Sweep-to-Unlock: Fingerprinting Smartphones based on Loudspeaker Roll-off Characteristics

Adriana Berdich, Bogdan Groza, René Mayrhofer, Efrat Levy, Asaf Shabtai and Yuval Elovici

Abstract—Fingerprinting smartphones based on acoustic characteristics of their loudspeaker may have a number of applications in device-to-device authentication as well as in forensic investigations. In this work we propose an efficient fingerprinting methodology by using the roll-off characteristics of the device speaker, i.e., the transition between the low and high stopbands to the passband segment of the speaker. We extract roll-off characteristics from sweep signals, also know as chirps, that are commonly used in practice to test speaker response. This procedure appears to be more stable against variations of the volume level and allows the use of simple linear approximations, which are intuitive and easy to compute, in order to extract the fingerprint. To increase detection accuracy, on the basis of the proven performance of deep learning techniques, a convolutional and a bi-directional long short term memory neural network are further proposed and their performance demonstrated for authentication purposes. While numerous applications may be envisioned, we specifically focus on the use of speaker characteristics in relation to in-vehicle infotainment units, checking if recordings from these units can be used to fingerprint a specific phone.

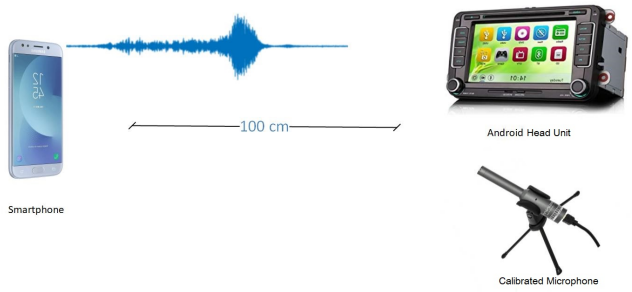


Fig. 1. Suggestive depiction of our setup: sound emitted by a smartphone is recorded by an Android headunit or microphone

1 Introduction

Device-to-device (D2D) authentication is a frequent task in IoT scenarios and using device's characteristics to assure authentication is crucial in order to remove user interaction — especially for embedded devices that do not have capable user interfaces or inside cars where physical access to the interface may be restricted. Since each physical sensor (e.g. microphone, accelerometer, etc.) or transducer (e.g. speaker) has its own characteristics, extracting fingerprints from the device peripherals is an immediate alternative. But extracting specific characteristics or deciding which classifier should be used is not always straightforward while external factors, such as environmental noise, may cause additional problems.

There are quite a number of related works in this direction, as we later discuss, and many characteristics that are specific to audio signals, e.g., mel-frequency cepstral coefficients, spectral centroid, spectral kurtosis, etc., as well as machine-learning classifiers have been proposed. In this work we study the use of roll-off characteristics of the device speaker which are extracted from a linear sweep signal. Computing the slope of the roll-off requires only a simple linear approximation and is very intuitive as a fingerprint. We show that even this simple characteristic works very well to separate between distinct devices. For higher accuracy however, on speakers coming from the same smartphone model, we further rely on deep learning algorithms that give much higher identification success rates.

Concept summary. Figure 1 provides a graphic depiction of our setup. We suggest that a smartphone is fingerprinted based on recordings done by an in-vehicle headunit, which is the main component in the scenario that we target. In this way, in-vehicle infotainment units may use the device fingerprint in order to unlock certain functions and users may authenticate without using physical keys based on the device characteristics. A similar head unit was used to make the 3.000 recordings with the 28 devices from our experiments. While our proposed concept should also be applicable to other scenarios, within the scope of this paper we focus on the in-vehicle setting as a first area of analysis and use recordings performed by a vehicle headunit. Our specific interest for this scenario comes from the recent intentions of the industry and researchers in using smartphones as keys for smart vehicles, e.g., [1], [2], [3] or [4], a task in which phone identification by vehicle headunits may find an immediate application. Nonetheless, the use of physical characteristics has been suggested as an authentication method in several automotive scenarios, e.g., for the generation of secure keys and component identification [5]. Several sources for creating physical unclonable functions (PUFs) have been suggested for automotive environments, including SRAM [5], optical channels [6] and look-up tables (LUTs)

Adriana Berdich and Bogdan Groza are with the Faculty of Automatics and Computers, Politehnica University of Timisoara, Romania, René Mayrhofer is with Institute of Networks and Security and LIT Secure and Correct Systems Lab, Johannes Kepler University Linz, Austria, Efrat Levy, Asaf Shabtai and Yuval Elovici are with the Faculty of Information Systems Engineering, Ben-Gurion University of the Negev, Israel. Email: {adriana.berdich, bogdan.groza}@aut.upt.ro, rm@ins.jku.at, elevy@post.bgu.ac.il, {shabtaia, elovici}@bgu.ac.il. Corresponding author: Bogdan Groza.

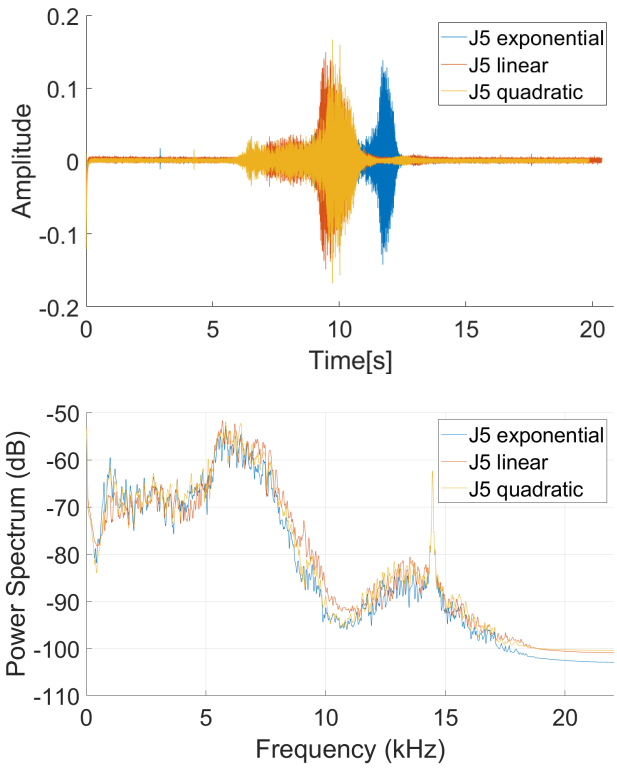


Fig. 2. Measurements in the time domain (top) and frequency domain (bottom) for the three types of chirps played by a Samsung J5 (linear, quadratic and exponential)

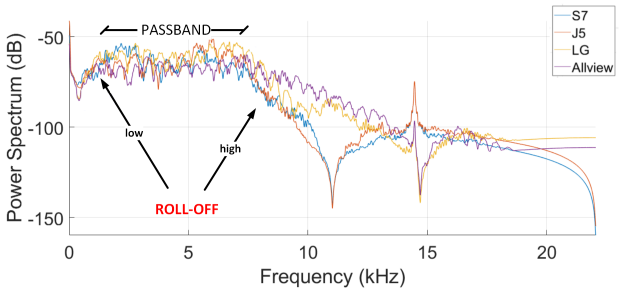


Fig. 3. Frequency sectors after applying a smoothness filter

[7]. As Android phones and head units become ubiquitous, other sources such as sound and vibrations will become available for use. Recent results in [8] also suggest the in-vehicle environment as an exemplary scenario for non-interactive device pairing.

In an analogy to the now customary *swipe-to-unlock* action, the title of our work suggests the use of *sweep* signals as acoustic fingerprints for smartphones which may be recognized and allow their owners to gain access to, i.e., unlock, other devices. The *linear sweep* function is commonly used to test speaker response by HiFi enthusiasts and professionals. The *sweep signal*, also referred to as *chirp*, is a signal in which the frequency increases over time. Ideally for speakers, one targets a linear response in the 20Hz-20kHz range (or even further), but due to inherent technological limitations the response is not linear and rises or falls at the low and high ends respectively. Three kinds of chirps are commonly used and they are readily available in the Matlab numerical environment: i) linear, i.e., $f(t) = f_0 + \frac{(f_1-f_0)}{t_1}t$, ii) quadratic $f(t) = f_0 + \frac{(f_1-f_0)}{t_1^2}t^2$ and iii) exponential, i.e., $f(t) = f_0\,(f_1/f_0)^{t/t_1}$. Here t denotes time, f_0 is the start frequency at time 0 and f_1 the instantaneous frequency at time t_1. Concretely, in our implementation we used $f_0 = 20$Hz, $f_1 = 20$kHz, $t_1 = 10$s. Figure 2 shows the result from the three chirp functions in the time and frequency domain after they are processed following a recording from an in-vehicle infotainment unit. The signals were played by one of the phones in our experiments, i.e., a Samsung J5. Note that in the time domain differences are more visible since the frequencies change distinctly with time. The frequency response, i.e., the power spectrum, is similar for the same phone, which is expected since in all three types of chirps, i.e., linear, quadratic, and exponential, the same frequency range 20-20kHz is probed.

