pattern noise (FPN) is the noise generated by the sensor which makes some pixels brighter than the average intensity. Based on the image type, there are two types of FPN: i) dark signal nonuniformity (DSNU) [20] which appears in the absence

of light (dark images) and ii) photograph response nonuniformity (PRNU) [21], [22], [23], [24], [25], [26], [27], [28], [29], [30], [31], [32], [33], [34], [35], [36], [37] which appears in conditions when light is present. PRNU is the most used technique for camera identification [38].

b) DCT is a common technique used to convert an image from the spatial domain to the frequency domain. In JPEG compression, DCT is applied on 8×8 image blocks, while for decompression the inverse DCT (IDCT) is used [39]. This transformation can be used with both DSNU and PRNU [40], [41].

c) Local binary pattern (LBP) and local phase quantization (LPQ) are another two features commonly used for processing images in the scope of camera identification [42]. LBP is a local texture pattern descriptor for images. The image is split in 3x3 blocks and the center pixel is considered the threshold for the neighbor pixels [43], [44]. LPQ is a descriptor based on the blur invariance from the Fourier phase spectrum extracted from images.

3) *Features Extracted From Audio Signals:*

a) The power spectrum, i.e., the frequency–amplitude pair obtained by applying the Fourier transform is the most basic method used to extract frequencies of the spectral estimates of the audio signal. Such features are commonly used for audio signals, in the scope of loudspeaker and microphone identification [45].

b) Mel-frequency Cepstral coefficients (MFCCs) are another commonly used feature for audio signals. This technique is used in several research works to extract features from human speech in the scope of microphone identification since these coefficients are frequently employed in speech recognition [46], [47], [48], [49], [50], [51], [52]. They have also been used for loudspeaker identification [9], [53], [54]. To extract the MFCC coefficients, the audio signals is split into windows and for each such window the fast fourier transform (FFT) is computed. The Mel filter is applied to the result and the logarithm of each Mel frequency is computed to which the DCT is finally applied giving the MFCCs.

c) Linear frequency Cepstral coefficients (LFCCs) is a technique similar to MFCC, except that a linear filter is used instead of the Mel filter [48]. Linear predictive codes coefficients (LPCCs) and perceptual linear prediction coefficients (PLPCs) are also used for human speech analysis [46].

C. Metrics and Classification Techniques

In what follows, we give a brief summary of the most frequently used classification techniques for fingerprinting each of the previously mentioned smartphone components. Starting from some basic metrics up to deep learning, several approaches have been considered.

1) The Euclidean distance is used in [55] for loudspeaker identification. It is computed as the square root of the sum of squared differences between two samples: $\text{dist}(a,b) = \sqrt{\sum_{i=1}^{n}(a_i - b_i)^2}$, where a and b are the signals from two devices expressed as vectors, i.e., a_i is the ith sample from signal a, and b_i is the ith sample from signal b.

2) The Hamming distance defines the number of indices at which the corresponding symbols are distinct and it is given as: $d(s,t) = \sum_{i=1}^{n} |s_i - t_i|$, where s and t are signals (vectors) from two devices, s_i is the ith sample from signal s and t_i is the ith sample from signal t.

3) The Mahalanobis distance is the distance between a distribution and a sampling point. It is given by $d = \sqrt{(y-\mu)\text{cov}^{-1}(y-\mu)'}$, where y is a vector, μ is the mean value, and cov is the covariance.

4) The intra and interdistances are useful in separating between devices based on established distance metrics, e.g., such as the Euclidean or Hamming distance.

a) The intrachip distance is calculated as the arithmetic mean between fingerprints extracted at different times from the same chip. While this metric can be computed for any fingerprint, most commonly, it is used to evaluate PUFs, such as those based on CMOS sensor [56], [57], where the intrachip Hamming distance indicates the average number of flipped bits among the PUFs from different images. Also, the bit error rate (BER) can be calculated by the intrachip Hamming distances. The reliability can be also calculated based on intrachip Hamming distances. We define these according to [58]

$$\text{dist}_{\text{INTRA}} = \frac{1}{m}\sum_{j=1}^{m}\frac{\text{dist}(R_i, R_{i,j})}{n} \times 100\%$$

$$\text{BER} = \text{dist}_{\text{INTRA}}, \text{Reliability} = 100\% - \text{dist}_{\text{INTRA}}$$

where R_i is the correct PUF calculated from the average of all PUFs of the evaluated chip and $R_{i,j}$ is the PUF of the jth image, n is the number of bits, and m is the number of images.

b) The interchip distance describes the uniqueness of a PUF, which is calculated as the Hamming distance between the PUFs of two distinct chips. Again, according to [58], it can be defined as

$$\text{dist}_{\text{INTER}} = \frac{2}{m(m-1)}\sum_{u=1}^{m-1}\sum_{v=u+1}^{m}\frac{\text{dist}(R_u, R_v)}{n} \times 100\%$$

$$\text{Uniqueness} = \text{dist}_{\text{INTER}}$$

where R_u is the PUF of the uth chip, R_v is the PUF of the vth chip, n is the number of bits, and m is the number of images. The intra and interchip distance are used in various works, e.g., [20], [59], [60], and [61].

5) Thresholding is a known approach for image segmentation, i.e., to convert a gray-scale image into a binary one. It is also used for classification for various sensor

data. In the case of smartphone sensor fingerprinting, thresholding is mostly used within the scope of camera identification, both for feature extraction but also as a stand-alone method for classification [20], [28], [36], [56], [57], [60], [62]. This approach is also used for classification when other signals are involved such as accelerometers [19] or for various device properties [63].

6) Correlation, i.e., corr(x, y), is a function which describes a statistical relationship between two distinct variables x and y. It is computed as: $\text{corr}(x, y) = [(\text{cov}(x, y))/(\sigma_x \times \sigma_y)]$, where $\text{cov}(x, y)$ is the covariance of x and y, σ_x is the standard deviation of x and σ_y is the standard deviation of y. The correlation is used by many works for fingerprinting smartphones, such as [21], [22], [23], [24], [25], [26], [27], [62], [64], [65], [66], [67], [68], [69], and [70].
7) *Classical Machine Learning Approaches:*
   a) Support vector machine (SVM) is a supervised machine learning algorithm which can be used to train binary or multiclass models. SVM is a common classification algorithm and, based on the literature we surveyed, appears to be more commonly used for camera sensor identification [30], [33], [43], [44], [71], [72], [73], [74], [75], [76], [77] and microphone identification [47], [48], [51], [78], [79], [80], [81], [82], [83], [84], [85], [86], [87], [88]. Occasionally, it was also used for other transducers, e.g., accelerometers [12], [13] or loudspeakers [54].
   b) KNN is another commonly used supervised classification algorithm which is employed in the literature for smartphone identification based on various components, e.g., microphones [78], [79], [83], [84], [87], loudspeakers [9], [53], accelerometers [12], [13], etc. KNN usually employs the Euclidean distance between the training samples and the test samples.
   c) Gaussian mixture model (GMM) is a probability function defined as a sum of Gaussian component densities. GMM is recommended to be used in speech recognition tasks. For device sensor fingerprinting, GMM was used for microphone [46], [48], [49], [52] and loudspeaker-based identification [9], [53]. It seems to be particularly useful when the underlying signal is human speech.
   d) Gaussian supervector (GSV) is an algorithm based on GMM which concatenates all the means of the features from each Gaussian component into a supervector [89]. GSV was used for microphone identification based on human speech [82], [85].
   e) Random forest (RF) is an ensemble classifier algorithm that can employ different methods for classification, including AdaBoost learners, Bagged Trees, Subspace Discriminant, RUSBoost Trees, Subspace KNN, and GentleBoost. RF was used for accelerometer identification [12], [13], camera identification [40], [76], [90], loudspeaker identification [54], and smartphone recognition based on multiple sensors [16], [17], etc.
   f) Decision tree is another supervised machine learning algorithm, the data is structured as a tree in which the internal nodes store the features from the data sets. Branches contain the decision rules and leaf nodes, which are the end nodes, represent the outputs. This technique was used for smartphone identification based on magnetometer [18], gyroscope [91], multiple sensors [14], [15], [92], etc.
   g) Linear discriminant analysis (LDA) and quadratic discriminant analysis (QDA) are supervised machine learning algorithms based on the Gaussian distribution. LDA uses linear Gaussian distributions, i.e., it creates linear boundaries between classes and QDA uses quadratic Gaussian distributions, i.e., it creates nonlinear boundaries between classes. LDA was used for microphone identification [93], smartphone identification based on wireless charging [14], and for smartphone identification based on magnetic induction emitted by the CPU [94]. QDA was used for smartphone recognition based on accelerometer and gyroscope data [14], [15].
8) *Deep Learning Approaches:*
   a) Convolutional neural networks (CNNs) are deep learning algorithms which are commonly used to extract patterns from images, but they were also used for audio data and various other time-domain series. In terms of device identification based on their sensors, CNNs are mostly used for camera sensors [33], [34], [37], [42], [77], [95], [96], [97], [98], [99], [100], [101], [102], [103], microphones [83], [84], [87], [104], [105], [106], loudspeakers [45], [54], as well as for other signals such as peripheral input timestamps [107]. AlexNet is a CNN proposed in 2012 for image classification [108]. AlexNet can be used as a pretrained neural network which contains five convolutional layers, max-pooling layers, three fully connected layers, and a soft max layer. It was used for camera sensor identification in [95]. GoogLeNet is another CNN with 22 layers proposed in 2015 [109]. GoogLeNet is used for camera sensor identification in [37] and [95]. Residual neural network (ResNet) was introduced in 2016 [110]. Based on the number of layers there are several types of ResNet, e.g., ResNet18 which contains 18 layers, ResNet50 with 50 layers, and ResNet101 with 101 layers. RetNet is a pretrained neural network, but it can be adapted. It is used for camera sensor identification as a pretrained network as well as an adapted network [35], [100], [111], [112].
   b) Long short-term memory (LSTM) and Bidirectional LSTM (BiLSTM) are recurrent neural network layers used for time series and sequence data. They were used

for microphone [106] and loudspeaker identification [45], [54]. The results in [45] show that their performance is comparable to the CNN for loudspeaker identification.

D. Commonly Used Performance Criteria for Classifiers

We now give an overview of common performance metrics used in the literature. The first metrics are commonly used and it makes no sense to point to specific papers that use them. In the next section, we will give some concrete results corresponding to these metrics.

1) All of the following metrics are expressed based on the following quantities: the true positives TP, false negatives FN, true negatives TN, and false positives FP.
2) Accuracy is the ratio between the number of correctly identified items and the total number of items: $\text{Accuracy} = (\text{TP} + \text{TN})/(\text{TP} + \text{FP} + \text{TN} + \text{FN})$. The validation accuracy can be also computed as: $\text{Accuracy} = 1 - kfoldLoss$, where $kfoldLoss$ is the classification error using k-fold cross validation.
3) Precision is the ratio of items correctly classified as positive: $\text{Precision} = \text{TP}/(\text{TP} + \text{FP})$.
4) The recall, or true-positive rate (TPR), is the ratio of correctly identified items out of all items that actually belong to the positive class: $\text{Recall} = \text{TPR} = \text{TP}/(\text{TP} + \text{FN})$.
5) True-negative rate (TNR) is the ratio of classified items that are genuinely negative: $\text{TNR} = \text{TN}/(\text{FP} + \text{TN})$.
6) $F1$-score, also referred to as the F-measure, is the harmonic mean between the precision and recall: $F1 = (2 \times Precision \times Recall)/(Precision + Recall)$.
7) False acceptance rate (FAR) is the ratio of negative items classified as positives: $\text{FAR} = \text{FP}/(\text{TN} + \text{FP})$.
8) False rejection rate (FRR) is the ratio of positive items classified as negatives: $\text{FRR} = \text{FN}/(\text{TP} + \text{FN})$.

Other metrics which are rarely used include the purity [113] and the adjusted rand index (ARI) [22], [113], [114].

E. Application Scenarios

There are many areas that can benefit from smartphone fingerprinting technologies, including include device authentication, various day-by-day applications, and even forensics investigations. We discuss each of them next.

1) Authentication: Device authentication and multifactor authentication based on a transducer fingerprint can minimize user interaction and reduce the vulnerabilities caused by weak security tokens, such as passwords. The unique fingerprint may act as one factor in user (or device) authentication which is specifically important for IoT applications where devices may not have a user interface or cannot be easily accessed (e.g., they are placed in an inconvenient location) while fast and secure authentication mechanisms are needed. There are various works which use the device fingerprints in the scope of authentication as we outlined next.

Generic Device Authentication: The PUFs extracted from camera sensors are proposed for authentication by using PRNU patterns [23], the DSNU, or FPN [57]. Live streaming surveillance footage is used for authentication in [61]. Microphones and loudspeakers are used in [115] for smartphone identification by exploiting the frequency response of a speaker-microphone pair belonging to two wireless IoT devices (this offers an acoustic hardware fingerprint). Audio signals with frequencies between 4 and 20 kHz, having an increment of 400 Hz, are emitted by a smartphone and recorded by another one while authentication relies on the correlation of the signals. Microphone fingerprints based on ambient sounds were also proposed for authentication [116]. Accelerometer fingerprints were proposed in a Web-based multifactor authentication scheme [19]. Some works have merged between data from multiple sensors, such as accelerometer, gyroscope, and camera for a robust smartphone authentication [92]. Also, acceleration, the magnetic field, orientation, gyroscope sensors, rotation vector, gravity, and linear acceleration are used in [16] to extract smartphone fingerprints for authentication in the context of Web applications. The hardware fingerprint of IoT sensors has been used for secret-free authentication in [117]. Authentication schemes for smartphones and IoT devices were also recently surveyed in [118].

Specific Environments for Authentication: Some works have been more specific regarding the exact area of application. One specific scenario which seems to be more interesting are the vehicular environments. In [45], smartphone fingerprinting is performed from data recorded by in-vehicle infotainment units. The smartphone emits a linear sweep between 20 Hz and 20 kHz while the infotainment unit records the sounds. Also, Anistoroaei et al. [119] proposed an in-vehicle authentication protocol between the smartphone and the infotainment unit. Specific acceleration patterns in various transportation environments have been also studied in the scope of device-to-device authentication [120].

2) Specific Applications for Sensor Data: In what follows, we show some positive use cases of sensor data but we must emphasize that exposing this data adds privacy risks for users as well. Some works have considered activity or transportation mode recognition based on accelerometer patterns [121], [122], [123]. Besides activity recognition, the accelerometer and other sensors were used for daily life monitoring and health recommendations [124]. Driving style recognition and driver behavior classification [125], [126], [127], [128], [129] is another application from which car rental services or insurance companies may benefit. The accelerometer data has been used for road condition monitoring [130], real-time pothole detection [131], or gait recognition [132]. Data from motion sensors has been also proposed for theft detection [133]. IoT sensor fingerprints are also commonly used for detecting attacks, unauthorized firmware modifications or fault diagnosis [8]. Another application mentioned in a recent survey is sensor quality control [4].

Privacy Concerns: Smartphone fingerprints can be exploited for tracking users which is a serious privacy concern. Motion sensors, i.e., accelerometers, have been used for tracking users [11], tracking metro riders [134], and detecting activities from the metro station [135]. Other works discuss preventing

privacy risks for distinct data, e.g., cameras [68] or loudspeakers [9], [55]. Smartphone operating systems are increasingly concerned with the exploitation of sensor data by apps for device fingerprinting and user tracking purposes. As a consequence, additional restrictions to accessing such (meta) data are being added.

3) Forensics Investigations: A complementary topic is forensics investigations. Microphones [86], [106] and cameras [33], [77], [99], [113] have been commonly discussed in the context of forensics investigations since they can be used for finding (suspected) criminals by recognizing their smartphones based on the sounds or images recorded in connection with the respective crime [136], [137]. Anti-forensics techniques have been discussed to falsify the source of audio signals by adding specific noise [51]. Another recently emerged topic is combating the dangerous effects of the AI. Machine learning techniques are already being employed to create deepfake audio or video recordings. These applications use deep learning to create very realistic recordings [138]. This technology can be used to manipulate the public opinion by creating fake news or for public persons defamation, which endangers national security and can be used as a tool by the organized crime.[1] To combat the dangerous effects of deepfake applications, deepfake detection algorithms are (currently) not very efficient, but source camera identification can be used to improve the results [139], [140]. By using unique fingerprints extracted from cameras or microphones, deep fakes could potentially mitigated by creating an end-to-end trust chain to the raw sensor data.

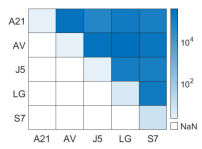
III. Brief Comparative Analysis of Sensor Data

To bring a clearer image on the quality of data retrieved from smartphone transducers, in this section we briefly present some concrete results. As an experimental basis, we compare data from five distinct and five identical smartphones.