However, for distinct speakers, the frequency response is quite distinct since speakers do not cope well at the chirp edges: bass response is limited while the high frequencies may start to beam, all these causing distinct roll-offs. Figure 3 shows the power spectrum as recorded from four smartphones: Samsung S7 (blue), Samsung J5 (red), LG Optimus P700 (orange) and Allview V1 Viper I (magenta) that performed a linear sweep function between 20Hz and 20kHz. The recordings were done by the same Android Infotainment Unit and the plot is done in Matlab based on the recorded data after applying a smoothness filter. Specifically, after recording the linear sweep function from 20Hz to 20kHz we split the range into three sectors: i) the first sector between 700Hz and 3kHz, ii) the second sector between 3kHz and 7kHz and iii) the last sector between 7kHz and 11kHz. Based on the power spectrum of the recording, computed in Matlab, speaker response was poor below 700Hz and above 11kHz, a reason for which we decided to focus our analysis in the 700Hz-11kHz range. It is easy to distinguish a passband sector in the middle which corresponds to the midrange frequencies of the speaker. The left and right stop-bands cause the low and high roll-offs, they represent a frequency range that the phone speaker has trouble to reproduce. This separation into three sectors corresponding to the low, middle and high frequencies is natural, e.g., most high-end HiFi system employ a 3-way architecture that uses distinct drivers for reproducing the bass, midrange and treble. Selecting 700Hz, 3kHz, 7kHz and 11kHz as cut-off frequencies was done based on empirical observation of the rising and falling edges of the signals and it fit well the devices in our experiments. The heterogeneity of the selected devices suggests these ranges should be suitable for most smartphones. Finally, the neural network classifiers presented in the experimental section can easily cope with the full audio spectrum, usually in the range of 20Hz-20kHz but as shown in Figure 3 smartphone speakers are hardly capable of covering it. For this reason we use only the band of 700Hz-11kHz for our neural network classifiers which reduces the computational overhead, i.e., smaller inputs, and also eliminates sectors which will be more easily susceptible to noise (since loudspeaker response is mostly absent in that area). By careful analysis, we determine that the roll-off, i.e., the slope of this transfer function, provides a good characteristic that is specific to each device. This is already visible in Figure 3 and we explore more on this in the forthcoming sections. Briefly, the contributions of our work can be summarized as follows:

1) we build a comprehensive data-set containing 3000 samples collected from 28 devices which will be publicly released to serve for future investigations,

2) we explore the use of a simple classifier based on linear approximations of the slope roll-off which proves to be a good discriminator especially between different smartphone models,
3) to grasp on finer grain characteristics and distinguish between identical speakers, we design two deep neural network architectures that have high accuracy in distinguishing identical devices,
4) to account for environmental changes, we study the influence of both environmental noise as well as of synthetic noise over the recordings.

The rest of the work is organized as follows. In Section 2 we survey some related works. Section 3 sets the background and methodology also pointing out to some limitations in related works. Section 4 presents our identification results with simple linear approximations both on distinct smartphones and speakers from the same smartphone model. In Section 5 we proceed to an analysis based on deep neural networks which leads to very small false acceptance or false rejection rates even for identical speakers that are extracted from identical smartphones and mounted on the same device (to avoid influences from the electronic circuits inside the phone). Section 6 holds the conclusion of our work.

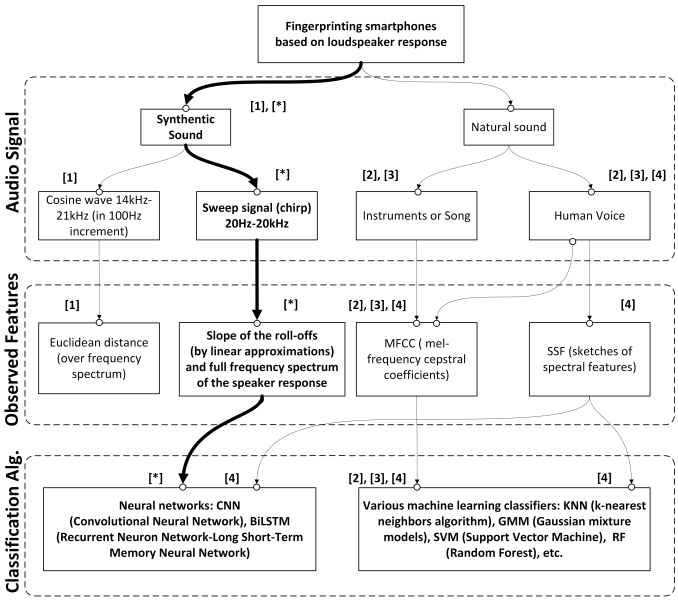


Fig. 4. Paths taken by various works in fingerprinting smartphones based on accoustic data ([*] refers to this work)

2 Related work

There are only a few works so far that have focused on fingerprinting smartphones based on the device speaker alone. Figure 4 provides an overview of the paths taken by these works. Since we are interested in device-to-device authentication we rely on synthetically produced sounds rather than on musical instruments or human voice. We also avoid the use of human voice for fingerprinting because of privacy concerns. The earliest work which took this approach is [9] which fingerprints smartphones based on the frequency response of speakers between 14kHz and 21kHz at 100Hz frequency steps. The fingerprints are compared based on the Euclidean distance to the reference data. We note that one potential limitation is that by increasing the volume, the distance increases as well making the fingerprint problematic. In contrast, the slope of the roll-off should provide better resilience to this. Nonetheless, not all of the smartphones from our experiments were capable of correctly reproducing frequencies around 20kHz and thus fingerprinting may become unreliable for some phones (the aforementioned paper uses speakers of the same smartphone model).

Rather than synthetic sounds, other works have used natural sounds, such as human voice or instrumental music, to fingerprint the device. The works from [10] and [11] are fingerprinting devices based on human voice or instrumental music by using features such as mel-frequency cepstral coefficients (MFCCs), root mean square (RMS), spectral centroid, entropy, skewness, kurtosis, tonal centroid, and others. K-nearest neighbors (KNN) and Gaussian mixture models (GMM) are used as classification algorithms to identify the device. According to [11], the best results are obtained using MFCCs (we omit the rest of the features from the drawing in Figure 4 to avoid overloading). As we later discuss, this approach appears to have the same limitation by being highly dependent on the volume level of the device. More recently, the work in [12] uses human speech along with various classifiers (support vector machine (SVM), Random Forest, etc.) and convolutional neural networks (CNN) as well as and Recurrent Neuron Network-Long Short-Term Memory Neural Network (RNN-BLSTM) for smartphone identification.

Other works have used microphones to extract fingerprints rather than loudspeakers. Note that in our authentication scenario we assume that the recording device is fixed, in particular, we used the vehicle head-unit in Figure 5 for all the recordings and a calibrated UMIK-1 microphone from time to time only to check the accuracy of our measurements. This is done for example by the works in [13] and [14] which use CNN and also KNN and SVM classifiers for smartphone identification based on the microphone response. The smartphones record and save a tone at 1kHz and 2kHz and the frequency domain representation of the normalized recorded sound is used for classification. In [15] the use of environmental sounds such as pneumatic hammer and gunshot is also considered for smartphone classification based on microphone response. The microphone is also identified based on Gaussian Supervector (GSV) using SVM and sparse representation-based (SRC) classifiers in [16]. For forensics purposes, the authors in [17] use Band Energy Differences (BED) to prove that a recording was done with a particular device. Electrical Network Frequency (ENF) analysis is used to identify the recording device in [18]. The extracted features are later used with an SVM classifier for device identification. Smartphone identification based on encoding parameters of various audio files, e.g., MP3, AAC and M4A, is studied by the authors in [19]. In [20], MFCCs is used for recorder recognition based on audio signals. In a distinct vein, serving as a protection mechanism for user's privacy, the authors in [21] propose to modify the frequency response of the phone loudspeaker in order to avoid user from being tracked.

Other systems for mobile device fingerprinting are based on audio signals and multiple motion smartphone sensors, e.g., accelerometer, gyroscope are proposed in [22] and [23]. The device speakers along with other smartphone sensors such as the accelerometer, gyroscope, and the magnetometer are used in [24]. We note that fingerprinting approaches based on sensor characteristics that require on-device measurement through respective apps such as [25] are complementary to our focus of fingerprinting

smartphones based on externally measurable attributes. In [26] the mobile devices are identified based on speakers, microphones and accelerometer data. A survey on various smartphone sensors that can be used for fingerprinting can be found in [27] and an overview on techniques for secure device pairing is available in [28].