A. Brief Experiments With Smartphone Camera Identification

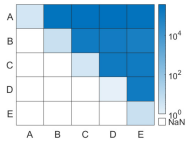
We now evaluate the interdistances for five distinct devices (Samsung Galaxy S7, Samsung Galaxy A21s, Allview V1 Viper I, LG Optimus P700, and Samsung Galaxy J5) and the intradistances for five identical devices (Samsung Galaxy J5). We select only the green channel because it has more encoding power, i.e., there are two green pixels for every red and blue pixel, and filter each image using a `wiener2` filter. To extract the DSNU from each image, we compute the difference between the original image and the filtered image. The noise which results is used to compute the Euclidean distances between devices. To clarify the computation, the distance between two distinct images is computed as $\sqrt{\sum_{i=1}^{4\,458\,240}(a_i - b_i)^2}$ where a_i and b_i are the DSNU coefficients extracted by the DCT transform (see [141] for details). The 4 458 240 values correspond to the number of coefficients that can be extracted from a 1920×2322 pixel matrix.

[1]https://lionbridge.ai/articles/deepfakes-a-threat-to-individuals-and-national-security/



	A21	AV	LG	J5	S7
A21	6.0	255110.0	50683.0	137790.0	94949.0
AV	-	4.0	287460.0	349400.0	219050.0
LG	-	-	6.0	115010.0	94129.0
J5	-	-	-	6.0	152130.0
S7	-	-	-	-	4.0

Fig. 8. Camera sensor: Mean of the Euclidean distance for distinct devices.



	A	B	C	D	E
A	7.0	309980.0	328500.0	364340.0	294420.0
B	-	1.0	142310.0	121970.0	90819.0
C	-	-	3.0	151710.0	145360.0
D	-	-	-	1.0	129060.0
E	-	-	-	-	3.0

Fig. 9. Camera sensor: Mean of the Euclidean distance for identical devices.

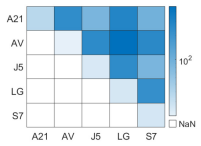
1) Distinct Smartphones: We captured 50 dark images with each device. Since devices may have different resolutions, we consider only the top left corner from each image leading to images of equal sizes, i.e., 1920×2322. In Fig. 8, we show the results as a heatmap (left) and numeric values (right). The values form the main diagonal are clearly much lower than the rest, which means that devices can be easily identified.

2) Identical Smartphones: In the case of identical devices, we used the data set from [141] which contains 50 dark pictures captured by five identical Galaxy J5 cameras. To compute the distance for a single smartphone we split the data set into two distinct data sets, i.e., one with 25 pictures chosen randomly and another one with the rest of 25 pictures. In Fig. 9, we show the results as a heatmap (left) and numeric values (right). The values form the main diagonal are lower than the rest of the values which means again that the devices can be identified correctly with ease.

B. Brief Experiments With Microphone Identification

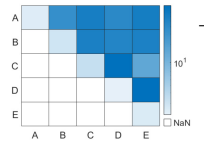
Using the public data set from [93], we evaluate the interdistances for five distinct devices (Samsung Galaxy S7, Samsung Galaxy A21s, Allview V1 Viper I, LG Optimus P700, and Samsung Galaxy J5) and the intradistances for five identical devices (Samsung Galaxy S6 smartphones). We use the live recordings of hazard lights to separate between distinct devices and the prerecorded vehicle's horn sound to separate between identical devices, according to the public data set from [93]. From each recorded sound we extract the power spectrum which is used to compute the mean of the Euclidean distances between devices. Each file contains 4096 samples which correspond to a frequency range between 0 and 22 050 Hz at a resolution of 5.384615 Hz (which results in 4096 sampling points). Therefore, the distance between two microphone samples is computed as: $\sqrt{\sum_{i=1}^{4096}(a_i - b_i)^2}$ where a_i and b_i are the power spectrum coefficients (amplitudes) for the two microphones represented as real numbers (floating points). The values of these coefficients were usually in the range of 0 to 70 db.

1) Distinct Smartphones: The data set in [93] contains 500 measurements with distinct devices of hazard lights sound for which we compute the mean of the Euclidean distances (between each pair of smartphones). To compute the distances



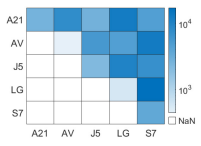
	A21	AV	LG	J5	S7
A21	36.685	237.030	86.545	283.460	96.181
AV	-	20.016	246.530	461.160	258.840
LG	-	-	23.928	239.850	87.086
J5	-	-	-	24.985	221.630
S7	-	-	-	-	25.611

Fig. 10. Microphones: Mean of the Euclidean distance for distinct devices.



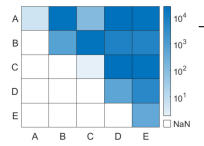
	A	B	C	D	E
A	1.452	34.246	59.393	41.723	60.829
B	-	1.896	55.341	47.725	53.456
C	-	-	2.346	91.308	15.217
D	-	-	-	1.270	92.006
E	-	-	-	-	1.546

Fig. 11. Microphones: Mean of the Euclidean distances for five identical devices (50 measurements).



	A21	AV	LG	J5	S7
A21	2651.2	7908.9	2720.8	12479.0	4649.1
AV	-	441.5	5545.4	4570.2	12558.0
LG	-	-	2196.9	10116.0	7012.7
J5	-	-	-	572.9	17128.0
S7	-	-	-	-	3588.4

Fig. 12. Loudspeakers: Mean of the Euclidean distance for distinct devices (four measurements).



	A	B	C	D	E
A	5.1	22335.0	109.7	25392.0	22343.0
B	-	675.5	22430.0	7137.8	6082.9
C	-	-	1.9	25488.0	22439.0
D	-	-	-	541.3	4601.5
E	-	-	-	-	416.5

Fig. 13. Loudspeakers: Mean of the Euclidean distances for five identical devices.

for a single smartphone, we split the data set into two distinct data sets each of 250 measurements selected randomly and extract the distances between the two. In Fig. 10, we show the results as a heatmap (left) and numeric values (right). The values from the main diagonal are lower than the rest of the values. While the differences are smaller than in the case of camera sensors, the microphones can still be clearly separated.

2) Identical Smartphones: For this case, the data set in [93] contains 50 measurements with identical microphones of the same Samsung Galaxy S6 which records a car honking sound generated by a Hi-Fi system. To compute the distance for identical devices, we split the data set in two random sets of 25 measurements. In Fig. 11, we depict the mean of the Euclidean distances between each pair of smartphone microphones. Again, the devices separate clearly as the values from the main diagonal are lower than the rest of the values.

C. Brief Experiments With Loudspeaker Identification

Using the public data set from [45], we compute the interdistances for five distinct smartphones and the intradistances for five identical Samsung Galaxy J5 smartphones. The data set contains a linear sweep between 20 Hz and 20 KHz played by the smartphones and recorded by an infotainment head-unit. The distance between the smartphones and the head-unit was 1 m. To evaluate the interdistances for distinct devices (Samsung Galaxy S7, Samsung Galaxy A21s, Allview V1 Viper I, LG Optimus P700, and Samsung Galaxy J5), we performed five additional measurements with each smartphone in the same circumstances as in the data set from [45]. For each recorded sound we extract the power spectrum, which is used to compute the mean of the Euclidean distances between devices. Each file contains 1914 samples which correspond to a frequency range between 700 Hz and 11 kHz with a resolution of 5.384615 Hz. The distance between two samples is $\sqrt{\sum_{i=1}^{1914}(a_i - b_i)^2}$ where a_i and b_i are the power spectrum coefficients (amplitudes) for the two speakers represented as real numbers (floating points).

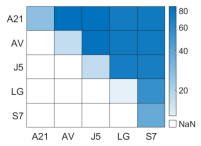
1) Distinct Smartphones: We select five measurements in a random order and compute the mean of the Euclidean distances between each pair of smartphones. To compute the distance for a single smartphone we split the data set into two equal data sets containing random samples. In Fig. 12, we show the results as a heatmap (left) and numeric values (right). Again, the values from the main diagonal are lower than the distances between distinct devices. Compared to microphones, the distances are more variable which suggests that microphones are a better alternative for classification (still, not as good as camera sensors).

2) Identical Smartphones: The data set contains 100 measurements with identical microphones for the same Samsung Galaxy J5 smartphone. To compute the distance for the same device, we randomly split the data set in two equal subsets. In Fig. 13, we depict the mean of the Euclidean distances between each pair of smartphone loudspeakers. The distance between the smartphones A and C is lower than the values from the main diagonal, which means that the loudspeaker C was misidentified as A and vice versa. This suggests that simple inter and intradistances are not enough for separating between loudspeakers. Indeed, for a better separation between two loudspeakers, the work in [45] has used two deep neural networks: 1) a BiLSTM and 2) a CNN.

D. Brief Experiments With Accelerometer Identification

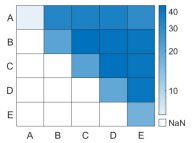
Now, we evaluate the interdistances for distinct devices (Samsung Galaxy S7, Samsung Galaxy A21s, Allview V1 Viper I, LG Optimus P700, and Samsung Galaxy J5) and intradistances for five identical devices (Samsung Galaxy J5). We collected data at a sampling rate of 10 ms in an environment with constant vibrations. The data is scaled and aligned to have the same amplitude and also time-aligned to compute the Euclidean distance. The amplitudes on each axis are squared, summed and the square root extracted to get the overall amplitude, i.e., $a = \sqrt{a_X^2 + a_Y^2 + a_Z^2}$. The distances between devices are computed on subsets of 5000 elements. To compute the intradistance, we choose several samples, split them in four subsets of the same size, and we compute the mean of the Euclidean distances between two subsets randomly selected. The distance is thus computed as $\sqrt{\sum_{i=1}^{5000}(a_i - b_i)^2}$ where a_i and b_i are the amplitudes.

1) Distinct Smartphones: In Fig. 14, we show the results as a heatmap (left) and numeric values (right). In the case of interdistances, the values from the main diagonal are lower than the rest of the values, allowing some separation, though not



	A21	AV	LG	J5	S7
A21	26.75	86.20	84.49	75.37	75.45
AV	-	17.01	83.05	75.70	74.12
LG	-	-	17.20	72.40	71.19
J5	-	-	-	12.72	54.27
S7	-	-	-	-	37.00

Fig. 14. Accelerometers: Euclidean distances for five distinct devices.



	A	B	C	D	E
A	6.49	32.78	34.54	34.97	29.81
B	-	20.93	44.43	42.68	41.08
C	-	-	21.20	43.68	41.29
D	-	-	-	19.51	41.21
E	-	-	-	-	16.91

Fig. 15. Accelerometers: Euclidean distances for five identical devices.

as clear as in case of any of the previous transducers (camera sensors, microphones, and loudspeakers).

2) Identical Smartphones: In Fig. 15, we show the results for identical smartphones as a heatmap (left) and numeric values (right). In the case of intradistances, again the values from the main diagonal are lower than the rest of the values, but the intradistances are slightly reduced. This suggests the same conclusion that accelerometer imperfections can be used to separate between devices, but likely produce a poorer separation compared to other transducers.

E. Overall Interpretation of Heatmap Data

The previously presented heatmaps with data collected from all four sensor show significant differences. We now try to briefly clarify why it is so. Smartphone camera sensors give a significantly higher amount of information compared to other sensors, i.e., microphones, loudspeakers, or accelerometers. Concretely, the resolution of the images was 1920×2322 pixels for the cameras that we used (or we cropped the image to this size in case of higher resolutions), while each pixel encodes 24 bits of information (1 byte for each color R, G, B). This leads to a matrix of 1920×2322 bytes for each color on which we compute the Euclidean distances. That is, the Euclidean distance is computed as a sum of more than four million values and unsurprisingly leads to values in the order of hundreds of thousands, as can be seen in Figs. 8 and 9. In the case of loudspeakers and microphones, the audio signal is in the range of 20 Hz–20 kHz and we extract the power spectrum from it which yields a vector of 1914 coefficients expressed as 24-bit floats. Therefore, when we compute the Euclidean distances, this is done over a vector of less than 2000 values and results in a much smaller sum compared to camera sensors, generally in the order of tens of thousands at most as can be seen in Figs. 12 and 13. For accelerometers, the sampled data is on 24 bits (8 bits for each axis) and we choose a vector of 5000 elements. However, as done in most previous works and explained previously, we normalized the data on the three axis in order to avoid orientation issues by extracting the square root from the sum of squared accelerations, which technically reduces the 24-bit data to at most 9 bits. Therefore, the Euclidean distance is even smaller, less than 100 as can be seen in Figs. 14 and 15. Clearly, in the case of all sensors, the value of the Euclidean distances will depend on the specific inputs and the previous discussion only tries to clarifies what should be expected in general.

Another observation is that the intradistances may seem unexpectedly higher in case of the identical speakers from Fig. 13, but this is easily explainable. Smartphone loudspeakers are electromechanical devices that consist of a coil and a plastic diaphragm which may be affected over time by various environmental factors. The speakers from the data set that we used come from disassembled smartphones that had several years of use in different conditions. Aging is very likely why the interdistances vary so much between otherwise identical loudspeakers. Regarding the number of measurements, in the data set from [45] that we used, in case of different smartphone models, only five measurements were made since the differences were quite obvious and the separation immediate. In the case of identical speakers, 100 measurements were needed to make the separation clearer since the results were much closer [45]. This may also contribute to the variations.

The same information about the statistical distances is also suggestive about the effectiveness of each fingerprint type. Clearly, images are the most effective for fingerprinting due the large amount of information that a sensors captures and because an image can be taken in an instant. Second to this are microphone and loudspeaker data, but this may require seconds or more of collected data. For example, in the experiments from [45], a sweep signal took about 10 s, in the experiments in [93], a car honking took about 1 s, hazard lights took about 2 s, wipers took about 3 s, etc. Accelerometers seem to be the least effective as previous works used 30 s [11] or 3 s per sample [12], etc. Regarding the efficiency of the fingerprinting process, it is worth mentioning that some scenarios may call for high efficiency. One such example is the advertisement ecosystem, where users may access the websites only for brief moments of time and a fast response is needed in order to create unique user profiles and recognize them. Aspects related to the advertisement ecosystem are mentioned in various fingerprinting works like [3], [11], [13], [142], [143], [144], and [145]. Other apps may not require a fast fingerprint extraction since they have access to sensor data for prolonged periods of time, such as various e-health, social media, or communication apps.

IV. Mobile Device Identification Based on Camera Sensors

In this section, we survey works on device identification from camera sensors. In Fig. 16, we show an overview on the camera identification techniques and the amount of works that has been done through the years. Almost half of the surveyed papers use machine learning algorithms, including deep learning techniques. A large number of these works, about 17%, proposes PUFs, while 35% use other techniques, e.g., thresholding, correlation, etc. The past three years account for more than half of the publications we survey. In Table III, we compare the features, classifiers, results, number of devices,

TABLE III
OVERVIEW OF VARIOUS WORKS WHICH IS PROPOSED FINGERPRINTING SMARTPHONES BASED ON THEIR CAMERA

no.	work	year	feature	classifier	results	devices	dataset
1.	[64]	2006	PRNU	correlation		9	n/a
2.	[71]	2006	lens radial distortion	SVM	91%	3	n/a
3.	[65]	2007	peak signal to noise ratio (PSNR)	correlation		4	n/a
4.	[21]	2008	PRNU and Maximum Likelihood	correlation		6	n/a
5.	[146]	2008	Dust spots, Gaussian intensity loss model and shape properties	computing a confidence value for each matching dust spot	99.1%	4	n/a
6.	[43]	2012	LBP	multi-class SVM	98%	18	Dresden
7.	[147]	2013	SPN, high pass	compare with SPN database		5	n/a
8.	[30]	2013	PRNU + wavelet transform	SVM	87.214%	14	n/a
9.	[56]	2014	FPN	thresholding	uniqueness 50.12%, reliability 100%	na	n/a
10.	[57]	2015	FPN	thresholding	uniqueness 49.37%, reliability 99.8%	5	n/a
11.	[29]	2015	PRNU	Hierarchical Search using MapReduce	precision 0.91	1174	Flickr
12.	[95]	2016	highpass filter	CNNmodels,compared with AlexNet and GoogleNet	CNN:91.9%, AlexNet:94.5%, GoogleNet:83.5%	33	own + Dresden
13.	[28]	2016	PRNU + LADCT	threshold		23	own + Dresden
14.	[72]	2016	LPQ and LBP	multi-class SVM	100%	14	Dresden
15.	[148]	2016	DVS based PUF	-			
16.	[66]	2017	SPN and correlation	ADMM and spectral clustering	F1 97%	31	Raise + Dresden
17.	[22]	2017	PRNU	correlation		39	Dresden
18.	[23]	2017	PRNU	correlation		14	n/a
19.	[40]	2017	DCT + PCA	RF based ENS	99.1%	10	Dresden
20.	[73]	2017	Radial Basis Kernel	SVM	>99%	7	own + Dresden
21.	[74]	2017	I-Vector	SVM	99.01%	8	Dresden
22.	[59]	2017	SRAM PUF			20	no
23.	[60]	2018	optical (photonic crystal)	-		14	n/a
24.	[20]	2018	DSNU	thresholding		10	n/a
25.	[114]	2018	linear dependencies among SPN	Large-scale sparse subspace	F1 92%	107	Dresden + Vision
26.	[75]	2018	average pixel value	SVM	87.6	27	Dresden
27.	[112]	2018	social network (facebook)	ResNet50	96%	5	Dresden
28.	[41]	2018	DCT	SEA+HF	TPR 88.54	57	Dresden
29.	[31]	2018	PRNU	-	TPR 70.55%	57	Dresden
30.	[42]	2018	LBP and LPQ features	CNN	99.5%	10	SPCUP
31.	[96]	2018	split image	CNN	100%	74	Dresden
32.	[149]	2019	inherent information	camera fingerprint ordering	F1 90%	53	Dresden
33.	[113]	2019	SPN approximation	Markov and Hybrid Clustering	F1 86.6%	35	Vision
34.	[32]	2019	PRNU	external and (CVIs)		142	Dresden + criminal
35.	[24]	2019	G-PRNU	2D cross correlation	100%	35	Vision
36.	[25]	2019	PRNU	correlation	F1 81.8%	7	Dresden
37.	[97]	2019	-	CNN	98%	3	Miche-1
38.	[98]	2019	-	CA-CNN	97.37%	74	Dresden
39.	[99]	2019	CNN based denoise	binary classification	F1 44.4%	125	own + Dresden + Socrates + Vision
40.	[100]	2019	multi-scale HPF	CNN+ ResNet	84.3%	125	own + Dresden
41.	[76]	2019	transfer learning	SVM, Logistic Regression, and RF + CNN	100%	5	their PUBLIC
42.	[26]	2019	PRNU	correlation		25	Dresden
43.	[77]	2019	Supervised pipeline (rich features, CFA and CNN derived features)	WSVM, DBC, SSVM, PISVM and OSNN	98.68%	312	Dresden + ISA + Flickr
44.	[101]	2019	top left corner	max(log(PCE),0)	n/a	90	yes
45.	[150]	2019	DATASET	correlation			
46.	[62]	2019	adaptive thresholding	correlation		72	own
47	[34]	2019	PRNU	CNN	> 80%	87	Dresden + Vision
48	[33]	2019	PRNU and noiseprint extracted by CNN	SVM, LRT and FLD	95.2%	625	own (public) + Dresden + Vision
49.	[67]	2019	Spatial Domain Averaged (SDA)	correlation	TPR > 83.1%	78	Vision + Nyuad-MMD
50.	[90]	2019	Discrete wavelet transform (DWT)	BN, L, LMT, MLP, NB, NBM, RF, SL, SVM	99.25%	4	IITD-I + AMI + WPUT + AWE
51.	[61]	2019	DVS based PUF	-	R >9%		n/a
52.	[44]	2020	WLBP texture descriptor	SVM	99.52%	9	Dresden
53.	[35]	2021	PRNU	RESNet101 + SVM	99.58%	28	Vision + Warwich + Daxing + own
54.	[151]	2021	-	EfficientNet	99.1%	27	FODB - yes
55.	[36]	2021	SPN and PRNU	thresholding	95.03%	34	Vision
56.	[68]	2021	Gaussian blurring and removing LSB	correlation	correlation< 0.075	48	own + Dresden
57.	[102]	2021	Remnant Block	CNN +RemNet	100%	18	Dresden
58.	[27]	2021	PRNU	peak to correlation ratio	FPR>5%	70	Flickr
59.	[152]	2021	demosaicing residual	Ensemble	98.14%	68	Dresden
60.	[111]	2021	patchwise mean, variance scoring and K-means clustering	Res2Net	92.62%	74	Dresden
61.	[153]	2021	-	MCIFFN	97.14%	73	Dresden
62.	[103]	2021	demosaicing	CNN	>99%	35	Vision
63.	[37]	2021	PRNU	CNN, GoogleNet, SqueezeNet, Densenet201 and Mobilenetv2	F1 >91%	18	own + Vision
64.	[141]	2022	DSNU	NN, LD, ENS, SVM, NB, KNN	97%	6	own

and data sets used in related works starting from 2006. In the results column of Table III, we generally refer to the accuracy reported by the works. However, not all of the works have reported the accuracy and in this case, we refer to other metrics as stated in the table or in the accompanying text.

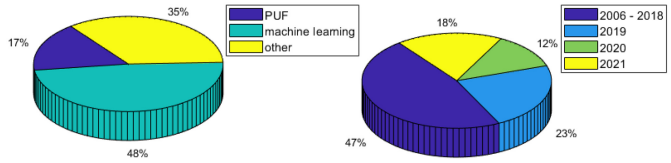


Fig. 16. Overview of the camera identification techniques (left) and research evolution (right).

A. PUF-Based Approaches

PRNU noise is used in [64] to build a PUF from camera sensors. The authors validate their proposed method using 320 images from nine cameras and use the correlation function as classifier. In terms of results, they obtain a FRR between 1.36×10^{-1} and 4.41×10^{-14} depending on the applied correction factor and JPEG compression. PRNU is also used in [23] for camera identification. The noise is removed from the images by applying a high-pass filter and then the high frequencies are used to obtain the camera fingerprints. The authors use 14 cameras, i.e., one digital single-lens reflex (DSLR) and 13 smartphones, to validate the approach and the resulting correlation for full images is between 0.0022 and 0.02. A different approach based on dust spots from images captured by DSLR cameras is proposed in [146]. Dust spots are detected using the shape properties and a Gaussian identity loss model. For the experiments, the authors use four cameras and, to cluster them, a confidence value based on occurrence, smoothness, and shift validity metrics for each dust spot is computed. The identification reaches 99.1% accuracy.

Specific PUFs for distinct technologies for CMOS sensors are proposed in the literature. A PUF for 65-nm CMOS sensors using hardware changes is proposed in [56]. A thresholding technique is used to validate the method and results are obtained at temperature fluctuations between 0 °C and 100 °C with a uniqueness of 50.12% and a reliability of 100%. Another PUF based on FPN is proposed in [57]. To validate the results, five chips of 180 nm camera sensors are used and for clustering the thresholding approach is applied. At temperature variations between 15 °C and 115 °C, the uniqueness is 49.37% and the reliability 99.80%. Zheng et al. [148] proposed an event-driven PUF for 1.8-V 180-nm CMOS sensors based on dynamic vision sensor (DVS). At temperature fluctuations between –35 °C and 115 °C the uniqueness is 49.96% and the reliability in between 96.3% and 99.2%. Another PUF for 180-nm CMOS sensors based on DVS is discussed in [61]. A reliability greater that 98% is obtained at temperature variations between –45 °C and 95 °C. An optical PUF for 65-nm CMOS sensors based on FPN is proposed in [60]. The experiments are performed on 14 CMOS sensors and to validate the method thresholding and 1-D autocorrelations are used. The authors obtain an interchip Hamming distance of 49.81% and intrachip Hamming distance of 0.251%.

A PUF for smartphone CMOS sensors based on DSNU is proposed in [20]. The image is denoised after which the DCT is applied, high-frequencies are extracted and then the IDCT is applied. Finally, the thresholding method is applied to remove bright pixels. The approach is validated on five identical sensors from two distinct smartphones and the obtained interchip Hamming distance is between 46% and 54% while the intrachip Hamming distance is lower than 10%. An PUF based on camera sensor SRAM is proposed in [59]. The average intrachip Hamming distance is 0.51% and the average interchip Hamming distance is 49.95% for 20 devices.

B. Machine Learning Approaches

A significant number of papers addressing identification with machine learning techniques are using the SVM classifier. The lens radial distortions are used in [71] as features for the SVM classifier. For three cameras the SVM classifier reaches an accuracy of 91%. Also, the multiclass SVM is used in [43], but the features are extracted based on LBP. The average accuracy reaches 98% for 18 cameras. PRNU and the wavelet transform are the features used by the SVM classifier in [30]. The average accuracy reached for 14 cameras models from 5 manufactures is 87.214%. LPQ and LBP are also used as input for the SVM classifier in [72]. For 14 camera models, the accuracy in between 98.13% and 100%. SVM with radial basis kernel is used in [73]. In the experiments, three distinct cameras are used, and the overall prediction accuracy is grater than 99%. Also, in [74], an accuracy of 99.01% is reached for eight camera models using the SVM classifier. For the green and red channels of the images, the authors extract an I-Vector using the LBP. A coupled feature representation is used as input for the SVM classifier in [75]. For 27 cameras, the identification accuracy reaches 87.6%. Weber's and LBP (WLBP) features are discussed in [44]. The features are translated in a vector which is used as input for the SVM classifier again. This method reaches 99.52% accuracy for nine cameras.

Also, deep learning algorithms are used in several research works. CNN, AlexNet and GoogleNet are used in [95] for camera identification. The images are first filtered using a high-pass filter and then deep learning algorithms are applied. For 33 cameras the accuracy is 91.9% in the case of CNN, 94.5% in the case of AlexNet, and 83.5% in the case of GoogleNet. A CNN based on features extracted using the LBP and LPQ is proposed in [42]. For ten camera models, the accuracy is between 84.1% and 99.5%. In [96], the images are split into k patches using sliding windows and the extracted features are used as input for a CNN. With this approach, the authors reach an average accuracy close to 100% for 74 cameras. CNNs were also used for source camera identification in [97]. Yang et al. [98] built a content-adaptive CNN (CA-CNN). The detection accuracy achieved is between 89.56% and 97.37% for 74 cameras. A method for source camera identification using images from Facebook is proposed in [112]. The authors propose a deep learning neural network based on an existing ResNet50 network. The network is tested with photographs from five cameras which are uploaded to Facebook and then downloaded back. The maximum classification accuracy was 96%.

A CNN is used in [99] to extract the noise of the images. For 125 cameras, the $F1$-score is between 0.205 and 0.444 and the average precision is between 0.144 and 0.399. Transfer learning and CNN are used in [76] for feature extraction while for camera identification, machine learning algorithms, i.e.,

SVM, logic regression (LR), and RF, are used. In the experiments, five cameras are classified with SVM as a final layer with 98.82% RANK-1 accuracy. With RF 97.16% RANK-1 accuracy was reached, while with LRs, 98.57% RANK-1 accuracy was reached. The RANK-5 accuracy was 100% for all the involved classifiers Ding et al. [100] used a multiscale high-pass filter (HPF) to remove the noise from the images. The authors use the multitask learning approach based on CNN and ResNet for camera clustering. This approach reaches 84.3% accuracy for 125 devices. In [77], a vector which contains features extracted using a statistical descriptor, color filter array (CFA), and CNN-derived is used as input for multiple classifiers: Weibull-calibrated SVM (WSVM), decision boundary carving (DBC), specialized SVM (SSVM), SVM with probability of inclusion (PISVM), and open-set nearest neighbors (OSNNs). The top-left corner of the images are used as input for a CNN in [101]. For 74 devices, the accuracy is between 0.943 and 0.961 for the same smartphone model and between 0.98 and 0.994 for the same brand. The accuracy unfortunately drops to 0.475 when a pool of 74 devices is used.

PRNU features and classification using CNN are discussed in [34]. In [33], a combination of PRNU and noise-print extracted by a CNN is used as feature, while for classification, the results from three classifiers are used: 1) SVM; 2) likelihood-ratio test (LRT); and 3) fishers linear discriminant (FLD). A maximum accuracy of 0.952 is reached with SVM. In [35], PRNU extracted from images is used as input for a neural network based on ResNet101 and SVM. For 28 devices, this approach reaches an accuracy of 99.58%. A neural network based on CNN, namely, EfficientNet, is discussed in [151]. For 23 000 images captured by 27 smartphones cameras, this neural network reaches a 99.1% accuracy. CNN and RemNet are used in [102]. This approach reaches a 97.59% accuracy for 18 distinct cameras. The use of the Ensemble classifier based on the demonsaicing residual features extracted from the CFA filter is discussed in [152]. The authors reach an average accuracy of 98.14% for the identification of 68 cameras. Also, in [103], the demosaicing approach for feature extraction is discussed. For clustering, a CNN is used which reaches an accuracy greater than 91% on 35 devices for WhatsApp images and 95% for YouTube scenes. Different pretrained CNNs, i.e., GoogleNet, SqueezeNet, Densenet201, and Mobilenetv2 are discussed in [37]. For 4500 images captured by 18 smartphones, the authors reach an $F1$-score greater than 91%. Features extracted using patchwise mean, variance scoring and K-means clustering are discussed in [111]. For classification, a Res2Net is used which reaches 92.62% accuracy for 74 cameras. A multiscale content-independent feature fusion network (MCIFFN) is discussed in [153].

In [40], the features are extracted from images using DCT and then the ensemble classifier is used. To improve the results, the authors also use principal component analysis (PCA). For 10 507 images captured by 10 cameras they reach an accuracy of 99.1%. In [90], features extracted by the discrete wavelet transform (DWT) are used with nine classifiers: 1) Bayes net (BN); 2) Logistic (L); 3) logistic model tree (LMT); 4) multilayer perceptron (MLP); 5) naive Bayes (NB); 6) NB multinomial (NBM); 7) RF; 8) simple logistic (SL) and 9) SVM. The average accuracy for the identification of four cameras is 99.25%.

C. Other Approaches

Adaptive thresholding is used in [62] for camera identification. For 74 cameras, the authors obtain an intercorrelation between 0.1 and 0.45 and intracorrelation between 0.46 and 0.7. Behare et al. [26] discussed camera identification based on PRNU using correlation. The experiments are done on 800 images from the Dresden database containing 25 distinct cameras.

Sensor pattern noise (SPN) and correction are discussed in [66] for camera identification. For clustering, the authors proposed an alternating direction method of multipliers (ADMMs) and spectral clustering. For 31 cameras, they obtain an $F1$-score between 0.90 and 0.97. PRNU and the locally adaptive DCT (LADCT) are used in [28] for camera identification. The authors use two data sets: their own data set with 13 cameras, for which they obtain an FNR between 5.46% and 21.27% and an FPR between 0.48% and 1.77%, and the Dresden data set with ten cameras for which they obtain an FNR between 0.93% and 14.11% and an FPR between 0.10% and 1.74%. SPN extracted from the green channel using a HPF is discussed in [147]. For five cameras, an FNR of 53% and an FPR of 10.75% were obtained. Also, in [36], SPN and PRNU are used to cluster 34 camera models. The features extracted from PRNU are used as input for a hierarchical search using MapReduce in [29]. For 1174, cameras a mean precision of 91% was obtained. The features extracted using the linear dependencies among SPN are used in [114] for camera identification using large-scale sparse subspace clustering. For 107 cameras, the precision is 0.92, recall is 0.88, the $F1$-score is 0.92, and ARI is 0.88. PRNU is also used in [22], [24], [25], [27], [31], and [32].

Rouhi et al. [113] used the SPN approximation for feature extraction while for classification, they use Markov clustering and a newly proposed hybrid clustering algorithm. For a data set with 35 smartphones, the precision is 0.997, the recall is 0.765, the $F1$-score is 0.866, the ARI is 0.863, and the purity is 0.997. A ranking index for the quality of each fingerprint is used in [149] to cluster cameras. For 10 960 images captured by 53 cameras, the precision is almost 1, the recall is between 0.65 and 0.85, and the $F1$-score is between 0.7 and 0.9.

In [41], using DCT, the low frequencies of SPN are removed from the images and the peaks are suppressed using the spectrum equalization algorithm high-frequency (SEA-HF). For 14 594 images from 57 cameras, the TPR is 88.54%. Spatial-domain averaged (SDA) frames are used in [67]. The peak-signal-to-noise ratio (PSNR) is used in [65] for camera identification. PRNU obtained using the maximum likelihood estimator is used in [21] for feature extraction from images. For six devices, with a FAR fixed at 10^{-5}, the FRR is between $9.6 * 10^{-2}$ and $8.4 * 10^{-15}$. In [68], a method based on Gaussian blurring and removing the least significant bit (LSB) from images is proposed. The authors obtain a correlation lower than 0.075 for 11 787 images captured with 48 cameras.