Besides fingerprinting, other works have linked authentication and key-exchange protocols to the fingerprint (this is indeed a straight-forward extension). An authentication protocol for two devices based on the frequency response is proposed in [29]. The authors fingerprint the speakers and the microphones from different smartphones in distinct locations by using frequency domain analysis. Also, in [30] an authentication protocol is proposed for mobile devices based on acoustic channel response. A smartphone authentication scenario, based on the frequency response of the loudspeaker, non-uniformity of camera and accelerometer features in the time and frequency domain is proposed in [31]. Another system for transmitting the data between mobile devices and fingerprinting devices based on audio signals in the inaudible range, between 17.5kHz and 21kHz is proposed in [32]. The fingerprinting is based on features of audio signals such as the RMS, the symmetry of the signal, correlation of frequency response and others. The fast Fourier transform (FFT) of the audio signal is used in [33] to extract information for authentication. Also, in [34] the audio signal is used for smartphone key exchange. Device pairing based on audio signals is also proposed in [35], [36], [37], [38] and ambient noise is particularly used by [39] and [40].

Since our work directly targets the ecosystem formed by cars and smartphones, we also consider to enumerate several works that have focused on acoustic data inside the car. The low-frequency noise inside the car with an open window at different vehicle speeds is analysed in [41]. In [42] the vehicle speed and other characteristics of the cars, e.g., the length and width of vehicles passing on the street, are estimated based on a microphone that records audio signals emitted by the car in motion, e.g., noise from the engine, tires, exhaust, and air turbulence. In [43] the authors distinguish the position of the phone between the driver and the passenger based on the audio signal emitted at a frequency greater than 15kHz. Detection of abnormal driver behaviour based on audio signals recorded by the phone using machine learning is proposed in [44]. In [45] the car is paired with the smartphone using out-of-band communication channels such as an audio channel and light. A method to prevent relay attacks in the case of key-less car access systems based on sound proximity is described in [46]. After symmetric key authentication, the car and the key begin to record the ambient sound while the key transmits this to the car which detects the proximity of the key.

Other applications based on acoustic data may be also worth mentioning for the context in our work. Device localization based on audio data extracted from the ecosystem is described in [47]. In [48] a system for tracking based on audio data is proposed. Another indoor tracking system based on audio data between 17kHz and 22kHz (emitted and received by smartphones) is proposed in [49]. The authors also analyse the smartphones frequency range for audio signals, the battery consumption at different volume levels and the influence of the distance on volume. In [50], sound propagation from the speaker to the microphone is analyzed along with sound reflections and the influence of volume levels.

Fig. 5. The headunit and four smartphones from the experiments: Allview V1 Viper I, LG Optimus P700, Samsung S7 and Samsung J5

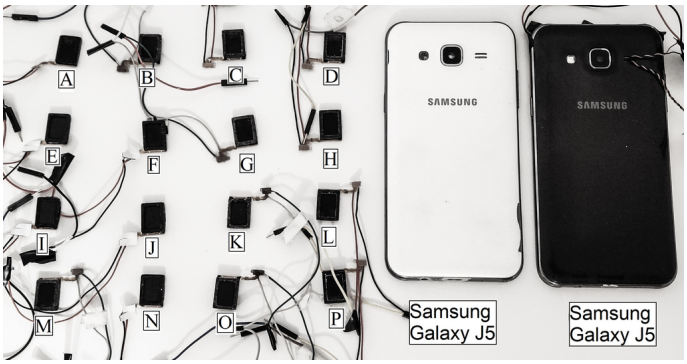


Fig. 6. Samsung J5 along with the 16 dismantled speakers and case

3 Setup and Analysis Overview

In this section we give an overview of the methodology and experiments. We discuss the environment configuration, tools, and devices used for data collection. We also give a preliminary view of the experimental data.

3.1 Setup and methodology

Devices. Our initial analysis started with several distinct smartphones. Noticing that these are easy to separate, we extended the set with 5 and later 16 identical Samsung J5 speakers which make separation more challenging. Besides these, we also use an after-market vehicle headunit manufactured by Erisin that was available to us. This unit is equipped with a microphone and supports external speakers. We are specifically interested by the vehicular setting since numerous recent works have proposed the use of smartphones for car access scenarios, e.g., [1], [2] or [3]. Four smartphones from the experiments along with the headunit are depicted in Figure 5. The 16 speakers disassembled from J5 smartphones along with a J5 case and the J5 used for fingerprinting are shown in Figure 6.

Table 1 provides a summary of the devices and associated measurements. A total of 28 devices have been fingerprinted (totaling 3000 measured sweep signals) out of which 16 are identical smartphone speakers placed in the same smartphone

TABLE 1
Summary of devices and associated measurements

	Phones	Label	No.	Meas.	Total
1.	Samsung Galaxy J5	A, C and F to P	13	100	1300
	(distinct speakers in phone)	B, D, E	3	500	1500
2.	Samsung Galaxy S7 Edge	S7	1	30	30
3.	LG Optimus P700	LG	1	30	30
4.	Allview V1 Viper I	AV	1	30	30
5.	Samsung Galaxy J5 (other)	J5′	1	30	30
6.	Samsung Galaxy J5 (other)	J5″/J5‴	2	10	20
7.	Samsung Galaxy Note 8	N8	1	10	10
8.	Samsung Galaxy A21s	A21	1	10	10
9.	Samsung Galaxy Tab S7	T7	1	10	10
10.	Xiaomi Mi A3	X3	1	10	10
11.	Xiaomi Redmi 7A	X7	1	10	10
12.	Leagoo Z10	LE	1	10	10
	Total		28		3000

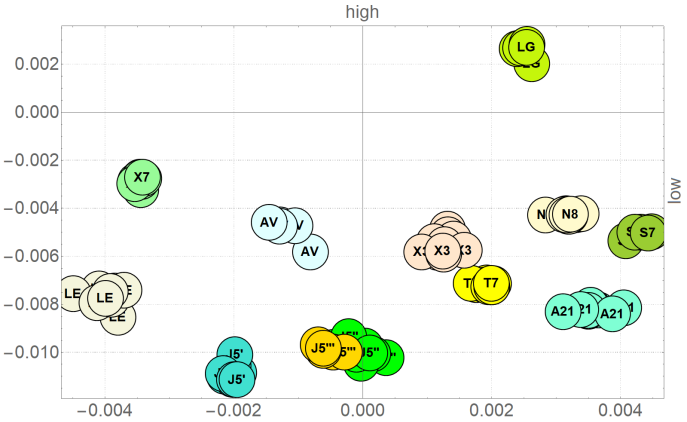


Fig. 7. Overview of the displacement according the roll-off slopes for 12 phones in our experiments

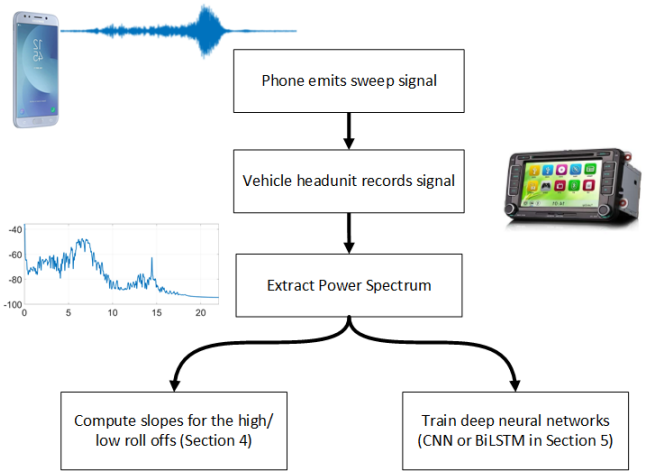


Fig. 8. Overview of the fingerprinting procedure

case (2800 measurements target these identical speakers)[1]. The first row contains the 16 identical speakers which are the main focus of our work being the most difficult to separate, i.e., a worst-case scenario. For 13 of them we took 100 measurements which were sufficient for fingerprinting with machine learning algorithms. We also selected 3 speakers out of these which had very close fingerprints and took 500 measurements to see how separation works in a much larger dataset. To serve for additional experiments, rows 2-5 contain 4 devices for which we took 30 measurements in order to test them at various volume and angle manipulations. Finally, rows 6-12 contain 8 devices, taken from casual visitors, for which we made only 10 measurements and determined that they do separate well enough even with simple linear approximations of the roll-off frequency. The separation between distinct devices is clearly visible in Figure 7 where we show the displacement for all the phones in the experiments. In this figure and similarly in several of the figures that follow, the phones (or speakers) are placed in a Cartesian coordinate system where the abscissa (x-axis) represents the slope of the *low roll-off* and the ordinate (y-axis) represents the slope of the *high roll-off*. There is clear separation between the devices but two of the J5s do overlap, i.e., J5″ and J5‴, which suggests right from the beginning that for better separation machine learning approaches are needed as the slope alone may lead to overlaps especially for similar devices.

We note that the fingerprint is dependent on the recorder as well since physical imperfections of the microphone will lead to small measurement variations. However, in the scenario that we target, i.e., device-to-device authentication, the recording device is fixed. We consider the case of a single device playing the sweep signal at a time, if multiple devices are playing then the overlap will likely make identification not possible. Background noise may have a similar effect as we discuss in a later section.

Tools and environments. We use Room EQ Wizard[2] (REW), a free room acoustic software widely used for various audio measurements. In particular, we used it to generate a LinearSweep signal with a frequency between 20Hz and 20kHz, the same signal was also generated later in Matlab which offers a more complex numerical processing environment and many other options for processing. The signal is saved as a .wav file with a sample rate of 48kHz at 16-bit resolution. The sample is played by each of the phones and the infotainment unit will record the produced sweep, more details follow in the next sections. For the off-line analysis of the recorded data, we used the Matlab[3] environment which has a vast signal processing tool-set. For a crisper image, Figure 8 provides an overview of the fingerprinting procedure. In the first step the phone, located 1m away from the headunit, emits a sweep signal which has a duration of about 10s. The headunit records the signal from which the power spectrum is extracted and used either to compute the slopes of the low and high roll-offs (Section 4) or with more complex machine-learning techniques (Section 5).

Experimental scenarios. Each smartphone plays the .wav file containing the Linear Sweep signal generated by REW. On the Erisin headunit, we run an Android application that records the sound from the smartphones. The recorded signal is saved as a .wav file for later analysis. The experiments are done in a room of $3 \times 3.7 \times 2.5\text{m}$ (room acoustics may also influence the results). Devices were placed on a working desk, the distance between the Erisin headunit and the smartphone that plays the sound is fixed at 1 meter. Regarding the volume of the smartphone, three scenarios were considered with four of the smartphones: 100% volume, 75% volume, and 50% volume. We performed five measurements in identical setups with all smartphones. For the individual J5 speakers we perform several kinds of experiments, an overview of

1. the samples from the 16 identical smartphone loudspeakers are publicly available at http://www.aut.upt.ro/~bgroza/projects/sweep-to-unlock/
2. https://www.roomeqwizard.com/
3. https://www.mathworks.com/products/matlab.html

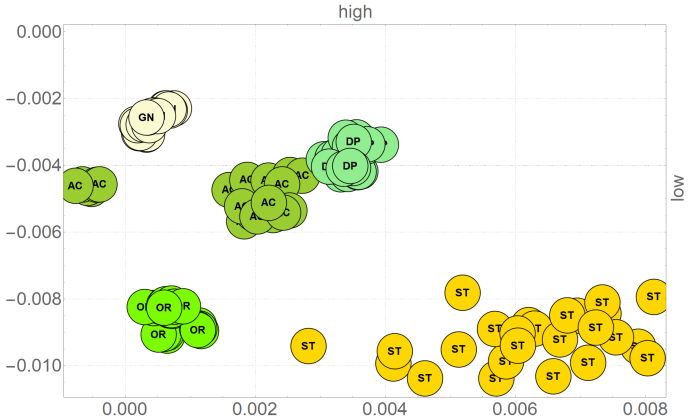


Fig. 9. Overview of the experimental results on distinct Samsung J5 speakers: speakers in original case (OR), speakers on acrylic board (AC), speakers on damping material (DP), suppression with AWGN (GN), and recording in street traffic (ST)

the results is graphically outlined in Figure 9. We first try three distinct placements of the speakers: in the original case (OR), on an acrylic board (AC) and on a damping material (DP). We further suppress the original signal with additive white Gaussian noise AWGN (GN) and perform recordings inside a car in traffic (ST). This change of environments has a clear impact on the recorded roll-offs as can be seen in Figure 9, but it is already visible that results from the same environment are clustered. A detailed discussion on separation in each such scenario will follow in the next section.

3.2 Potential limitations in related approaches

In the earlier development stages of our work we tried to use a more classical approach with simpler classifiers based on MFCCs which is also suggested in related works. However, we determined that the volume level may be misleading for such approaches and regular classifiers such as KNN do not cope well with changes in volume. That is, a classifier may correctly identify phones merely due to the output volume which is distinct on distinct phones (even if technically they are all set at the same volume level) and then mismatch them at a distinct volume (when changed by a user). In this way, correct identification is not a consequence of distinct patterns in the audio signal but rather of the volume level. When changing the volume level, the classifier fails to correctly identify the phone. The experimental data from Figure 14 discussed later, shows that roll-off characteristics remain more stable with changes in volume and orientation.

For this reason, we choose to focus our work on sweep signals and later rely on deep neural networks. In what follows we only briefly outline the potential limitation that we determined with classical machine learning algorithms applied on periodic signals. Namely, we found that by directly using the recorded audio signal with MFCCs (the discriminant recommended by [11]), the identification works mostly when the audio output is kept at the same level on distinct phones which results in fact in distinct amplitudes for the output. Once the volume changes, bringing the volume to the same actual level, the identification results become misleading.

In Table 2 we show that results by using KNN on the features extracted from a linear sweep with MFCCs at various volume levels may be inconsistent (the roll-offs that we later use seem to

TABLE 2
Misinterpretations with KNN classification and MFCCs at various volume levels 100%, 75% and 50% volume level

Volume	Phone	J5	S7	LG	AV
100%	J5	56.36%	7.18%	2.46%	34.01%
	S7	1.06%	96.57%	1.19%	1.19%
	LG	12.75%	14.70%	61.43%	11.12%
	AV	28.77%	0.57%	2.83%	67.89%
75%	J5	5.61%	80.42%	1.53%	12.45%
	S7	88.23%	6.95%	2.60%	2.22%
	LG	18.83%	33.12%	45.84%	2.21%
	AV	0.82%	75.71%	0.64%	22.84%
50%	J5	6.70%	78.60%	2.12%	12.58%
	S7	90.61%	1.82%	3.85%	3.73%
	LG	19.61%	28.98%	46.64%	2.78%
	AV	1.02%	65.47%	0.99%	32.51%

be less affected as shown in Figure 14). For the four smartphones in this experiment, we considered the features extracted from the audio signal from one experiment as training data and the features extracted from the another four experiments as test data. The first experiments show all phones at 100% volume. In this case, all smartphones are correctly identified. Then we reduced the volume for all phones to 75% and kept the same number of experiments 5. For the Samsung S7 however, we also add 5 experiments as test data with the volume set to 100%. In this case, only the LG is correctly identified while the S7 begins to overlap in identification with the J5 and the Allview. When proceeding to a reduction in volume to 50% and again adding misleading test data for S7 at 100% the situation remains similar. The volume also fluctuates and differences can be significant according to the frequency.

We have also tested the phones with a periodic tone as employed in related works, i.e., we use a sine wave $s(t) = a\sin(2\pi ft/f_s)$, where a is the amplitude, f the frequency, f_s the sampling frequency and t denotes time. This tone was used to encode a 1 while a 0 will be denoted by a period of silence. However, we noted similar differences between the volume levels of the phones. For example, at 1kHz with all phones kept at 100% volume, the J5 is louder than the rest, while the Allview is only at 68% of the J5 volume, LG at 67 % and the S7 at only 39% of the J5 volume (this is in fact visible in Figure 11 which we discuss in a later section). This distinction is enough to provide a fingerprint, but we cannot be sure of the volume level at which the user keeps the phone. Scaling the volume to the same value also changes the amplitude of the noise. As further clarifications, in Figure 10 we show the audio data from four phones as recorded by the car infotainment unit. The left side of the figure shows the original data and the right side of the figure shows the data after we scale it in order to remove differences due to the volume level. The classification works correctly at the same volume level, but it appears that the classifier is again dependent on the volume level. We return to the results obtained with the classifiers later. After the data was scaled to remove differences in volume levels (the right side of the picture), the noise level seems again to be the bigger discriminant. Noise is present during intervals when the speaker is silent and when scaling the data the noise also scales up creating a bigger discriminant. To remove measurement problems due to the poorly calibrated microphones in the car Android unit we also performed measurements with a calibrated microphone UMIK-1 from MiniDSP[4]. An example is shown in Figure 11.