TABLE IV
OVERVIEW OF VARIOUS WORKS WHICH IS PROPOSED FINGERPRINTING SMARTPHONES BASED ON THEIR MICROPHONE

no.	work	year	signal	classifier	results	devices	dataset
1.	[78]	2009	synthetic sound	NB, SVM, trees and KNN	>93.5%	7	n/a
2.	[69]	2012	synthetic sound	inter-class cross correlation	100%	8	n/a
3.	[79]	2012	synthetic sound	GN, GNM, KNN, PCA, ISVM	recall between 0.7354 and 0.885	5	n/a
4.	[166]	2012	human speech	MFCC +SVM	96.42%		own (LIVE REC)
5.	[80]	2014	human speech	RBF-NN, the MLP and SVM	97.6%	21	own (MOBIPHONE)
6.	[46]	2014	human speech	LPCC, PLPC, MFCC + GMM	99.58%	16	own + TIMIT
7.	[47]	2014	human speech	GMM + SVM with SMO	90%	26	n/a
8.	[48]	2014	human speech	MFCC, LFCC + GMM and SVM	98.39%	14	TIMIT and LIVE RECORDS
9.	[145]	2014	synthetic sound	Maximum-Likelihood	95%	16	n/a
10.	[81]	2015	ambient noise	SVM	>96.72%	21	n/a
11.	[49]	2015	human speech	MFCC + GMM	99.27%	116	TIMIT
12.	[82]	2015	human speech	GSV, MFCC + SVM	error between 2.08% and 7.08%	14	LIVE database + TIMIT
13.	[70]	2016	-	mean, std, creast factor, dynamic range, autocorr	error: 1% - 3%	2	n/a
14.	[50]	2017	human speech	MFCC, GSV + NN	87.6%	21	MOBIPHONE, T-L-PHONE and SCUTPHONE
15.	[167]	2018	human speech	band energy difference descriptor	>96%	172	n/a
16.	[104]	2018	human speech	CNN	99.3%	40	own + MOBIPHONE
17.	[83]	2019	synthetic sound	SVM, KNN and CNN	around 96%	32	n/a
18.	[84]	2019	synthetic sound	SVM, KNN and CNN	CNN 80%, SVM 40%, KNN 10%	34	n/a
19.	[168]	2019	synthetic sound	artificial neural networks	100%	6	n/a
20.	[85]	2019	human speech	SVM, GSV, SRC	85.58%	4	Ahumanda
21.	[86]	2019	human speech	SVM-RFE and variance threshold	98.04%	24	own: CKC-SD, TIMIT-RSD
22.	[51]	2019	human speech	MFCC + SVM	97%	16	TIMIT-RD + LIVE-RECORD
23.	[52]	2019	human speech	MFCC + GMM	88.35%	7	n/a
24.	[116]	2019	ambient noise	RMS, ZCR, low energy rate, spec centroid, etc. + 3 classifiers	TPR: 81%, 98%	12	n/a
25.	[87]	2020	synthetic sound	KNN, SVM and CNN	>90%	34	n/a
26.	[105]	2020	human speech	CNN	99.56%	20	n/a
27.	[88]	2020	-	ENF + SVM	TP >60%	7	n/a
28.	[106]	2021	human speech	CNN, LSTM, CRNN	98%	4	KSU-DB
29.	[93]	2022	human speech (from [80]) and collected in-vehicle sounds	LD, ENS, TREE, KNN, SVM, CNN	100%	32	own

Meng et al. [154] and Gupta et al. [155] surveyed some works focused on camera source identification.

D. Data Sets for Camera Identification

The most commonly used data sets for camera identification are enumerated as follows.

1) Dresden [156] contains 14 000 images of various indoor and outdoor scenes captured by 73 digital cameras.
2) Vision [157] contains 34 427 images and 1914 videos in their original format and in their social network format, i.e., Facebook, YouTube, and WhatsApp, captured by 35 devices from 11 brands.
3) Warwick [158] contains more than 58 600 images captured by 14 cameras.
4) ISA UNICAMP [159] contains 3750 images from 25 cameras, i.e, 150 images per camera.
5) Daxing [150] contains 43 400 images and 1400 videos captured by 90 smartphones from 22 models and five brands.
6) SPCUP [160] is the IEEE Signal Processing Cup for camera model identification involving teams of undergraduate students, the challenge data set contains ten cameras and 200 images collected for each of them.
7) Michei [161] contains 3732 images from three smartphones.
8) FODB [151] contains 23 000 images of 143 scenes captured by 27 smartphone cameras.
9) Banna et al. [76] provided 3900 images from three camera models.
10) Cozzolino et al. [33] provide a data set which contains 21 158 images captured by 625 devices.
11) The work in [90] uses the data sets for ear biometrics from the following works: IITD-I [162], AMI [163], WPUT [164], and AWE [165] to test a wavelet-based camera identification method.

V. SMARTPHONE IDENTIFICATION BASED ON MICROPHONES

In this section, we survey works addressing mobile device identification based on their microphones. Table IV compares the features, classifiers, results, the number of devices and whether the data sets used in these works are public. We discuss them in detail in what follows. In the results column from Table IV, we generally refer to the accuracy reported by the works. As already stated, some works did not report the accuracy of their method and in this case, we refer to other metrics as presented in the table or in the accompanying text.

A. Microphone Identification Based on Synthetic Sounds

Distinct music genres, i.e., metal, pop, techno and instrumental, as well as sine waves and white noise are used in [78]. The Fourier coefficients are extracted from the recorded sounds and distinct classifiers are applied, i.e., NB, multi class SVM, decision trees and KNN. This approach was tested with 7 microphones and the highest accuracy was 93.5%. Ambient noise generated by a fan cooler is used for microphone identification in [69]. The authors use interclass cross correlation for clustering eight commercial microphones based on 24 recordings and reach a 100% correct classification. Indoor sounds, outdoor park areas, and street noises are used in [79] for microphone identification. One class classification algorithms, i.e.,

Gaussian model (GM), GMM, KNN, PCA, and incremental SVM (ISVM) are used to identify five microphones. In terms of results for indoor measurements, the recall is between 0.774 and 0.859, for park, noise is between 0.7354 and 0.885, and for street, noise is between 0.206 and 0.784. This was improved, using a representative instance classification framework (RICF) proposed by the authors, to get a recall between 0.741 and 0.874. In [81], a method based on FFT features extracted from ambient noise is discussed. For 21 devices, the maximum accuracy achieved is 96.72% with the SVM classifier.

Sine waves at 1 and 2 kHz are used in [83]. For the classification of 32 smartphones, the authors use SVM, KNN, and a CNN. They test the proposed approach at distinct signal-to-noise ratio (SNR) levels. The accuracy for a 20-dB SNR is 96% for the 1-kHz wave and 96.8% for the 2-kHz wave, while for 10-dB SNR, the accuracy drops at 67.27% for 1 kHz and 82.75% for 2 kHz. Also, the work in [84] uses sine waves at 1 kHz and the SVM, KNN, and CNN classifiers. For 34 smartphones, at 10-dB SNR, the accuracy reaches 80% for CNN, 40% for SVM, and 10% for KNN. In [87], in addition to the 1-kHz sine wave, a pneumatic hammer and gunshot sounds are also used. Hafeez et al. [168] generated 80 sine waves in the range of 100 Hz–8 kHz and then uses an artificial neural network with a single layer which achieves 100% accuracy for six commercial microphones.

Ambient sounds from distinct places, e.g., bus, food court, kids playing, metro, restaurant, etc., are used in [116]. The authors extract 15 features from the time and frequency domains, e.g., RMS, ZCR, low energy rate, spectral centroid, etc., and apply three binary classifiers in cascade. This approach was tested on 12 smartphones from two distinct models and the TPR reached 81% for one model and 98% for the other.

B. Microphone Identification Based on Human Speech

Three classifiers, i.e., radial basis functions neural network (RBF-NN), MLP, and SVM are used in [80] for smartphone microphone identification using the MFCC coefficients extracted from the human speech of 12 males and 12 females recorded with 21 smartphones. The highest accuracy, i.e., 97.6%, was reached with RBF-NN. The work in [46] uses GMM and the highest accuracy reached is 99.58%. The features they use are the LPCC, PLPC, and MFCC coefficients extracted from the speech of four speakers recorded with 16 microphones. Also, in [47], the MFCC coefficients extracted from human speech are used with the SVM classifier to cluster 26 smartphones. The accuracy achieved was 90%. The SVM classifier was optimized with the sequential minimal optimization (SMO) algorithm. In [48], MFCC and LFCC with GMM and SVM are used to cluster 14 smartphones. The achieved accuracy is 98.39%. In the case of 16 devices, by using the GMM and the MFCC coefficients extracted from human speech, the highest reported accuracy is 99.27% in [49]. In [82], GSV and MFCC are used to extract the features from human speech. For clustering, the SVM classifier is used and an error rate between 2.08% and 7.08% is reported for 14 devices.

Audio signal characteristics, such as mean, standard deviation, crest factor, dynamic range, and autocorrelation are used in [70] to fingerprint two identical microphones. Li et al. [50] used a neural network and Gaussian SVM for the identification of 21 smartphones based on their microphones. The features extracted with MFCC from human speech were used as input for the classifiers. The reported accuracy reaches 88.1%. A band energy descriptor is proposed in [167] as classifier. This approach reaches 96% accuracy for 170 devices which record human speech. In [104], 40 smartphones are identified with the highest achieved accuracy of 99% based on human speech using CNN. The voice from 25 speakers is used in [85]. GSV and the sparse representation-based classifier (SRC) reaches an accuracy between 78.17% and 85.58% for 4 microphones. Human speech is also used in [51], [52], [86], [105], [106], and [166].

A distinctive approach based on electrical network frequency (ENF) analysis is proposed in [88]. For seven devices, the TPR is above 60%.

C. Data Sets Used for Microphone Identification

The following data sets for microphone identification are publicly available.

1) TIMIT [169] is a speech database for voice recognition which contains 6300 sentences from 630 speakers, i.e., ten sentences from each speaker, 439 males, and the rest are females, recordings from this data set were also replayed and recorded by various works for smartphone recognition, e.g., [48], [49], and [82]. There are also several reissues of the TIMIT data set, such as TIMIT-RSD [86] which recaptured the data set with 24 smartphones.
2) MOBIPHONE [80] is a speech database which contains recordings did with 21 smartphones. For each smartphone there are 12 males and 12 females who read ten sentences. The speakers are selected from the TIMIT database.
3) T-L-PHONE [48], [166] contains speech recorded with 14 mobile phones from six brands.
4) SCUTPHONE [170] contains speech recorded with 15 distinct mobile phones from six brands.
5) Ahumanda [171] contains speech recorded by six devices from 150 males and 150 females.
6) CRC-SD [86] contains speech recorded by 24 smartphones from seven brands (6 males and 6 females).
7) KSU-DB [172] is a speech database which contains 136 speakers (68 males and 68 females) recorded with four devices in three environments.
8) Live recordings [166] containing 10-min speech from a single speaker, recorded with 14 smartphones.
9) The microphone fingerprinting data set from [93] contains 19 200 samples with 16 different and 16 identical devices that record various automotive specific sounds, e.g., car honk, tiers, wipers, hazard lights, etc.

VI. Smartphone Identification Based on Loudspeakers

In this section, we survey some works which discuss mobile device identification based on their loudspeakers. In Table V we compare the features, classifiers, results, number of devices

TABLE V
OVERVIEW OF VARIOUS WORKS WHICH IS PROPOSED FINGERPRINTING SMARTPHONES BASED ON THEIR LOUDSPEAKER

no.	work	year	signal	features	classifier	results	devices	dataset
1.	[55]	2014	cosine at 14-21kHz in 100Hz increment		Euclidean distance	err.: $1.55 * 10^{-4}\%$	50	n/a
2.	[9]	2014	instrumental, human speech, song	time and freq. domain features	KNN GMM	98.8%	19	n/a
3.	[53]	2014	instrumental, human speech, song	time and freq. domain features	KNN GMM	100%	52	n/a
4.	[54]	2018	human speech	MFCC and SSF (CQT)	SVM, RF, CNN, BiLSTM	99.29%	24	own
5.	[45]	2021	linear sweep 20-20.000kHz	frequency response, roll-offs of the power spectrum	linear approximations, KNN, RF, KNN, CNN, BiLSTM	100%	28	yes

and data sets that are used for smartphone identification based on loudspeakers. Compared to camera sensors and microphones, there are far less papers addressing this topic.

A. Loudspeaker Identification Based on Synthetic and Natural Sounds

Two types of sounds are used for loudspeaker fingerprinting: 1) synthetic sounds, such as cosine waves [55] and linear sweeps [45] and 2) natural sound, such as instrumental, music [9], [53], and human speech [9], [53], [54].

Zhou et al. [55] fingerprint 50 identical smartphones based on a cosine wave between 14 and 21 kHz, with an increment step of 100Hz, emitted by each loudspeaker. The smartphones are identified using the Euclidean distance and an error rate around $1.55 * 10^{-4}\%$ is reached. Berdich et al. [45], fingerprint 28 smartphones loudspeakers, out of which 16 are identical loudspeakers placed in the same smartphone case, using a linear sweep signal between 20 Hz and 20 kHz which is recorded by an in-vehicle head unit. In this work, the roll-off characteristics of the power spectrum are used. For classification, a linear approximation as well as machine learning algorithms, i.e., KNN, RF, and SVM and deep learning algorithms, i.e., CNN and BiLSTM, are used (the later two deep neural networks are the main subject of the investigation). An accuracy between 95% and 100% is achieved for identical smartphone speakers. In this work, the authors also analyzed the influence of the volume level and the speaker orientation angle in the fingerprinting process. For four distinct smartphones, the experiments are also done at 50%, 75% and 100% volume level, and the authors observe that the fingerprints for each smartphone are clustered around the volume level, but the smartphones can still be clearly identified. The same behavior was observed in the case of experiments for distinct loudspeaker orientation, i.e., 0°, 90°, and 180°.

A total of 15 features in the time and frequency domain i.e., RMS, ZCR, low energy rate, spectral centroid, spectral entropy, spectral irregularity, spectral spread, spectral skewness, spectral kurtosis, spectral rolloff, spectral brightness, spectral flatness, MFCCs, chronogram, and total centroid are used in [9] and [53]. The features were extracted from three types of sounds, i.e., instrumental, sound, and human speech. For classification, the authors use KNN and GMM classifiers. The experiments are done for both distinct and identical smartphones. In [9], for 15 identical smartphones the authors reach a 93% accuracy, while for 19 smartphones (identical and distinct), they achieve a 98.8% accuracy using the MFCC coefficients extracted from human speech. In [53], for 52 smartphones out of which at most 15 are identical, the authors achieved a 100% $F1$-score when they used the MFCC coefficients from each signal (instrumental, song and human speech) with the KNN classifier. When GMM on MFCC is used for instrumental sounds, the $F1$-score is 100%, while in the case of human speech and songs, the $F1$-score is 99.6%. From the 15 time and frequency-domain features used in both these papers, MFCC leads to the best results. MFCC and sketches of spectral features (SSFs) extracted from human speech are used in [54]. Machine learning algorithms, i.e., SVM and RF, as well as deep learning algorithms, i.e., CNN and BiLSTM are used to cluster 24 smartphones. The authors achieved a maximum accuracy of 99.29%.

B. Data Sets for Loudspeaker Identification

To the best of our knowledge, there is currently only a single public data set for smartphone identification based on their loudspeakers, which corresponds to the work in [45]. The data set contains linear sweep signals played by 28 smartphones (16 identical and 12 distinct) recorded by the a vehicle head unit at 1-m distance. A total of 2900 measurements are made public.

VII. SMARTPHONE IDENTIFICATION BASED ON ACCELEROMETERS

In this section, we survey several works which discuss device identification based on their accelerometer sensors. Interestingly, while there are a lot of papers which discuss device pairing based on data collected from accelerometers, only few works are focused on smartphone fingerprinting based on accelerometers. In Table VI, we compare the features, classifiers, results, and the number of devices that are used.

A. Time and Frequency-Domain Features for Accelerometer Fingerprinting

Besides smartphone identification based on their microphone (which was addressed previously), the authors from [145] also discuss smartphone identification based on accelerometer sensors. The measurements are collected when the smartphone is kept at a constant velocity or when it is in a resting position, and the first sample from each measurement is considered the smartphone fingerprint. With this approach, only 15.1% of devices were correctly identified.