4. https://www.minidsp.com/

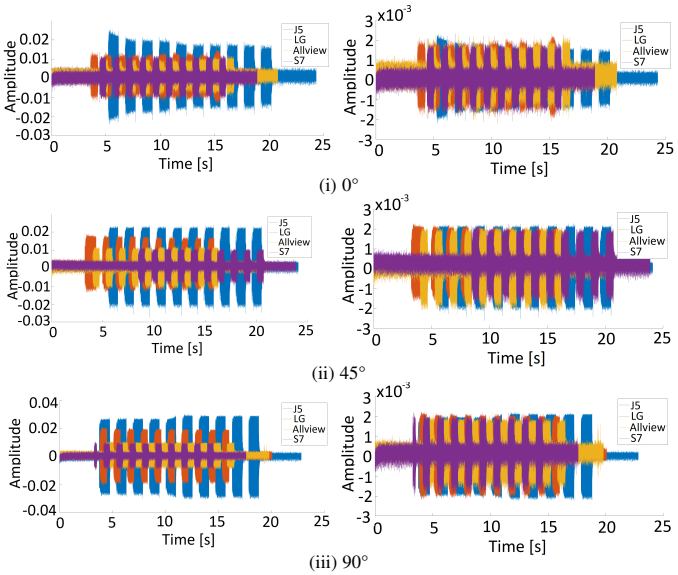


Fig. 10. Recorded signal (left) and scaled (right) on four phones for a periodic tone of 1kHz at 500ms periodicity (recordings by in-vehicle headunit)

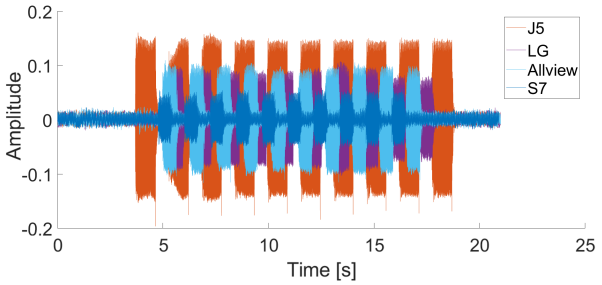


Fig. 11. Recorded signal on four phones for a periodic tone of 1kHz at 500ms periodicity with a HiFi setup microphone UMIK-1

While the noise level is clearly much smaller and the signal level is also higher due to the higher sensitivity of the microphone, the classification results were similar to the case of the infotainment unit. Also, since periodic signals have a much simpler frequency domain representation we were not comfortable with using them in the rest of our experiments. In Figure 12 we show the influence of the speaker placement and background material on a periodic tone (this can be contrasted to Figure 15 which shows the much more complex power spectrum of sweep signals). The figure shows the recorded signal from the five speakers for a periodic tone of 1kHz with 500ms periodicity. The signal is very poorly defined when the speakers are placed on the acrylic board (left) or on the damping material (middle). The signal gains clarity when the speakers are placed in the smartphone case (right). The power spectrum is shown in the bottom of the figure and it consists of a large spike at 1kHz.

Regarding the use of more classical machine learning algorithms, such as KNN, another limitation that we noticed was that when using the 500 samples dataset their performance was visibly lower than that of neural networks. Of course, there are many other traditional machine learning classifiers and various optimizations are possible, but we cannot address so many algorithms in a single work, a reason for which we chose to focus our contribution on the use of two newly designed deep neural networks that provided excellent results on the collected data. As a brief comparison with the deep-learning approach proposed in this work, we also add later some results for the KNN, SVM and Random Forest classifiers in the worst case scenario of our analysis, i.e., speech affected recordings, to serve as a comparison.

4 Fingerprinting Speakers Based on Roll-off Slopes

We now proceed to analyzing speakers based on their roll-off characteristics. Subsequently, for comparison, we also perform an analysis based on periodic signals that exhibit rising and falling edges at a faster rate for which we use more demanding machine learning algorithms. Finally, we analyze the impact of noise on fingerprinting speakers.

4.1 Roll-off characteristics on distinct smartphones

For a more comprehensive analysis of the signals recorded by the Erisin headunit we use the Matlab environment. To get a clear image on the recorded data we also use the Signal Analyser App from Matlab's Signal Processing Toolbox. Our analysis is based on the power spectrum of the signal, i.e., the frequencies of the spectral estimates from the power spectrum, which is extracted by calling the `pspectrum(data, sample rate)` function on the audio data and the corresponding sample rate.

Figure 13 shows plots of the power spectrum when using three volume levels: 100% volume (blue), 75% volume (orange), and 50% volume (red). The shape of the signal remains similar, but, as expected, the signal is shifted on the vertical axis, a reason which may lead to misclassification when the user (or an adversary) changes the volume level of the phone. When computing the slope of the signal by linear approximations, unwanted noise may affect the result. For this reason we also tried to reduce the noise by using a smoothness filter. This was achieved by using a moving mean filter which is implemented in the toolset by the `smoothdata(sampled data, 'movmean')` function that has as parameter the sampled data and the moving average method, i.e., '`movmean`'. We also tried other options, e.g., '`movmedian`' or '`gaussian`' but did not yield better results. Finally, for the neural networks used in Section 5 there was no need to remove the noise from the signal since it did not affect the accuracy of the result.

Based on observations from Figure 13 we analyse the frequency in the range of 700Hz – 11kHz. We split this frequency range into three sectors that are relevant for the roll-off characteristics: the first sector is between 700Hz and 3kHz, the second sector is between 3kHz and 7kHz and the last sector is between 7kHz and 11kHz as we show in figure 3 for each smartphone Samsung S7 (blue), Samsung J5 (red), LG (orange) and Allview (magenta). To separate between signals, we apply a linear approximation for each of the three sectors. For the linear approximation function, in Matlab we use the `polyfit(frequencies, power spectrum, degree)` which has as parameters the frequencies of the spectral estimates from the power spectrum, the power spectrum in decibels and the degree of the approximation polynomial (which is 1 in our case). The function returns the coefficients of an approximation polynomial of degree 1. We can also use the `polyval(polynomial coefficients,`

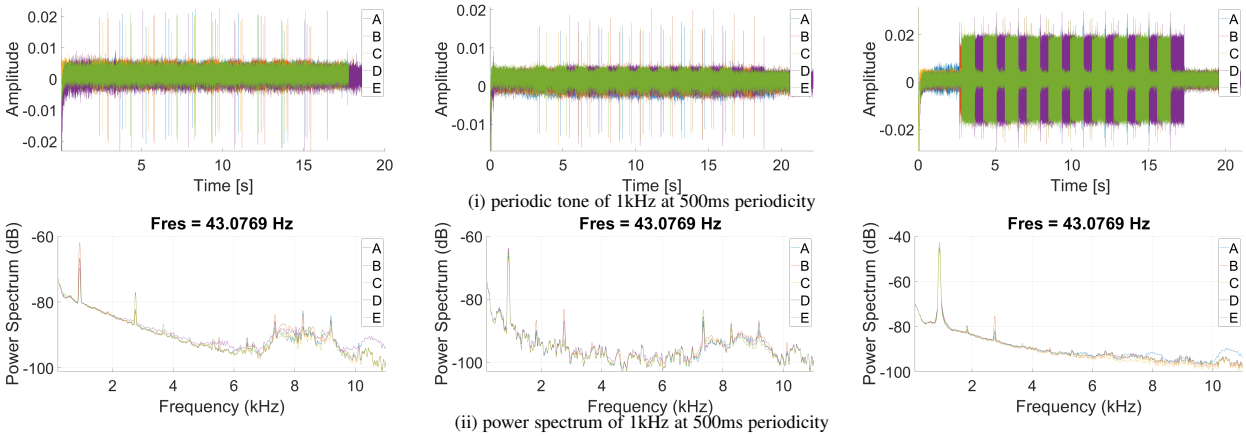


Fig. 12. Recorded signal on the five speakers on acrylic board (left), on damping material (middle) and inside the smartphone case (right) and power spectrum for a periodic tone of 1kHz at 500ms periodicity (recordings by in-vehicle headunit)

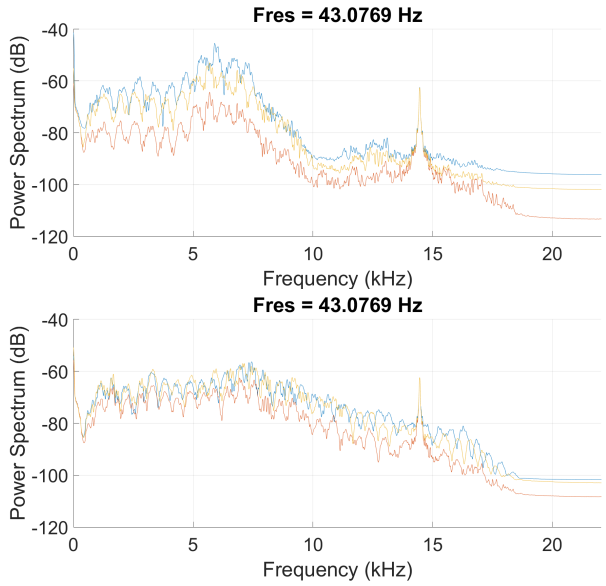


Fig. 13. Power spectrum of the audio signal at distinct volume levels 100% volume (blue), 75% volume (orange) and 50% volume (red) for Samsung J5 (up) and Allview (down)

`points)` function, which has as parameters the coefficients of the polynomial to query at evaluation points. The function returns the values of the polynomial for each point.

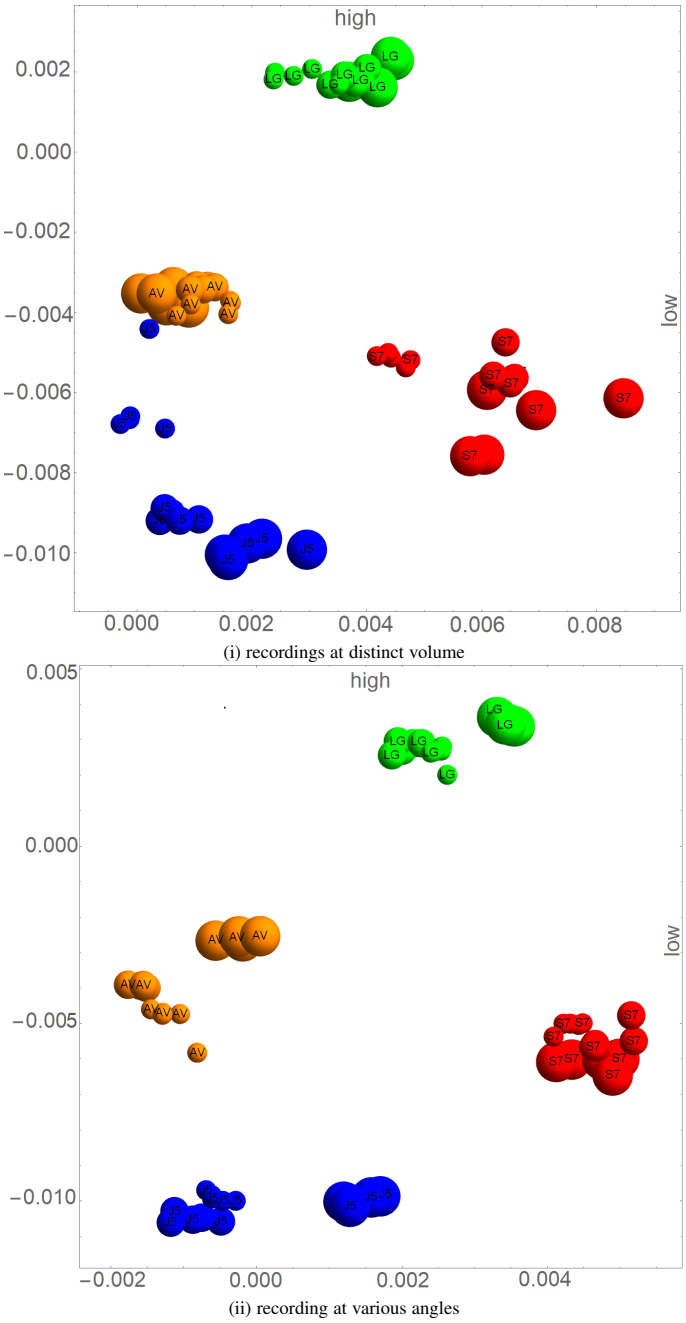
In Figure 14 we show how four phones cluster in a Cartesian coordinate system where the ordinate represents the slope of the high roll-off and the abscissa is the slope of the low roll-off at distinct volume levels (i) and distinct angles (ii). The plots account for three distinct volume levels on each phone, i.e., 50%, 75% and 100%, the size of the chart element depends on the volume level (bigger elements correspond to higher volume). Similarly, for the angle of the recording we considered three values $0°$, $45°$ and $90°$, we kept the size of the chart element inversely proportional with the angle, i.e., higher elements correspond to $0°$. It is easy to see that the four phone cluster distinctly. Volume level has some influence on clustering, but the only phone which may be misclassified is the J5 at 50% volume level which overlaps with the AV in one of the measurements. The angle has less influence and there are no overlaps between the results. To correctly separate between phones that are closer more measurements are likely needed. The results were very similar with or without the smoothness filter, to avoid overloading the paper we will generally present only the plots for the original signal (without smoothness).

4.2 Identifying speakers from the same smartphone model

We now consider to separate between different speakers from the same smartphone model which is the more challenging problem. For the Samsung Galaxy J5 we first obtained 5 identical speakers (labels A to E) on which our initial analysis is focused, then we extended this set with another 11 identical speakers (labels F to P). The speakers came from second-hand phones and were disassembled, soldered to new wires and connected to the same smartphone for the tests that followed. Many different factors may have contributed to measurable differences between the speakers, including manufacturing variations, material aging, physical stress, the different volume levels at which they were usually played, or other environmental affects during the use of those second-hand phones. However, these effects occur in the real-world use of smartphones, and thus our measurements are good indicators for practical scenarios.

We pursue three experiments in which the first five speakers are placed against three distinct background materials: an acrylic board, a sound damping material (felt) and inside the smartphone case which is the main use case. In Figure 15 we show the influence of the speakers placement and the background material. It can be easily seen that the smartphone case boosts frequencies below the 7KHz range which are poorly defined on the acrylic or damping boards. This shows that the case has a major influence over the sound of the speaker. Fortunately, for the same smartphone model the case will be identical. It is out of scope for our work to evaluate differences due to physical damage of the smartphone case or use of different outer shells.

We further investigate the separation based on low and high roll-offs against the three distinct background materials. Figure 16 shows the separation by using the first and third separation sectors. The five speakers seem to separate based on the slope of their roll-offs. The separation becomes clearer when they are placed in the original case, but it is also obvious when they are placed on the sound damping material. There is a higher amount of confusion between the speakers when they are placed on the acrylic board



(i) recordings at distinct volume

(ii) recording at various angles

Fig. 14. Clustering based on low-high roll-offs: at disctinct volume levels 50%, 75%, 100% (i) and various angles $0°$, $45°$, $90°$ (ii) - for the four phones S7 (red), J5 (blue), LG (green) and Allview (orange)

which may be due to sound reflections that are more pronounced on the acrylic board.

By looking closer at the result for the original case, which is also more relevant for practice, the separation between speakers B, D and E has to be more carefully addressed since the overlap is more pronounced. To achieve this, we first performed 100 measurements with each speaker. The clustering is depicted in Figure 17. By computing the Euclidean distance of each sample to the mean of the samples from the same speaker, i.e., the intra-distance, and to the mean of the samples from the other speaker, i.e., the inter-distance, we obtain the following separation ratios: 83% between B and D, 77% between B and E, 73% between D and E. The separation becomes obvious with repeated measurements although outliers may be present.

We now make a brief quantitative analysis on the inter-distances and intra-distances that can be extracted from the slopes of the roll-offs. Figure 18 shows the inter-distances and intra-distances between the 12 distinct smartphones both as a heatmap to the left and as numerical values to the right. The distances were computed as the average Euclidean distances between the planar coordinates formed by the slopes of the low and high roll-offs. The inter-distances are clearly greater, staying above 10^{-3} (with the exception of two identical J5 phones), while the intra-distances are always below this threshold. The diagonal of the matrix, representing intra-distances is also easy to distinguish in the heatmap. Figure 19 shows the inter-distances and intra-distances between the 16 identical J5 speakers both as a heatmap to the left and as numerical values to the right. In this case, while the intra-distances are almost always below the 10^{-3} threshold (speaker D is the only exception), it may happen for the inter-distances to drop below this threshold as well. Consequently, separation based on the slope of the roll-offs alone is more difficult. The heatmap is also more noisy and the diagonal, i.e., the intra-distances, while still visible, is harder to distinguish.

As a partial conclusion, the slopes may provide sufficient clues for separating between distinct smartphones but they may become problematic when identical speakers/phones are used. For this reason we will use in Section 5 deep neural networks to separate between identical speakers.

4.3 Noise influence on roll-offs slopes

Having in mind that in a real-world scenario environmental noise is present and can influence the fingerprinting process of the speakers, we consider to analyze the influence of noise as well. We consider two types of noise that are relevant: the *additive white Gaussian noise* (AWGN) which mimics the effects of many random processes that exist in nature and may also account for noise inside cars and the *street noise* which is specific for our car related scenario. The additive white Gaussian noise was also used to simulate the attenuation of the signal from the speaker to the microphone in [13].

We apply an AWGN signal over the clean recordings with a noise level proportional to the original signal power, i.e., by setting the SNR (signal-to-noise ratio) to 0dB. The clean recordings contain the signal recorded by the infotainment unit and played by the Samsung J5 speakers (the speakers were placed inside the smartphone case as described previously). In Figure 20 we depict the audio signal recorded and the signal with AWGN in the time domain (up) and the power spectrum of the signals (down).

We now analyze the influence of the AWGN on the linear sweep signal played by the first five speakers of the Samsung J5. Based on the power spectrum of the audio signal with AWGN, we analyze the frequency in the range of 700Hz – 11kHz and split it in three sectors as we did in the previous sections. Figure 21 shows plots from two experiments with the linear fit of the power spectral signals for the first sector (up) and the third sector (down). Each plot shows the five speakers of the Samsung J5 with 100% volume level. Based on the slope of the linear approximation for each of the three sectors we can separate between the speakers. Figure 22 shows the separation based on the low and high roll-offs. The results still show some clustering when AWGN is added but overlaps are more visible.