Multiple time and frequency-domain features are used in [11] and [12], while in [19], only time-domain features are used. Dey et al. [11] used time-domain features i.e., mean, standard deviation, average deviation, skewness,

TABLE VI
OVERVIEW OF VARIOUS WORKS WHICH IS PROPOSED FINGERPRINTING SMARTPHONES BASED ON THEIR ACCELEROMETERS

no.	work	year	features	classifier	results	devices	dataset
1.	[145]	2014	measurements at constant velocity	first sample	58.7%	>543	yes
2.	[11]	2014	time and frequency domain features	-	precision, recall > 99%	107	yes
3.	[19]	2016	time and frequency domain features	thresholding	TPR 0.7444, FPR 0.0978	15	n/a
4.	[12]	2019	time and frequency domain features	SVM, KNN, LR, RF, Extra Tree, XGBoost	prec. 100%, recall 94.3%	7	n/a

kurtosis, RMS, lowest and highest values, and frequency-domain features i.e., spectral standard deviation, spectral centroid, spectral skewness, spectral kurtosis, spectral crest, irregularity-k and J, smoothness, flatness, and roll off. For 107 accelerometers (25 smartphones, two tablets, and 80 standalone accelerometers), the mean precision and recall are above 99%. In [12], ten time and ten frequency-domain features are used, i.e., mean, min, max, variance, standard deviation, most frequently occurring value, range, skewness, kurtosis, RMS, dc, spectral mean, spectral variance, spectral standard deviation, spectral spread, spectral centroid, spectral entropy, spectral skewness, spectral kurtosis, and spectral flatness. For classification, six classifiers are used: 1) SVM; 2) KNN; 3) LR; 4) RF; 5) extra tree; and 6) extreme gradient boosting (XGBoost) and for seven devices, the authors obtain a precision between 54.5% and 100% and a recall between 88.9% and 94.3%. Only eight time-domain features i.e., min, max, kurtosis, RMS amplitude, mean deviation, skewness, standard deviation, and mean were used in [19]. For classification, the thresholding approach was used, which reached a 0.7444 TPR and a 0.0978 FPR for 15 devices.

B. Data Sets for Accelerometer Fingerprinting

Bojinov et al. [145] reported a public website which holds accelerometer related data,[2] however, the website was not accessible at the time of writing this article. Also, Dey et al. [11] reported another data set but the link was again not functioning at the time of this writing.[3]

VIII. OTHER SENSORS AND TECHNOLOGIES FOR FINGERPRINTING

In this section, we briefly present other sensors which have been used for fingerprinting, as well as some combined approaches that used multiple sensors.

A. Other Sensors: Magnetometers and Gyroscopes

The time and frequency-domain features extracted from magnetometer sensors are used in [18] for smartphone fingerprinting. For classification, the SVM, KNN, and Bagged Tree classifiers were used. This approach reached an $F1$-score between 61.3% and 90.7% for nine smartphones. A more recent work [91] uses the gyroscope resonance for smartphone fingerprinting. Ten features based on resonance, e.g., resonance peak, position of resonance peak, etc., are extracted and used as input for decision trees and regression tree classifiers to cluster 20 smartphones and five gyroscope sensors. The highest accuracy reached with this approach was 96.5%.

B. Combined Approaches Based on Multiple Sensors

Rather than using single transducers, several research works discussed smartphone fingerprinting from multiple sensors. We address them separately in this section. Most of these works start from analyzing individual sensor data and then combine several sensors to improve the identification rate. In Table VII, we compare the features, classifiers, results, and the number of devices used in the literature for smartphone identification based on multiple sensors. We detail each of these works in what follows.

Amerini et al. [92] used data extracted from accelerometers, gyroscopes and cameras for smartphone identification. For accelerometers and gyroscopes they extract ten time-domain features and 11 frequency-domain features while for cameras they use the PRNU. In terms of classification, decision trees are used and 10 smartphones are clustered with an $F1$-score greater than 75% for combined data from accelerometer, gyroscope and camera. Data extracted from accelerometers, gyroscopes, magnetometers, and microphones are used in [17]. The authors extract for each sensor several features. In the case of accelerometers and magnetometers, they again extract ten time-domain and 11 frequency-domain features for the normalized signals, while in the case of gyroscopes, they extract the same features for each axis. For microphones, they generate sine waves between 100 and 1300 Hz and for each signal, the value of the dominant frequency is considered as a feature. The classification was done using the NB and RF machine learning algorithms and for ten devices, the authors reach an $F1$-score of 90% for the combined data.

Combined accelerometer and gyroscope data is also used in [13], [14], [15], [173], and [174]. Das et al. [13] and [14] used 25 time and frequency-domain features, while in [15], they use 26 features. Several machine learning algorithms are used in these works which include SVM, NB, KNN, decision tree, QDA, and bagged decision trees. In [173], the entropy features are extracted from the collected data and used as input for the SVM classifier. For three devices, the authors reach an accuracy greater than 90%. A multidimensional balls-into-bins model is proposed in [174] to extract the features from the collected data and then a multi-LSTM network is used to cluster the devices. For 117 devices from 77 users, this approach reaches an accuracy higher than 98.8%. Acceleration, magnetic field, orientation, gyroscope, rotation vector, gravity, and linear acceleration are used in [16]. Five sensor combinations are discussed: 1) individual accelerometers; 2) accelerometers and gyroscopes; 3) all

[2]http://sensor-id.com/
[3]http: //web.engr.illinois.edu/sdey4/AccelPrintDataSourceCode.html

TABLE VII
OVERVIEW OF VARIOUS WORKS WHICH IS PROPOSED FINGERPRINTING SMARTPHONES BASED ON MULTIPLE SENSORS

no.	work	year	signal	features	classifier	results	devices	dataset
1.	[14]	2015	accelerometer + gyroscope	25 time and frequency features	SVM, NB, MDT, KNN, QDA, BDT	F1 96%	30	n/a
2.	[15]	2016	accelerometer + gyroscope	26 time and frequency features	SVM, NB, MDT, KNN, QDA, BDT	F1 96%	30	n/a
3.	[92]	2016	accelerometer + gyroscope + camera	21 time and freq. features for acc. and gyro., PRNU for camera	decision tree	F1>75%	10	n/a
4.	[16]	2016	7 sensors: acceleration, magnetic field, orientation, gyroscope, rotation vector, gravity, linear acceleration	time and frequency features	KNN, SVM, B Tree, RF, Extra Tree	99.995%	5000	n/a
5.	[173]	2016	accelerometer + gyroscope	Threshold Entropy, Sure Entropy and Norm Entropyc	SVM	>90%	3	n/a
6.	[18]	2017	magnetometer	time and frequency features	SVM, KNN, Bagged Decision Tree	F1 90.7%	9	n/a
7.	[17]	2017	accelerometer + gyroscope + magnetometer + microphone	time and frequency features	NB, RF	F1 90%	20	n/a
8.	[13]	2018	accelerometer + gyroscope	time and frequency features	SVM, RF, Extra tree, LR, GNB, SGD, KNN, BAgging, LDA, MLP	98%	550	n/a
9.	[174]	2019	accelerometer + gyroscope	multidimensional Balls-into-Bins	multi-LSTM	>98.8%	117	n/a
10.	[144]	2019	gyroscope + magnetometer	sensorID	ADC Value Estimation	67bits entropy	797	n/a
11.	[175]	2020	accelerometer + gyroscope + magnetometer	sensorID	ADC Value Estimation	gyro.: 42 bits entropy, mag.: 25 bits	1006	n/a
12.	[91]	2021	gyroscope	resonance peak	decision tree, regression tree	96.5%	25	n/a

TABLE VIII
BRIEF OVERVIEW OF SOME WORKS WHICH FINGERPRINTING SMARTPHONES BASED ON OTHER TRANSDUCERS OR SOFTWARE CHARACTERISTICS

no.	work	year	signal	features	classifier	results	devices	dataset
1.	[176]	2013	ICMP timestamp	clock drifts	linear prog minimization	-	5	n/a
2.	[177]	2013	side-channel features from network traffic from several popular applications	packet size, bit size, etc.	KNN and SVM	success 90%	20	n/a
3.	[178]	2015	capacitive touchscreen	MFCC, RMS	GMM, KNN	F1 100	14	n/a
4.	[142]	2016	OS	IOS	SVM	93.67%	8000	n/a
5.	[179]	2016	sw fingerprinting, package, etc.	sw features, device properties	NB	precision>99%, recall>98%	2239	n/a
6.	[63]	2018	sw fingerprinting, package, etc.	device properties and config	thresholding	99.97%	815	n/a
7.	[180]	2018	TCP	TCP performance	KNN	75%	3	n/a
8.	[181]	2018	browser	-	-	-	-	n/a
9.	[94]	2019	magnetic signals emitted by CPU	DC/DC converter	LR, GNB, KNN, LDA, QDA, DT, SVM, ET, RF, GB	99.9%	90	n/a
10.	[182]	2020	battery power consumption	time and freq. domain features	unsupervised learning	>86%	15	n/a
11.	[183]	2020	wireless charging	clock oscillator and control scheme of the power receiver	ENS, SVM, AdaBoost, KNN, LD	96.1%	52	n/a
12.	[184]	2020	Radio Frequency	skewness, kurtosis and variance	SVM and NN	99.6%	27	yes
13.	[108]	2022	peripheral input timestamp	modular residue	FPNET, CNN	97.36%	151483+76768	[185], [186]
14.	[187]	2022	Remote GPU	normalization, Euclidean distance	KNN, CNN	< 95.8%	88	yes

sensors; 4) all sensors except accelerometers; and 5) all sensors except accelerometers and gyroscopes. For each sensor, the time and frequency domain features are extracted and five machine learning algorithms are evaluated, i.e., KNN, SVM, bagging tree, RF, and extra tree. The authors reach a maximum precision of 99.995% when all sensors are involved.

Zhang et al. [144] and [175] proposed a new method, called factory calibration fingerprinting, that is able to bypass existing protections for tracking users based on motion sensor data. They extract data from gyroscopes and magnetometers in [144] and accelerometers, gyroscopes and magnetometers in [175]. Their work involves distinct Android and iOS devices. The fingerprint is generated based on a gain matrix (squared Euclidean 2-norm function) of the data processed by computing the difference between two consecutive axes and the estimated value of the ADC.

C. Other Technologies for Device Fingerprinting

Now, we enumerate additional device fingerprinting technologies, some of which are based on other components while others are based on software (which are not part of the main scope of this work, therefore, the list is not exhaustive). In Table VIII, we compare the features, classifiers, results, and the number of devices used in the literature for smartphone identification using these different approaches.

Chen et al. [182] proposed a technique based on battery power consumption. Distinct tasks are running on the smartphones having different power consumption rates, e.g., heavy file writing and reading, computations with large numbers, broadcast transfer, etc. Time and frequency-domain features are extracted for the recorded power consumption and an unsupervised learning algorithm is applied to cluster the smartphones. The accuracy in identifying the phone was higher than 86% for 15 smartphones. Mobile devices are identified based on wireless charging fingerprints by [183]. The clock oscillator and the power receiver are used to extract the features which are then used in the SVM, AdaBoost, decision tree, KNN, and LDA classifiers. This approach reaches 97.9% accuracy for 52 devices.

Another interesting approach for device fingerprinting based on magnetic induction signals radiated by the CPU is discussed in [94]. The authors measure the CPU magnetic induction when the CPU load is at 100% as the inductor from the DC/DC converter of the CPU may produce high magnetic induction at high currents. They use for the experiments 90 devices (20 smartphones and 70 laptops) and to validate this approach 10 machine learning algorithms are used, i.e., LR,

NB, KNN, LDA, QDA, decision tree, SVM, ExtraTrees, RF, and gradient boosting. The authors report a maximum accuracy of 99.9%. The peripheral input timestamps are used in [107] for device identification. Dhakal et al. [185] and Palin et al. [186] used two public data sets, the peripherals include keyboard, mouse connected via USB and collection was done automatically on a Web based platform which evaluate the typing skill. For classification, the FPNET CNN is used and a maximum accuracy of 97.36% was achieved for 76 768 mobile devices and 151 483 desktop devices. Capacitive screen fingerprints are used in [178] for smartphone recognition. RMS and MFCC features are computed from the signature segmentation extracted from the voltage consumption. For classification, the authors use the KNN and GMM classifiers and reach an $F1$-score of 100% for 14 smartphones.

ICMP timestamp requests from which the device clock skew is extracted are proposed in [176] for smartphone fingerprinting. 10 min of collected ICMP timestamps are sufficient to distinguish between five smartphones as their oscillator skews differ in several parts-per-million (ppm). The slope of the clock skews is computed as a linear programming minimization problem. The network traffic from popular apps, e.g., Facebook, WhatsApp, Skype, Dropbox, etc., is used in [177]. Distinct features, e.g., packet size, packet ratio, number outgoing packets, byte ratio, etc., are extracted. For classification KNN and SVM are used on 14 devices with an $F1$-score of 100%. Khodzhaev et al. [180] discussed an approach based on the performance of the transmission control protocol (TCP). For classification, KNN is used and for 3 distinct devices this method reaches only 75% accuracy.

The device configuration and parameters are used for smartphone fingerprinting in [142]. The authors discuss 29 features of the Apple iOS platform, e.g., device name, language settings, installed applications, played songs, etc., and extract them from 8,000 distinct devices. The SVM classifier reaches an accuracy of 97% for this approach. In [179], 38 features are used: 1) hardware related, e.g., name, device model, manufacturer, storage capacity, etc.; 2) OS related, e.g., kernel information, Android version, etc.; and 3) user-setting related, e.g., time-zone, hour format, data format, ringtone, notification, etc. A fingerprint matching algorithm (FMA) and a fingerprint association algorithm (FAA) are used to select the relevant features and then the NB classifier is applied to cluster the devices. For 2239 devices, they reach an $F1$-score of 99.46%. Similar features are also used in [63], but here a thresholding method is used for clustering and an accuracy of 99.97% is reached for 815 devices.

In [184], a method for smartphone fingerprinting based on the radio frequency emitted by Bluetooth is discussed. The authors achieved a test accuracy between 96.9% and 99.2% using SVM and between 96.5% and 99.6% using a neural network classifier for 27 smartphones. Device identification based on remote GPU fingerprinting is proposed in [187]. The authors use 26 smartphones and 62 desktop/laptops and obtain a maximum accuracy of 95.8%. Vastel et al. [181] showed that it is possible to detect countermeasures for browser fingerprinting by using the inconsistencies that these countermeasures introduce and, besides spotting the altered fingerprints, the original fingerprint values can be also obtained.

IX. Countermeasures and Stability in Front of External Factors

In this section, we discuss countermeasures for fingerprinting and the resilience of fingerprints in front of external factors that can change them over time.

A. Countermeasures

Smartphone fingerprints can be also used by malicious apps to infringe on user's privacy. This is a very serious concern and we cannot end our survey without mentioning it along with some countermeasures. Briefly, to combat these attacks, several countermeasures can be implemented: 1) adding noise to the sampled data (which is also commonly referred to as obfuscation); 2) calibrating the sensors so that differences become negligible; 3) restricting the access to sensors' data; or 4) lowering the sampling fidelity. These approaches can be also combined. We discuss them in what follows.

Adding Noise (Obfuscation): A simple method to modify the smartphone fingerprints is to add noise. This approach does not affect the smartphone functionally [4] and it is not expensive in computations and power consumption. The addition of noise has been also discussed in [93] within scope of microphone identification. This work considers various types of sounds e.g., traffic, train, barrier, etc., and reports that the accuracy drops below 50% at a SNR below a specific threshold, e.g., –40 db for car horn, –20 db for car tiers, so that microphone identification no longer works. Also, Baldini and Amerini [83] analyzed the influence of additive white Gaussian noise (AWGN) at distinct SNR levels and the accuracy drops below 50% at a SNR of 0–5 db. The work in [45] also shows that in the case of loudspeaker identification, the volume can influence the fingerprints.

Sensor Calibration: Calibration is generally used to increase the precision of measurements performed by various sensors, but it was also proposed as a countermeasure against sensor fingerprinting. More commonly, it is proposed for accelerometers and gyroscopes. For example, the calibration of accelerometers and gyroscopes is discussed in [15] as a countermeasures against sensor fingerprinting. Notably, some works have managed to fingerprint accelerometers and gyroscopes even if factory calibrations were performed [144], [145], [175]. To prevent this and make fingerprinting infeasible, the last two of these works propose that one can round the factory calibrated sensor output to the nearest multiple of the nominal gain [144], [175].

Restricted Access to Device Peripherals and Data: Implementing policies that control the access rights of other applications on sensor data is another countermeasure proposed in [3] and also discussed in [4]. It may be also worth recalling here that malicious apps with access to the microphone can allow the interception of the phone's PIN code [188]. This proves how serious are the implications of giving access to such peripherals. Notably, smartphones also

leverage the use of various IoT devices that surround our home, exposing even more data about owners. Having this in mind, the work in [189] discusses a mobile-cloud framework with fine-grained permission authorization for IoT. A privacy risk assessment for mobile applications, which considers permissions and information flow leakage, is presented in [190].