To analyze the influence of the *street noise* on fingerprinting the speakers, we pursue experiments with the infotainment unit

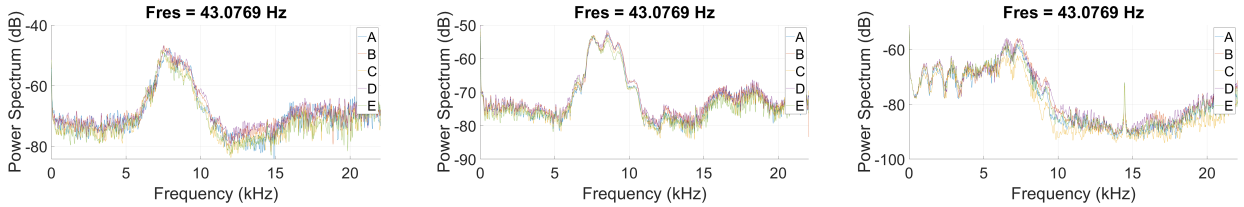


Fig. 15. Power spectrum for the recorded signal on the five speakers on acrylic board (left), on damping material (middle) and inside the smartphone case (right) for linear sweep (recordings by in-vehicle headunit)

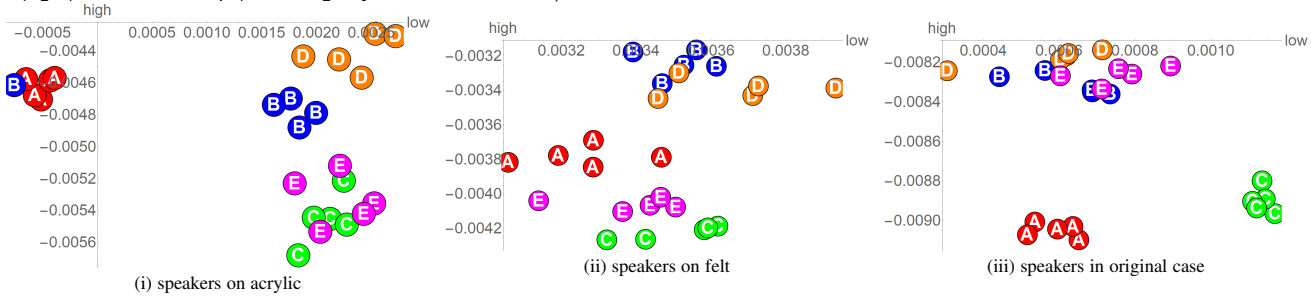


Fig. 16. Separation based on low-high roll-offs for the first five identical J5 speakers A to E

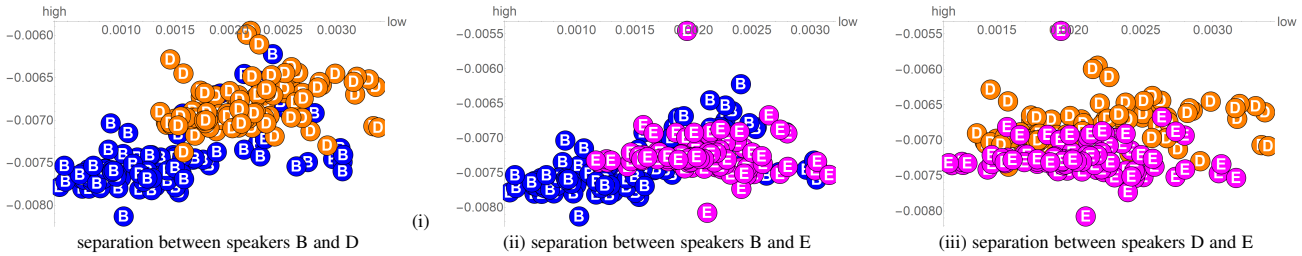
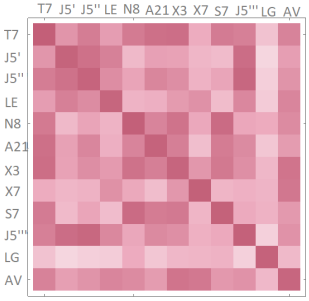


Fig. 17. Separation based on low-high roll-offs for the three closer J5 speakers B, D and E with 100 samples



	T7	J5′	J5″	LE	N8	A21	X3	X7	S7	J5‴	LG	AV
T7	**0.0001**	0.0053	0.0034	0.0059	0.0031	0.0019	0.0018	0.0069	0.0031	0.0036	0.0097	0.0038
J5′	0.0053	**0.0003**	0.0022	0.0037	0.0083	0.0061	0.0062	0.0081	0.0085	0.0018	0.0141	0.0059
J5″	0.0034	0.0022	**0.0004**	0.0045	0.0065	0.0040	0.0046	0.0078	0.0065	0.0005	0.0127	0.0051
LE	0.0059	0.0037	0.0045	**0.0004**	0.0079	0.0075	0.0056	0.0048	0.0087	0.0041	0.0121	0.0039
N8	0.0031	0.0083	0.0065	0.0079	**0.0001**	0.0039	0.0023	0.0067	0.0013	0.0067	0.0068	0.0044
A21	0.0019	0.0061	0.0040	0.0075	0.0039	**0.0002**	0.0035	0.0088	0.0032	0.0044	0.0108	0.0057
X3	0.0018	0.0062	0.0046	0.0056	0.0023	0.0035	**0.0004**	0.0054	0.0031	0.0047	0.0081	0.0025
X7	0.0069	0.0081	0.0078	0.0048	0.0067	0.0088	0.0054	**0.0001**	0.0080	0.0076	0.0080	0.0030
S7	0.0031	0.0085	0.0065	0.0087	0.0013	0.0032	0.0031	0.0080	**0.0002**	0.0068	0.0078	0.0055
J5‴	0.0036	0.0018	0.0005	0.0041	0.0067	0.0044	0.0047	0.0076	0.0068	**0.0002**	0.0128	0.0050
LG	0.0097	0.0141	0.0127	0.0121	0.0068	0.0108	0.0081	0.0080	0.0078	0.0128	**0.0002**	0.0083
AV	0.0038	0.0059	0.0051	0.0039	0.0044	0.0057	0.0025	0.0030	0.0055	0.0050	0.0083	**0.0005**

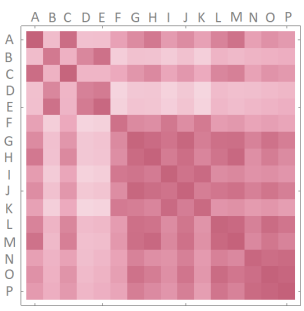
Fig. 18. Slope inter-distances and intra-distances for the 12 distinct smartphones as heatmap (left) and numerical values (right)

inside the car in a parking lot near an urban road with four lanes on a two-way street with tram lines. The infotainment unit was placed in the middle of the car dashboard and records the linear sweep signal played by the five speakers of the Samsung J5. The Samsung J5 was held by the passenger, pointing toward the infotainment unit microphone, at a distance of about 50 centimeters. To maximize the street noise, the front left window was left open. Separately, we also recorded street noise with the infotainment unit from the car with the front left window opened in the same parking lot. As proposed in [13], the recorded street noise is applied to the recorded audio signal and played by the first five speakers of the Samsung J5 placed inside the smartphone case as described in Section 3. When identical noise is added to the recording, the separation between speakers was still visible. However, with repeated measurements inside the car with the left window open, the separation became less clear as shown in Figure 23. This is very likely due to distinct street noises at each new measurement, e.g., traffic may vary between each measurement, a horn may ring, a tramway may pass nearby, etc.

This suggests that adding synthetic noise does not lead to a very good simulation of a real-world scenario, but it is by no means easy to test a high number of speakers on the street. For this reason, with the 16 identical speakers we will later use synthetic AWGN noise as employed in various related works.