Lowering Sampling Fidelity: Lowering the sampling rate can also be a countermeasure and it may also increases the battery life (especially in the case of data collected from motion sensors). Data filtering and reducing the sampling rate can hide part of features such that the fingerprinting process will no longer be possible. The Android platform is already considering risks related to fingerprinting by sensor sampling and started to limit the access for applications since Android 12 (API level 31). For a sample rate higher than 200 Hz (or about 50 Hz for direct, raw sensor data), apps need to be granted a new permission called `HIGH_SAMPLING_RATE_SENSORS`. Note that this is declared as a `normal` level permission and therefore granted automatically, but can be used for determining apps that potentially access higher sample rates [191]. As a further mitigation, motion sensors (including accelerometer) are always rate limited even for apps holding this permission if the microphone has been turned off by the user. Finally, Android 10 introduced an UI element in the form of the *Sensors Off* quick tile that can be used to disable app access to all sensors, including microphone, camera and motion sensors (with the exception of phone calls still using the microphone). However, this UI element needs to be enabled through developer options and is therefore not targeting end-users at the time of this writing [192]. On Apple iOS, apps seem to be able to use `Core Motion` to request sample rates as far as the hardware supports it [193]. Apple recommends as best practice to avoid using accelerometers or gyroscopes outside of active gameplay [194]. To the best of our knowledge, there seem to be no automatic limitations at the time of this writing.

It is also true that these countermeasures are not always applicable, or it is highly inconvenient to use them. For example, sometimes sampling restrictions cannot be applied, as in the case of gaming applications that require the maximum sampling rate from accelerometers for better accuracy. Reducing the sampling of accelerometers also has impact on physical activity monitoring apps [195]. Regarding camera sensors, photograph editing software may require access to the raw image data (that may contain even more phone-related artifacts) for optimal performance. As expected, all countermeasures come at a price.

B. Stability in Front of External Factors

Many of the existing works have also considered resilience in front of external factors. In the case of camera sensors, various factors have been considered like temperature or voltage variations [20], [56], [57], [60], [61], [64], [148]. Post processing of the images by third parties has been considered, as well as changes in the brightness level [22], [36], [74], [112]. For audio recordings, in the case of microphone and loudspeaker fingerprinting, various kinds of environmental factors were considered, such as ambient or environment specific noise [46], [69], [78], [79], [88], [93], [106], [116] [9], [45], [53], [54], [55], AWGN [45], [83], [84], distance from the speaker [9], [53], sampling rate, or even changes in the volume or orientation of the speaker [9], [45], [53]. In the case of accelerometers, temperature has been generally considered [15], [144], [174], [175], while some works also mention humidity [174]. These works have also considered the influence of the same factors on gyroscopes.

One important factor that seems to be omitted by most works is the stability of the samples over time. To the best of our knowledge, only the excellent work from [15] evaluates the stability of the samples by collecting data at one month distance. Concretely, accelerometer and gyroscope data is collected at an interval of 37 days and the F-score, which was 100% for data collected during the same day, drops between 88% and 92% for different days. Further evaluations may be needed to asses if samples are stable in the long run. Zhang et al. [144] also relied on the sensor factory calibration file, which is stored in the nonvolatile memory and should not change over time. Other works assess the stability of hardware fingerprints in the case of different electronic components. For example, the magnetic signals from the CPU are used in [94] and the authors prove that they do not change over the course of two days and in distinct locations. Fingerprinting the GPU from JavaScript collected data is proposed in [187] and the fingerprints are shown to be stable during 24 days of experimentation. The stability of clock-based fingerprinting is also discussed in [196] where measurements are performed two months apart.

X. Conclusion and Future Directions

There is a very largenumber of works that address smartphone identification based on the physical fingerprints of their embedded transducers, mainly cameras, microphones, loudspeakers, and accelerometers. The most consistent body of works which we surveyed was concerned with camera fingerprints. This is somewhat natural as users nowadays commonly upload photographs on various websites, making them very easy to collect. Also, a lot of samples and features can be extracted from images and there are several public data sets dedicated for research works. A lesser number of works used microphones and there are only a few works which are using loudspeakers. Device fingerprinting based on audio signals, from microphones and loudspeakers, may have attracted less research because, although this kind of data is easy to analyze, it may be more difficult to collect. For microphones, there are several public data sets (the majority of them are targeting speech recognition and crime related investigations) which were also used for device identification based on their microphones while for loudspeaker identification a single public data set is available. In the case of accelerometers, the number of works strictly dedicated to fingerprinting is also somewhat limited, despite the fact that accelerometers were so commonly employed for device-to-device authentication. There are also only isolated attempts in using gyroscopes and magnetometers for fingerprinting. Regarding accuracy, it

seems that camera sensors provide the best fingerprint, many of the works from Table III in our survey reporting an accuracy close to 100%. This happens because CMOS sensors collect high amounts of information due to the over-increasing resolution of modern cameras. Next to camera sensors, microphones and loudspeakers may be a reliable source, with a reported accuracy generally between 90%–100% according to Tables IV and V from our survey. Accelerometers seem to have a lower accuracy for fingerprinting, which according to Table VI in our survey is between 58.7% and 95%.

As future research directions, there are several gaps that need to be covered. As outlined previously, there is only a very limited number of works that have addressed sample stability over time and this happened only over a small period of one month [15]. The use of multiple sensors can be also considered for improving the reliability of the fingerprinting process over time, since various sensors may be unevenly affected by wear and tear. Running the experiments over extended time periods and using a larger number of devices in the field may be considered by OEMs or large app developers with a considerable install base (but it is generally out of reach for nonprofit academic research). Last but not least, incremental learning, a well-known method of machine learning which requires to continuously update the existing model as new data becomes available, may be one way to address this problem by ensuring an up-to-date trained model for the device. Also, almost all of the existing works have dealt with closed-world models in which only devices coming from a limited set are to be identified. There are only a few works [12], [197] which address open-world scenarios, that are more relevant for practice since the methodology is also tested against devices that were not part of the training data set. Related to this, the use of one-class classification, which requires a single device in the training data set and later separates it from the rest in the testing data set, is of significant interest. Most of the papers so far tried to separate between multiple devices that were already learned, while only a few works explicitly used one-class classifiers [12], [77], [79], [133]. The selection of specific inputs that give a more accurate classification for the transducers is also one possible area of investigation. It is well known that certain inputs can yield a better response in the case of PUFs, e.g., the RowHammer PUF [198]. As previously stated, in the case of CMOS sensors, dark images seem to give a better response [141], while for loudspeakers, a sweep signal offers a more complete characterization [45]. Other works have considered those inputs which are more realistic for practice, such as human speech in the case of microphones, or music in the case of loudspeakers. Finding specific inputs for which the transducer gives the most specific response is one possible area for future investigations.

There is also a significant number of works that use other technologies instead of transducers, such as software fingerprinting, ICMP timestamp, OS, TCP, battery consumption, wireless charging, capacitive touchscreens, CPU magnetic field, and the input from various peripherals. These works were only briefly accounted here and do not form the main target of our survey. We may consider an in-depth analysis of them as future work.

ACKNOWLEDGMENT

The authors would like to express their gratitude to the reviewers for their comments that helped them to improve this work.

REFERENCES

[1] GSMA Association. "2021 mobile industry impact report: Sustainable development goals." 2021. [Online]. Available: https://www.gsma.com/betterfuture/2021sdgimpactreport/wp-content/uploads/2021/09/GSMA-SDGreport-singles.pdf

[2] Statista. "Forecast number of mobile devices worldwide from 2020 to 2025 (in billions)." 2022. Accessed: Nov. 1, 2022. [Online]. Available: https://www.statista.com/statistics/245501/multiple-mobile-device-ownership-worldwide/

[3] V. K. Khanna, "Remote fingerprinting of mobile phones," *IEEE Wireless Commun.*, vol. 22, no. 6, pp. 106–113, Dec. 2015.

[4] G. Baldini and G. Steri, "A survey of techniques for the identification of mobile phones using the physical fingerprints of the built-in components," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 3, pp. 1761–1789, 3rd Quart., 2017.

[5] N. Kanjan, K. Patil, S. Ranaware, and P. Sarokte, "A comparative study of fingerprint matching algorithms," *Int. Res. J. Eng. Technol.*, vol. 4, no. 11, pp. 1892–1896, 2017.

[6] K. Ren, Z. Qin, and Z. Ba, "Toward hardware-rooted smartphone authentication," *IEEE Wireless Commun.*, vol. 26, no. 1, pp. 114–119, Feb. 2019.

[7] S. J. Alsunaidi and A. M. Almuhaideb, "Investigation of the optimal method for generating and verifying the smartphone's fingerprint: A review," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 34, no. 5, pp. 1919–1932, 2022.

[8] P. M. S. Sánchez, J. M. J. Valero, A. H. Celdrán, G. Bovet, M. G. Pérez, and G. M. Pérez, "A survey on device behavior fingerprinting: Data sources, techniques, application scenarios, and datasets," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 2, pp. 1048–1077, 2nd Quart., 2021.

[9] A. Das, N. Borisov, and M. Caesar, "Fingerprinting smart devices through embedded acoustic components," 2014, *arXiv:1403.3366*.

[10] D. Litwiller, "CCD vs. CMOS," *Photon Spectra*, vol. 35, no. 1, pp. 154–158, 2001.

[11] S. Dey, N. Roy, W. Xu, R. R. Choudhury, and S. Nelakuditi, "AccelPrint: Imperfections of accelerometers make smartphones trackable," in *Proc. NDSS*, 2014, pp. 1–8.

[12] Z. Ding and M. Ming, "Accelerometer-based mobile device identification system for the realistic environment," *IEEE Access*, vol. 7, pp. 131435–131447, 2019.

[13] A. Das, N. Borisov, and E. Chou, "Every move you make: Exploring practical issues in smartphone motion sensor fingerprinting and countermeasures," *Proc. Privacy Enhanc. Technol.*, no. 1, pp. 88–108, 2018.

[14] A. Das, N. Borisov, and M. Caesar, "Exploring ways to mitigate sensor-based smartphone fingerprinting," 2015, *arXiv:1503.01874*.

[15] A. Das, N. Borisov, and M. Caesar, "Tracking mobile Web users through motion sensors: Attacks and defenses," in *Proc. NDSS*, 2016, pp. 1–5.

[16] T. Hupperich, H. Hosseini, and T. Holz, "Leveraging sensor fingerprinting for mobile device authentication," in *Proc. Int. Conf. Detect. Intrusions Malware Vulnerability Assess.*, 2016, pp. 377–396.

[17] I. Amerini, R. Becarelli, R. Caldelli, A. Melani, and M. Niccolai, "Smartphone fingerprinting combining features of on-board sensors," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 10, pp. 2457–2466, Oct. 2017.

[18] G. Baldini, G. Steri, I. Amerini, and R. Caldelli, "The identification of mobile phones through the fingerprints of their built-in magnetometer: An analysis of the portability of the fingerprints," in *Proc. Int. Carnahan Conf. Security Technol.*, 2017, pp. 1–6.

[19] T. Van Goethem, W. Scheepers, D. Preuveneers, and W. Joosen, "Accelerometer-based device fingerprinting for multi-factor mobile authentication," in *Engineering Secure Software and Systems*. London, U.K.: Springer, 2016, pp. 106–121.

[20] Y. Kim and Y. Lee, "CamPUF: Physically unclonable function based on CMOS image sensor fixed pattern noise," in *Proc. 55th Annu. Design Autom. Conf.*, 2018, pp. 1–6.

[21] M. Chen, J. Fridrich, M. Goljan, and J. Lukás, "Determining image origin and integrity using sensor noise," *IEEE Trans. Inf. Forensics Security*, vol. 3, no. 1, pp. 74–90, Mar. 2008.

[22] F. Marra, G. Poggi, C. Sansone, and L. Verdoliva, "Blind PRNU-based image clustering for source identification," *IEEE Trans Inf. Forensics Security*, vol. 12, no. 9, pp. 2197–2211, Sep. 2017.

[23] D. Valsesia, G. Coluccia, T. Bianchi, and E. Magli, "User authentication via PRNU-based physical unclonable functions," *IEEE Trans Inf. Forensics Security*, vol. 12, no. 8, pp. 1941–1956, Aug. 2017.

[24] R. Deka, C. Galdi, and J.-L. Dugelay, "Hybrid G-PRNU: Optimal parameter selection for scale-invariant asymmetric source smartphone identification," *Electron. Imag.*, vol. 31, no. 5, pp. 546–551, 2019.

[25] V. Bruni, A. Salvi, and D. Vitulano, "Joint correlation measurements for PRNU-based source identification," in *Proc. Int. Conf. Comput. Anal. Images Patterns*, 2019, pp. 245–256.

[26] M. S. Behare, A. Bhalchandra, and R. Kumar, "Source camera identification using photo response noise uniformity," in *Proc. IEEE 3rd Int. Conf. Electron. Commun. Aerosp. Technol.*, 2019, pp. 731–734.

[27] M. Iuliani, M. Fontani, and A. Piva, "A leak in PRNU based source identification—Questioning fingerprint uniqueness," *IEEE Access*, vol. 9, pp. 52455–52463, 2021.

[28] A. Lawgaly and F. Khelifi, "Sensor pattern noise estimation based on improved locally adaptive DCT filtering and weighted averaging for source camera identification and verification," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 2, pp. 392–404, Feb. 2017.

[29] D. Valsesia, G. Coluccia, T. Bianchi, and E. Magli, "Large-scale image retrieval based on compressed camera identification," *IEEE Trans. Multimedia*, vol. 17, no. 9, pp. 1439–1449, Sep. 2015.

[30] J. R. Corripio, D. A. González, A. S. Orozco, L. G. Villalba, J. Hernandez-Castro, and S. J. Gibson, "Source smartphone identification using sensor pattern noise and wavelet transform," in *Proc. ICDP*, 2013, pp. 1–6.

[31] M. Tiwari and B. Gupta, "Efficient prnu extraction using joint edge-preserving filtering for source camera identification and verification," in *Proc. IEEE Appl. Signal Process. Conf.*, 2018, pp. 14–18.

[32] L. Debiasi, E. Leitet, K. Norell, T. Tachos, and A. Uhl, "Blind source camera clustering of criminal case data," in *Proc. IEEE 7th Int. Workshop Biometrics Forensics (IWBF)*, 2019, pp. 1–6.

[33] D. Cozzolino, F. Marra, D. Gragnaniello, G. Poggi, and L. Verdoliva, "Combining PRNU and noiseprint for robust and efficient device source identification," *EURASIP J. Inf. Security*, no. 1, pp. 1–12, 2020.

[34] S. Mandelli, D. Cozzolino, P. Bestagini, L. Verdoliva, and S. Tubaro, "CNN-based fast source device identification," *IEEE Signal Process. Lett.*, vol. 27, pp. 1285–1289, 2020.

[35] C.-T. Li et al., "Beyond PRNU: Learning robust device-specific fingerprint for source camera identification," 2021, *arXiv:2111.02144*.

[36] B. N. Sarkar, S. Barman, and R. Naskar, "Blind source camera identification of online social network images using adaptive thresholding technique," in *Proc. Conf. Front. Comput. Syst.*, 2021, pp. 637–648.

[37] R. Rouhi, F. Bertini, and D. Montesi, "No matter what images you share, you can probably be fingerprinted anyway," *J. Imag.*, vol. 7, no. 2, p. 33, 2021.

[38] J. Bernacki, "A survey on digital camera identification methods," *Forensic Sci. Int. Digit. Invest.*, vol. 34, Sep. 2020, Art. no. 300983.

[39] *Information Technology—Digital Compression and Coding of Continuous—Tone Still Images: Requirements and Guidelines*, ISO/IEC Standard 10918-1, 1994.

[40] A. Roy, R. S. Chakraborty, V. U. Sameer, and R. Naskar, "Camera source identification using discrete cosine transform residue features and ensemble classifier," in *Proc. CVPR Workshops*, 2017, pp. 1848–1854.

[41] B. Gupta and M. Tiwari, "Improving performance of source-camera identification by suppressing peaks and eliminating low-frequency defects of reference SPN," *IEEE Signal Process. Lett.*, vol. 25, no. 9, pp. 1340–1343, Sep. 2018.

[42] A. El-Yamany, H. Fouad, and Y. Raffat, "A generic approach CNN-based camera identification for manipulated images," in *Proc. IEEE Int. Conf. Electron. Inf. Technol.*, 2018, pp. 0165–0169.

[43] G. Xu and Y. Q. Shi, "Camera model identification using local binary patterns," in *Proc. IEE Int. Conf. Multimedia Expo*, 2012, pp. 392–397.

[44] N. Zandi and F. Razzazi, "Source camera identification using WLBP descriptor," in *Proc. Int. Conf. Mach. Vis. Image Process. (MVIP)*, 2020, pp. 1–6.

[45] A. Berdich, B. Groza, R. Mayrhofer, E. Levy, A. Shabtai, and Y. Elovici, "Sweep-to-unlock: Fingerprinting smartphones based on loudspeaker roll-off characteristics," *IEEE Trans. Mobile Comput.*, vol. 22, no. 4, pp. 2417–2434, Apr. 2023.

[46] Ö. Eskidere, "Source microphone identification from speech recordings based on a Gaussian mixture model," *Turkish J. Electron. Eng. Comput. Sci.*, vol. 22, no. 3, pp. 754–767, 2014.

[47] R. Aggarwal, S. Singh, A. K. Roul, and N. Khanna, "Cellphone identification using noise estimates from recorded audio," in *Proc. IEEE Int. Conf. Commun. Signal Process.*, 2014, pp. 1218–1222.

[48] C. Hanilçi and T. Kinnunen, "Source cell-phone recognition from recorded speech using non-speech segments," *Digit. Signal Process.*, vol. 35, pp. 75–85, Dec. 2014.

[49] O. Eskidere and A. Karatutlu, "Source microphone identification using multitaper MFCC features," in *Proc. IEEE Int. Conf. Electron. Elect. Eng.*, 2015, pp. 227–231.

[50] Y. Li, X. Zhang, X. Li, Y. Zhang, J. Yang, and Q. He, "Mobile phone clustering from speech recordings using deep representation and spectral clustering," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 4, pp. 965–977, Apr. 2018.

[51] X. Li, D. Yan, L. Dong, and R. Wang, "Anti-forensics of audio source identification using generative adversarial network," *IEEE Access*, vol. 7, pp. 184332–184339, 2019.

[52] V. A. Hadoltikar, V. R. Ratnaparkhe, and R. Kumar, "Optimization of MFCC parameters for mobile phone recognition from audio recordings," in *Proc. Int. Conf. Electron. Commun. Aero Technol.*, 2019, pp. 777–780.

[53] A. Das, N. Borisov, and M. Caesar, "Do you hear what i hear? Fingerprinting smart devices through embedded acoustic components," in *Proc. ACM Conf Comput. Commun. Security*, 2014, pp. 441–452.

[54] T. Qin, R. Wang, D. Yan, and L. Lin, "Source cell-phone identification in the presence of additive noise from CQT domain," *Information*, vol. 9, no. 8, p. 205, 2018.

[55] Z. Zhou, W. Diao, X. Liu, and K. Zhang, "Acoustic fingerprinting revisited: Generate stable device ID stealthily with inaudible sound," in *Proc. ACM Conf. Comput. Commun. Security*, 2014, pp. 429–440.

[56] Y. Cao, L. Zhang, S. S. Zalivaka, C.-H. Chang, and S. Chen, "CMOS image sensor based physical unclonable function for smart phone security applications," in *Proc. IEEE Symp. Integr. Circuits*, 2014, pp. 392–395.

[57] Y. Cao, L. Zhang, S. S. Zalivaka, C.-H. Chang, and S. Chen, "CMOS image sensor based physical unclonable function for coherent sensor-level authentication," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 62, no. 11, pp. 2629–2640, Nov. 2015.

[58] A. Maiti, V. Gunreddy, and P. Schaumont, "A systematic method to evaluate and compare the performance of physical unclonable functions," in *Embedded Systems Design With FPGAs*. Cham, Switzerland: Springer, 2013, pp. 245–267.

[59] R. Arjona, M. A. Prada-Delgado, J. Arcenegui, and I. Baturone, "Using physical unclonable functions for Internet-of-Things security cameras," in *Interoperability, Safety and Security in IoT*. Valencia, Spain: Springer, 2017, pp. 144–153.

[60] X. Lu, L. Hong, and K. Sengupta, "CMOS optical PUFs using noise-immune process-sensitive photonic crystals incorporating passive variations for robustness," *IEEE J. Solid-State Circuits*, vol. 53, no. 9, pp. 2709–2721, Sep. 2018.

[61] Y. Zheng, X. Zhao, T. Sato, Y. Cao, and C.-H. Chang, "Ed-PUF: Event-driven physical unclonable function for camera authentication in reactive monitoring system," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 2824–2839, 2020.

[62] A. T. Erozan, M. Hefenbrock, M. Beigl, J. Aghassi-Hagmann, and M. B. Tahoori, "Image PUF: A physical unclonable function for printed electronics based on optical variation of printed inks," in *Proc. Cryptol. ePrint Arch.*, 2019, pp. 1–9.

[63] Z. Ding, W. Zhou, and Z. Zhou, "Configuration-based fingerprinting of mobile device using incremental clustering," *IEEE Access*, vol. 6, pp. 72402–72414, 2018.

[64] J. Lukas, J. Fridrich, and M. Goljan, "Digital camera identification from sensor pattern noise," *IEEE Trans. Inf. Forensics Security*, vol. 1, no. 2, pp. 205–214, Jun. 2006.

[65] T. Gloe, M. Kirchner, A. Winkler, and R. Böhme, "Can we trust digital image forensics?" in *Proc. 15th ACM Int. Conf. Multimedia*, 2007, pp. 78–86.

[66] Q.-T. Phan, G. Boato, and F. G. De Natale, "Image clustering by source camera via sparse representation," in *Proc. 2nd Int. Workshop Multimedia Forensics Security*, 2017, pp. 1–5.

[67] S. Taspinar, M. Mohanty, and N. Memon, "Camera fingerprint extraction via spatial domain averaged frames," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 3270–3282, 2020.

[68] J. Bernacki, "On robustness of camera identification algorithms," *Multimedia Tools Appl.*, vol. 80, no. 1, pp. 921–942, 2021.

[69] S. Ikram and H. Malik, "Microphone identification using higher-order statistics," in *Proc. 46th Int. Conf. Audio Forensics*, 2012, pp. 1–8.

[70] F. Kurniawan, M. S. M. Rahim, M. S. Khalil, and M. K. Khan, "Statistical-based audio forensic on identical microphones," *Int. J. Electron. Comput. Eng.*, vol. 6, no. 5, p. 2211, 2016.

[71] K. S. Choi, E. Y. Lam, and K. K. Wong, "Source camera identification using footprints from lens aberration," in *Proc. Digit. Photography II*, vol. 6069, 2006, Art. no. 60690J.

[72] B. Xu, X. Wang, X. Zhou, J. Xi, and S. Wang, "Source camera identification from image texture features," *Neurocomputing*, vol. 207, pp. 131–140, Sep. 2016.

[73] V. U. Sameer, A. Sarkar, and R. Naskar, "Source camera identification model: Classifier learning, role of learning curves and their interpretation," in *Proc. Int. Conf. Wireless Commun. Signal Process. Netw.*, 2017, pp. 2660–2666.

[74] A. Rashidi and F. Razzazi, "Single image camera identification using i-vectors," in *Proc. IEEE Int. Conf. Comput. Knowl. Eng.*, 2017, pp. 406–410.

[75] Y. Huang, L. Cao, J. Zhang, L. Pan, and Y. Liu, "Exploring feature coupling and model coupling for image source identification," *IEEE Trans Inf. Forensics Security*, vol. 13, no. 12, pp. 3108–3121, Dec. 2018.

[76] M. H. A. Banna, M. A. Haider, M. J. A. Nahian, M. M. Islam, K. A. Taher, and M. S. Kaiser, "Camera model identification using deep CNN and transfer learning approach," in *Proc. IEEE Int. Conf Robot. Electron. Signal Process. Technol.*, 2019, pp. 626–630.

[77] P. R. M. Júnior, L. Bondi, P. Bestagini, S. Tubaro, and A. Rocha, "An in-depth study on open-set camera model identification," *IEEE Access*, vol. 7, pp. 180713–180726, 2019.

[78] R. Buchholz, C. Kraetzer, and J. Dittmann, "Microphone classification using fourier coefficients," in *Proc. Int. Workshop Inf. Hiding*, 2009, pp. 235–246.

[79] H. Q. Vu, S. Liu, X. Yang, Z. Li, and Y. Ren, "Identifying microphone from noisy recordings by using representative instance one class-classification approach," *J. Netw.*, vol. 7, no. 6, pp. 908–917, 2012.

[80] C. Kotropoulos and S. Samaras, "Mobile phone identification using recorded speech signals," in *Proc. IEEE 19th Int. Conf. Digit. Signal Process.*, 2014, pp. 586–591.

[81] J. Zeng et al., "Audio recorder forensic identification in 21 audio recorders," in *Proc. IEEE Int. Conf. Progr. Inf. Comput.*, 2015, pp. 153–157.

[82] L. Zou, Q. He, and X. Feng, "Cell phone verification from speech recordings using sparse representation," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2015, pp. 1787–1791.

[83] G. Baldini and I. Amerini, "Smartphones identification through the built-in microphones with convolutional neural network," *IEEE Access*, vol. 7, pp. 158685–158696, 2019.

[84] G. Baldini, I. Amerini, and C. Gentile, "Microphone identification using convolutional neural networks," *IEEE Sensors Lett.*, vol. 3, no. 7, pp. 1–4, Jul. 2019.

[85] Y. Jiang and F. H. F. Leung, "Source microphone recognition aided by a kernel-based projection method," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 11, pp. 2875–2886, Nov. 2019.

[86] C. Jin, R. Wang, and D. Yan, "Source smartphone identification by exploiting encoding characteristics of recorded speech," *Digit. Invest.*, vol. 29, pp. 129–146, Jun. 2019.

[87] G. Baldini and I. Amerini, "An evaluation of entropy measures for microphone identification," *Entropy*, vol. 22, no. 11, p. 1235, 2020.

[88] D. Bykhovsky, "Recording device identification by ENF harmonics power analysis," *Forensic Sci. Int.*, vol. 307, Feb. 2020, Art. no. 110100.

[89] X. Zhou, X. Zhuang, H. Tang, M. Hasegawa-Johnson, and T. S. Huang, "Novel Gaussianized vector representation for improved natural scene categorization," *Pattern Recongit. Lett.*, vol. 31, no. 8, pp. 702–708, 2010.

[90] D. P. Chowdhury, S. Bakshi, P. K. Sa, and B. Majhi, "Wavelet energy feature based source camera identification for ear biometric images," *Pattern Recognit. Lett.*, vol. 130, pp. 139–147, Feb. 2020.

[91] J. Tian et al., "Mobile device fingerprint identification using gyroscope resonance," *IEEE Access*, vol. 9, pp. 160855–160867, 2021.

[92] I. Amerini, P. Bestagini, L. Bondi, R. Caldelli, M. Casini, and S. Tubaro, "Robust smartphone fingerprint by mixing device sensors features for mobile strong authentication," *Electron. Imag.*, no. 8, pp. 1–8, 2016.

[93] A. Berdich, B. Groza, E. Levy, A. Shabtai, Y. Elovici, and R. Mayrhofer, "Fingerprinting smartphones based on microphone characteristics from environment affected recordings," *IEEE Access*, vol. 10, pp. 122399–122413, 2022.

[94] Y. Cheng, X. Ji, J. Zhang, W. Xu, and Y.-C. Chen, "DeMiCPU: Device fingerprinting with magnetic signals radiated by CPU," in *Proc. ACM SIGSAC Conf. Comput. Commun. Security*, 2019, pp. 1149–1170.

[95] A. Tuama, F. Comby, and M. Chaumont, "Camera model identification with the use of deep convolutional neural networks," in *Proc. IEEE Int. Workshop Inf. Forensics Security*, 2016, pp. 1–6.

[96] H. Yao, T. Qiao, M. Xu, and N. Zheng, "Robust multi-classifier for camera model identification based on convolution neural network," *IEEE Access*, vol. 6, pp. 24973–24982, 2018.

[97] D. Freire-Obregón, F. Narducci, S. Barra, and M. Castrillón-Santana, "Deep learning for source camera identification on mobile devices," *Pattern Recognit. Lett.*, vol. 126, pp. 86–91, Sep. 2019.

[98] P. Yang, R. Ni, Y. Zhao, and W. Zhao, "Source camera identification based on content-adaptive fusion residual networks," *Pattern Recognit. Lett.*, vol. 119, pp. 195–204, Mar. 2019.

[99] D. Cozzolino and L. Verdoliva, "Noiseprint: A CNN-based camera model fingerprint," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 144–159, 2019.

[100] X. Ding, Y. Chen, Z. Tang, and Y. Huang, "Camera identification based on domain knowledge-driven deep multi-task learning," *IEEE Access*, vol. 7, pp. 25878–25890, 2019.

[101] M. Zhao, B. Wang, F. Wei, M. Zhu, and X. Sui, "Source camera identification based on coupling coding and adaptive filter," *IEEE Access*, vol. 8, pp. 54431–54440, 2019.

[102] A. M. Rafi, T. I. Tonmoy, U. Kamal, Q. J. Wu, and M. K. Hasan, "RemNet: Remnant convolutional neural network for camera model identification," *Neural Comput. Appl.*, vol. 33, no. 8, pp. 3655–3670, 2021.

[103] D. D. Cortivo, S. Mandelli, P. Bestagini, and S. Tubaro, "CNN-based multi-modal camera model identification on video sequences," *J. Imag.*, vol. 7, no. 8, p. 135, 2021.

[104] V. Verma and N. Khanna, "CNN-based system for speaker independent cell-phone identification from recorded audio," in *Proc. CVPR Workshops*, 2019, pp. 53–61.

[105] X. Lin, J. Zhu, and D. Chen, "Subband aware CNN for cell-phone recognition," *IEEE Signal Process. Lett.*, vol. 27, pp. 605–609, 2020.

[106] M. A. Qamhan, H. Altaheri, A. H. Meftah, G. Muhammad, and Y. A. Alotaibi, "Digital audio forensics: Microphone and environment classification using deep learning," *IEEE Access*, vol. 9, pp. 62719–62733, 2021.

[107] J. Monaco, "Device fingerprinting with peripheral timestamps," in *Proc. IEEE Symp. Security Privacy*, Los Alamitos, CA, USA, May 2022, pp. 243–258.

[108] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 25, 2012, pp. 1097–1105.

[109] C. Szegedy et al., "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern. Rec.*, 2015, pp. 1–9.

[110] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Rec.*, 2016, pp. 770–778.

[111] Y. Liu, Z. Zou, Y. Yang, N.-F. B. Law, and A. A. Bharath, "Efficient source camera identification with diversity-enhanced patch selection and deep residual prediction," *Sensors*, vol. 21, no. 14, p. 4701, 2021.

[112] V. U. Sameer, I. Dali, and R. Naskar, "A deep learning based digital forensic solution to blind source identification of Facebook images," in *Proc. Int. Conf. Inf. Syst. Securty*, 2018, pp. 291–303.

[113] R. Rouhi, F. Bertini, D. Montesi, X. Lin, Y. Quan, and C.-T. Li, "Hybrid clustering of shared images on social networks for digital forensics," *IEEE Access*, vol. 7, pp. 87288–87302, 2019.

[114] Q.-T. Phan, G. Boato, and F. G. B. De Natale, "Accurate and scalable image clustering based on sparse representation of camera fingerprint," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 7, pp. 1902–1916, Jul. 2019.

[115] D. Chen et al., "S2M: A lightweight acoustic fingerprints-based wireless device authentication protocol," *IEEE Internet Things J.*, vol. 4, no. 1, pp. 88–100, Feb. 2017.

[116] Y. Lee, J. Li, and Y. Kim, "MicPrint: Acoustic sensor fingerprinting for spoof-resistant mobile device authentication," in *Proc. 16th EAI Int. Conf. Mobile Ubiquitous Syst. Comput. Netw. Services*, 2019, pp. 248–257.

[117] F. Lorenz et al., "Fingerprinting analog IoT sensors for secret-free authentication," in *Proc. IEEE 29th Int. Conf. Comput. Commun. Netw.*, 2020, pp. 1–6.

[118] M. T. Ahvanooey et al., "Modern authentication schemes in smartphones and IoT devices: An empirical survey," *IEEE Internet Things J.*, vol. 9, no. 10, pp. 7639–7663, May 2022.

[119] A. Anistoroaei, A. Berdich, P. Iosif, and B. Groza, "Secure audio–visual data exchange for Android in-vehicle ecosystems," *Appl. Sci.*, vol. 11, no. 19, p. 9276, 2021.

[120] B. Groza, A. Berdich, C. Jichici, and R. Mayrhofer, "Secure accelerometer-based pairing of mobile devices in multi-modal transport," *IEEE Access*, vol. 8, pp. 9246–9259, 2020.

[121] S. Wang, C. Chen, and J. Ma, "Accelerometer based transportation mode recognition on mobile phones," in *Proc. Asia–Pac. Conf. Wearable Comput. Syst. (APWCS)*, 2010, pp. 44–46.

[122] S. Reddy, M. Mun, J. Burke, D. Estrin, M. Hansen, and M. Srivastava, "Using mobile phones to determine transportation modes," *ACM Trans. Sensor Netw.*, vol. 6, no. 2, p. 13, 2010.

[123] T. Feng and H. J. Timmermans, "Transportation mode recognition using GPS and accelerometer data," *Transp. Res. C Emerg. Technol.*, vol. 37, pp. 118–130, Dec. 2013.

[124] X. Su, H. Tong, and P. Ji, "Activity recognition with smartphone sensors," *Tsinghua Sci. Technol.*, vol. 19, no. 3, pp. 235–249, Jun. 2014.

[125] D. A. Johnson and M. M. Trivedi, "Driving style recognition using a smartphone as a sensor platform," in *Proc. IEEE 14th Int. IEEE Conf Intell. Transp. Syst.*, 2011, pp. 1609–1615.

[126] M. Van Ly, S. Martin, and M. M. Trivedi, "Driver classification and driving style recognition using inertial sensors," in *Proc. IEEE Intell. Veh. Symp. (IV)*, 2013, pp. 1040–1045.

[127] N. Kalra and D. Bansal, "Analyzing driver behavior using smartphone sensors: A survey," *Int. J. Electron. Elect. Eng.*, vol. 7, no. 7, pp. 697–702, 2014.

[128] P. Singh, N. Juneja, and S. Kapoor, "Using mobile phone sensors to detect driving behavior," in *Proc. 3rd ACM Symp. Comput. Develop.*, 2013, p. 53.

[129] J. Yu, Z. Chen, Y. Zhu, Y. J. Chen, L. Kong, and M. Li, "Fine-grained abnormal driving behaviors detection and identification with smartphones," *IEEE Trans. Mobile Comput.*, vol. 16, no. 8, pp. 2198–2212, Aug. 2017.

[130] K. Chen, M. Lu, X. Fan, M. Wei, and J. Wu, "Road condition monitoring using on-board three-axis accelerometer and GPS sensor," in *Proc. CHINACOM*, 2011, pp. 1032–1037.

[131] A. Mednis, G. Strazdins, R. Zviedris, G. Kanonirs, and L. Selavo, "Real time pothole detection using android smartphones with accelerometers," in *Proc. IEEE DCOSS*, 2011, pp. 1–6.

[132] M. Muaaz and R. Mayrhofer, "Accelerometer based gait recognition using adapted Gaussian mixture models," in *Proc. 14th Int. Conf Adv. Mobile Comput. Multimedia*, 2016, pp. 288–291.

[133] G. Hu, Z. He, and R. Lee, "Smartphone impostor detection with built-in sensors and deep learning," 2020, *arXiv:2002.03914*.

[134] J. Hua, Z. Shen, and S. Zhong, "We can track you if you take the metro: Tracking metro riders using accelerometers on smartphones," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 2, pp. 286–297, Feb. 2017.

[135] A. Mongia, V. M. Gunturi, and V. Naik, "Detecting activities at metro stations using smartphone sensors," in *Proc. 10th Int. Conf. Commun. Syst. Netw.*, 2018, pp. 57–65.

[136] A. Pandya and R. K. Shukla, "New perspective of nanotechnology: Role in preventive forensic," *Egypt. J. Forensics Sci.*, vol. 8, no. 1, pp. 1–11, 2018.

[137] F. Obodoeze, F. Ozioko, F. Okoye, C. Mba, T. Ozue, and E. Ofoegbu, "The escalating nigeria national security challenge: Smart objects and Internet-of-Things to the rescue," *Int. J. Comput. Netw. Commun.*, vol. 1, no. 1, pp. 81–94, 2013.

[138] S. Negi, M. Jayachandran, and S. Upadhyay, "Deep fake: An understanding of fake images and videos," *Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol.*, vol. 7, no. 3, pp. 183–189, May 2021.

[139] E. Altinisik and H. T. Sencar, "Camera model identification using container and encoding characteristics of video files," 2022, *arXiv:2201.02949*.

[140] I. C. Camacho and K. Wang, "A comprehensive review of deep-learning-based methods for image forensics," *J. Imag.*, vol. 7, no. 4, p. 69, 2021.

[141] A. Berdich and B. Groza, "Smartphone camera identification from low-mid frequency DCT coefficients of dark images," *Entropy*, vol. 24, no. 8, p. 1158, 2022.

[142] A. Kurtz, H. Gascon, T. Becker, K. Rieck, and F. Freiling, "Fingerprinting mobile devices using personalized configurations," *Proc. Privacy Enhancing Technol.*, no. 1, pp. 4–19, 2016.

[143] 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, *arXiv:1605.08763*.

[144] J. Zhang, A. R. Beresford, and I. Sheret, "SensorID: Sensor calibration fingerprinting for smartphones," in *Proc. IEEE Symp. Security Privacy (SP)*, 2019, pp. 638–655.

[145] H. Bojinov, Y. Michalevsky, G. Nakibly, and D. Boneh, "Mobile device identification via sensor fingerprinting," 2014, *arXiv:1408.1416*.

[146] A. E. Dirik, H. T. Sencar, and N. Memon, "Digital single lens reflex camera identification from traces of sensor dust," *IEEE Trans. Inf. Forensics Security*, vol. 3, no. 3, pp. 539–552, Sep. 2008.

[147] A. Lawgaly, F. Khelifi, and A. Bouridane, "Image sharpening for efficient source camera identification based on sensor pattern noise estimation," in *Proc. IEEE 4th Int. Conf. Emerg. Security Technol.*, 2013, pp. 113–116.

[148] Y. Zheng, Y. Cao, and C.-H. Chang, "A new event-driven dynamic vision sensor based physical unclonable function for camera authentication in reactive monitoring system," in *Proc. IEEE Asian Hardw. Oriented Security Trust (AsianHOST)*, 2016, pp. 1–6.

[149] S. Khan and T. Bianchi, "Fast image clustering based on camera fingerprint ordering," in *Proc. IEEE ICME*, 2019, pp. 766–771.

[150] H. Tian, Y. Xiao, G. Cao, Y. Zhang, Z. Xu, and Y. Zhao, "Daxing smartphone identification dataset," *IEEE Access*, vol. 7, pp. 101046–101053, 2019.

[151] B. Hadwiger and C. Riess, "The Forchheim image database for camera identification in the wild," in *Proc. Int. Conf. Pattern Recognit.*, 2021, pp. 500–515.

[152] C. Chen and M. C. Stamm, "Robust camera model identification using demosaicing residual features," *Multimedia Tools Appl.*, vol. 80, no. 8, pp. 11365–11393, 2021.

[153] C. You, H. Zheng, Z. Guo, T. Wang, and X. Wu, "Multiscale content-independent feature fusion network for source camera identification," *Appl. Sci.*, vol. 11, no. 15, p. 6752, 2021.

[154] X. Meng, K. Meng, and W. Qiao, "A survey of research on image data sources forensics," in *Proc. 3rd Int. Conf. Artif. Intell. Pattern Recognit.*, 2020, pp. 174–179.

[155] S. Gupta, N. Mohan, and M. Kumar, "A study on source device attribution using still images," *Arch. Comput. Methods Eng.*, vol. 28, pp. 2209–2223, Jun. 2021.

[156] T. Gloe and R. Böhme, "The'Dresden image database'for benchmarking digital image forensics," in *Proc. ACM Symp. Appl. Comput.*, 2010, pp. 1584–1590.

[157] D. Shullani, M. Fontani, M. Iuliani, O. Al Shaya, and A. Piva, "VISION: A video and image dataset for source identification," *EURASIP J. Inf. Security*, no. 1, pp. 1–16, 2017.

[158] Y. Quan, C.-T. Li, Y. Zhou, and L. Li, "Warwick image forensics dataset for device fingerprinting in multimedia forensics," in *Proc. IEEE Int. Conf. Multimedia Expo*, 2020, pp. 1–6.

[159] F. D. O. Costa, E. Silva, M. Eckmann, W. J. Scheirer, and A. Rocha, "Open set source camera attribution and device linking," *Pattern Recognit. Lett.*, vol. 39, pp. 92–101, Apr. 2014.

[160] "IEEE signal processing cup 2018 database—Forensic camera model identification." 2018. [Online]. Available: https://ieee-dataport.org/open-access/ieee-signal-processing-cup-2018-database-forensic-camera-model-identification

[161] M. De Marsico, M. Nappi, D. Riccio, and H. Wechsler, "Mobile iris challenge evaluation (MICHE)-I, biometric Iris dataset and protocols," *Pattern Recognit. Lett.*, vol. 57, pp. 17–23, May 2015.

[162] A. Kumar and C. Wu, "Automated human identification using ear imaging," *Pattern Recognit.*, vol. 45, no. 3, pp. 956–968, 2012.

[163] E. G. Sánchez, *Biometría de la Oreja*, Universidad de Las Palmas de Gran Canaria, Las Palmas, Spain, 2008.

[164] D. Frejlichowski and N. Tyszkiewicz, "The West Pomeranian University of Technology ear database–A tool for testing biometric algorithms," in *Proc. 7th Int. Conf. Image Anal. Recognit. (ICIAR)* 2010, Póvoa de Varzin, Portugal, Jun. 2010, pp. 227–234.

[165] Ž. Emeršič, V. Štruc, and P. Peer, "Ear recognition: More than a survey," *Neurocomputing*, vol. 255, pp. 26–39, Sep. 2017.

[166] C. Hanilci, F. Ertas, T. Ertas, and Ö. Eskidere, "Recognition of brand and models of cell-phones from recorded speech signals," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 2, pp. 625–634, Apr. 2012.

[167] D. Luo, P. Korus, and J. Huang, "Band energy difference for source attribution in audio forensics," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 9, pp. 2179–2189, Sep. 2018.
[168] A. Hafeez, K. M. Malik, and H. Malik, "Exploiting frequency response for the identification of microphone using artificial neural networks," in *Proc. Audio Forensics*, 2019, pp. 1–18.
[169] V. Zue, S. Seneff, and J. Glass, "Speech database development at MIT: TIMIT and beyond," *Speech Commun.*, vol. 9, no. 4, pp. 351–356, 1990.
[170] L. Zou, Q. He, and J. Wu, "Source cell phone verification from speech recordings using sparse representation," *Digit. Signal Process.*, vol. 62, pp. 125–136, Mar. 2017.
[171] J. Ortega-Garcia, J. Gonzalez-Rodriguez, and V. Marrero-Aguiar, "AHUMADA: A large speech corpus in spanish for speaker characterization and identification," *Speech Commun.*, vol. 31, nos. 2–3, pp. 255–264, 2000.
[172] M. Alsulaiman, Z. Ali, G. Muhammed, M. Bencherif, and A. Mahmood, "KSU speech database: Text selection, recording and verification," in *Proc. IEEE Eur. Model. Symp.*, 2013, pp. 237–242.
[173] G. Baldini, G. Steri, F. Dimc, R. Giuliani, and R. Kamnik, "Experimental identification of smartphones using fingerprints of built-in micro-electro mechanical systems," *Sensors*, vol. 16, no. 6, p. 818, 2016.
[174] X.-Y. Li et al., "Finding the stars in the fireworks: Deep understanding of motion sensor fingerprint," *IEEE/ACM Trans. Netw.*, vol. 27, no. 5, pp. 1945–1958, Oct. 2019.
[175] J. Zhang, A. R. Beresford, and I. Sheret, "Factory calibration fingerprinting of sensors," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 1626–1639, 2020.
[176] M. Cristea and B. Groza, "Fingerprinting smartphones remotely via ICMP timestamps," *IEEE Commun. Lett.*, vol. 17, no. 6, pp. 1081–1083, Jun. 2013.
[177] T. Stöber, M. Frank, J. Schmitt, and I. Martinovic, "Who do you sync you are? Smartphone fingerprinting via application behaviour," in *Proc. 6th ACM Conf. Security Privacy Wireless Mobile Netw.*, 2013, pp. 7–12.
[178] M. Huynh, P. Nguyen, M. Gruteser, and T. Vu, "Poster: Mobile device identification by leveraging built-in capacitive signature," in *Proc. 22nd ACM SIGSAC Conf. Comput. Commun. Security*, 2015, pp. 1635–1637.
[179] W. Wu, J. Wu, Y. Wang, Z. Ling, and M. Yang, "Efficient fingerprinting-based android device identification with zero-permission identifiers," *IEEE Access*, vol. 4, pp. 8073–8083, 2016.
[180] Z. Khodzhaev, C. Ayyildiz, and G. K. Kurt, "Device fingerprinting for authentication," in *Proc. Elektrik-Elektronik ve Biyomedikal Muhendisligi Konferansi (ELECO)*, 2018, pp. 193–197.
[181] A. Vastel, P. Laperdrix, W. Rudametkin, and R. Rouvoy, "FP-scanner: The privacy implications of browser fingerprint inconsistencies," in *Proc. 27th USENIX Security Symp.*, 2018, pp. 135–150.
[182] J. Chen, K. He, J. Chen, Y. Fang, and R. Du, "PowerPrint: Identifying smartphones through power consumption of the battery," *Security Commun. Netw.*, vol. 2020, p. 13, Nov. 2020.
[183] D. Yang, G. Xing, J. Huang, X. Chang, and X. Jiang, "QID: Identifying mobile devices via wireless charging fingerprints," in *Proc. IEEE/ACM 5th Int. Conf. IoT Design Implement.*, 2020, pp. 1–13.
[184] E. Uzundurukan, Y. Dalveren, and A. Kara, "A database for the radio frequency fingerprinting of Bluetooth devices," *Data*, vol. 5, no. 2, p. 55, 2020.
[185] V. Dhakal, A. M. Feit, P. O. Kristensson, and A. Oulasvirta, "Observations on typing from 136 million keystrokes," in *Proc. CHI Conf. Human Factors Comput. Syst.*, 2018, pp. 1–12.
[186] K. Palin, A. Feit, S. Kim, P. Kristensson, and A. Oulasvirta, "How do people type on mobile devices," in *Proc. 21st Int. Conf. Human Comput. Int. Mobile Dev. Services (MobileHCI)*, 2019, pp. 1–12.
[187] T. Laor et al., "DRAWNAPART: A device identification technique based on remote GPU fingerprinting," 2022, *arXiv:2201.09956*.
[188] I. Shumailov, L. Simon, J. Yan, and R. Anderson, "Hearing your touch: A new acoustic side channel on smartphones," 2019. [Online]. Available: https://arxiv.org/abs/1903.11137
[189] W. Dai, M. Qiu, L. Qiu, L. Chen, and A. Wu, "Who moved my data? Privacy protection in smartphones," *IEEE Commun. Mag.*, vol. 55, no. 1, pp. 20–25, Jan. 2017.
[190] Y. Yang, X. Du, and Z. Yang, "PRADroid: Privacy risk assessment for Android applications," in *Proc. IEEE Int. CSP*, 2021, pp. 90–95.
[191] Android. "Sensors overview." 2022. Accessed: Nov. 11, 2022. [Online]. Available: https://developer.android.com/guide/topics/sensors/sensors_overview#sensors-rate-limiting
[192] Android. "Sensors off." 2022. Accessed: Nov. 11, 2022. [Online]. Available: https://source.android.com/docs/core/interaction/sensors/sensors-off
[193] Apple. "Getting raw accelerometer events." 2022. Accessed: Nov. 11, 2022. [Online]. Available: https://developer.apple.com/documentation/coremotion/getting_raw_accelerometer_events
[194] Apple. "Gyroscope and accelerometer." 2022. Accessed: Nov. 11, 2022. [Online]. Available: https://developer.apple.com/design/human-interface-guidelines/inputs/gyro-and-accelerometer/
[195] S. R. Small et al., "Impact of reduced sampling rate on accelerometer-based physical activity monitoring and machine learning activity classification," *J. Meas. Phys. Behav.*, vol. 4, no. 4, pp. 298–310, 2021.
[196] I. Sanchez-Rola, I. Santos, and D. Balzarotti, "Clock around the clock: Time-based device fingerprinting," in *Proc. ACM SIGSAC Conf. Comput. Commun. Security*, 2018, pp. 1502–1514.
[197] D. Ahmed, A. Das, and F. Zaffar, "Analyzing the feasibility and generalizability of fingerprinting Internet of Things devices," *Proc. Privacy Enhanc. Technol.*, no. 2, pp. 578–600, 2022.
[198] A. Schaller et al., "Intrinsic rowhammer PUFs: Leveraging the rowhammer effect for improved security," in *Proc. IEEE Int. Symp. Hardw. Oriented Security Trust*, 2017, pp. 1–7.