5 A FINER-GRAINED ANALYSIS WITH NEURAL NETWORKS

In the previous section we used a linear approximation of the roll-offs which is a simple and effective procedure but has shortcomings in separating identical speakers, i.e., an accuracy of only 70-80% was achieved between identical speakers B, D and E. In



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
A	**0.0002**	0.0035	0.0006	0.0037	0.0037	0.0021	0.0014	0.0009	0.0019	0.0014	0.0021	0.0011	0.0008	0.0020	0.0016	0.0018
B	0.0035	**0.0009**	0.0031	0.0013	0.0008	0.0046	0.0037	0.0038	0.0044	0.0038	0.0047	0.0032	0.0034	0.0032	0.0031	0.0029
C	0.0006	0.0031	**0.0003**	0.0032	0.0032	0.0025	0.0017	0.0014	0.0023	0.0017	0.0025	0.0013	0.0011	0.0021	0.0016	0.0018
D	0.0037	0.0013	0.0032	**0.0011**	0.0010	0.0050	0.0041	0.0041	0.0048	0.0041	0.0051	0.0035	0.0037	0.0036	0.0035	0.0033
E	0.0037	0.0008	0.0032	0.0010	**0.0005**	0.0048	0.0039	0.0040	0.0046	0.0039	0.0049	0.0033	0.0036	0.0033	0.0032	0.0030
F	0.0021	0.0046	0.0025	0.0050	0.0048	**0.0008**	0.0014	0.0014	0.0009	0.0014	0.0010	0.0019	0.0017	0.0022	0.0020	0.0022
G	0.0014	0.0037	0.0017	0.0041	0.0039	0.0014	**0.0003**	0.0006	0.0008	0.0004	0.0011	0.0008	0.0007	0.0011	0.0008	0.0011
H	0.0009	0.0038	0.0014	0.0041	0.0040	0.0014	0.0006	**0.0003**	0.0010	0.0007	0.0012	0.0008	0.0005	0.0015	0.0011	0.0014
I	0.0019	0.0044	0.0023	0.0048	0.0046	0.0009	0.0008	0.0010	**0.0003**	0.0008	0.0006	0.0014	0.0013	0.0016	0.0015	0.0017
J	0.0014	0.0038	0.0017	0.0041	0.0039	0.0014	0.0004	0.0007	0.0008	**0.0003**	0.0011	0.0008	0.0008	0.0011	0.0008	0.0011
K	0.0021	0.0047	0.0025	0.0051	0.0049	0.0010	0.0011	0.0012	0.0006	0.0011	**0.0005**	0.0016	0.0015	0.0018	0.0017	0.0019
L	0.0011	0.0032	0.0013	0.0035	0.0033	0.0019	0.0008	0.0008	0.0014	0.0008	0.0016	**0.0004**	0.0005	0.0011	0.0006	0.0008
M	0.0008	0.0034	0.0011	0.0037	0.0036	0.0017	0.0007	0.0005	0.0013	0.0008	0.0015	0.0005	**0.0002**	0.0013	0.0008	0.0011
M	0.0020	0.0032	0.0021	0.0036	0.0033	0.0022	0.0011	0.0015	0.0016	0.0011	0.0018	0.0011	0.0013	**0.0005**	0.0007	0.0006
O	0.0016	0.0031	0.0016	0.0035	0.0032	0.0020	0.0008	0.0011	0.0015	0.0008	0.0017	0.0006	0.0008	0.0007	**0.0002**	0.0004
P	0.0018	0.0029	0.0018	0.0033	0.0030	0.0022	0.0011	0.0014	0.0017	0.0011	0.0019	0.0008	0.0011	0.0006	0.0004	**0.0001**

Fig. 19. Slope inter-distances and intra-distances for the 16 identical J5 speakers as heatmap (left) and numerical values (right)

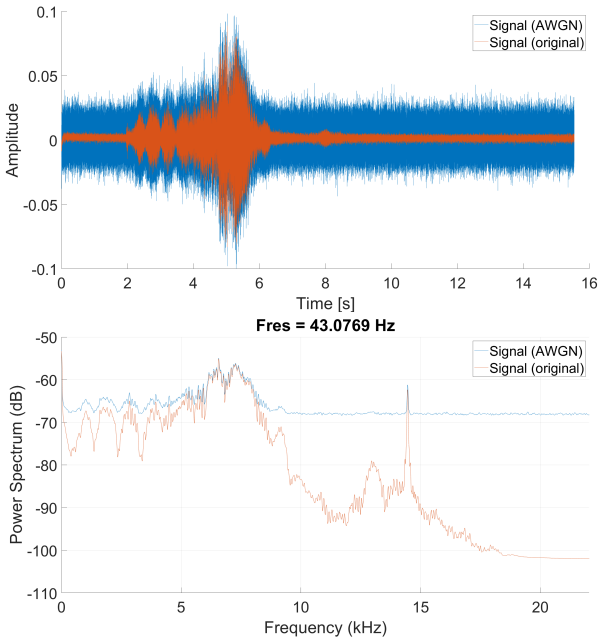


Fig. 20. The audio signal recorded and the signal with AWGN in time domain (top) and the power spectrum of the signals (bottom)

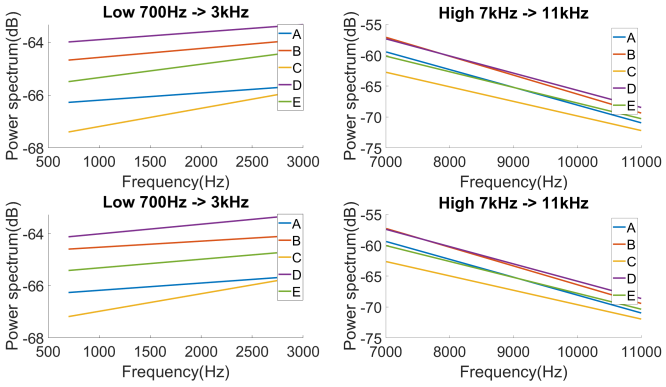


Fig. 21. Linear fit results over the audio signal recorded from 2 experiments (up, down) depicting five speakers of the Samsung J5 in the low (left) and high sector (right).

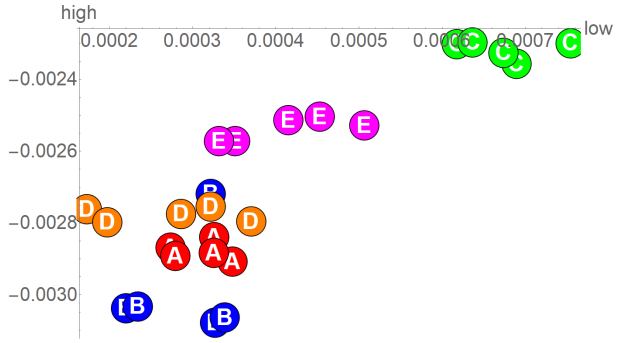


Fig. 22. Separation based on low-high roll-offs in case when the original signal is overlap with AWGN

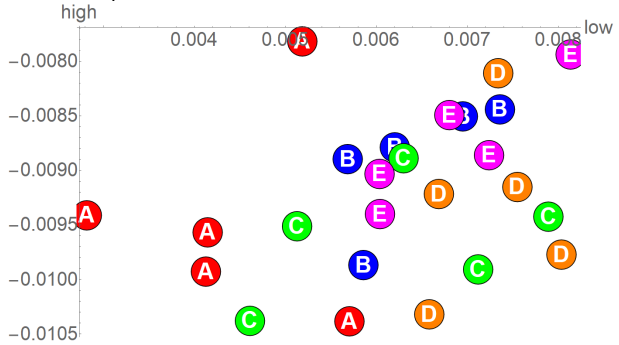


Fig. 23. Separation based on low-high roll-offs in case of street recording

this section we proceed to a finer-grained analysis based on convolutional neural networks (CNN) and bi-directional long short-term memory networks (BiLSTM) which gives an identification success rate close to 100%.

5.1 The deep neural network architectures

We now present the two deep neural network architectures that we employ in for analyzing the collected audio samples. During training, validation and testing, each of our network architectures receives as input 1914 features which are the values of the power spectrum between 700Hz and 11kHz at ∼5Hz resolution. We choose to rely on the 700Hz-11kHz since as shown by the previous power spectrum plots this is the most significant portion of the sweep signal, i.e., it is the portion which carries most of the signal power. Our datasets are however over the entire 20Hz-20kHz range and future works may attempt to use distinct portions of the sweep as well.

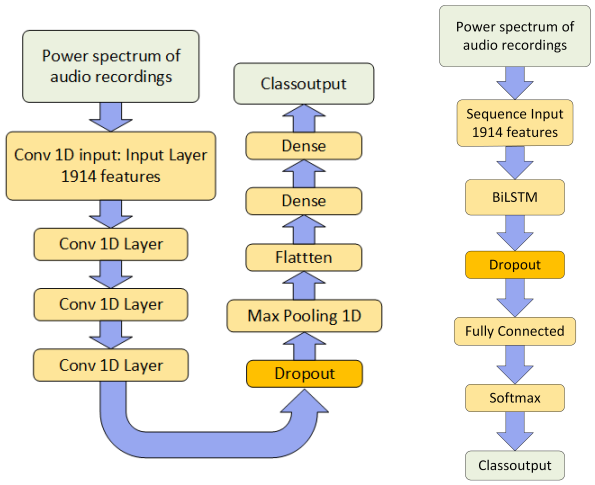


Fig. 24. The proposed CNN (left) and BiLSTM (right) architectures

CNN Classifier Architecture. We begin with the classifier that we built based on Convolutional Neural Networks (CNN). For each speaker *i*, we induce a binary classifier that is responsible for authenticating it (i.e., classify it as *'speaker i'* or *'other'*). The left side of Figure 24 shows the architecture of the CNN-based network. The following settings were used in our experiments.

- Each classifier includes three convolutional layers followed by max pooling to reduce the size. We evaluated the classifiers' performance with a larger number of convolutional layers (4 to 6 layers) and number of filters (16 to 32 filters) and no significant gain in accuracy was obtained.
- All convolutional layers use the rectified linear unit (ReLU) as an activation function.
- A Sigmoid layer with a single unit is attached. This layer is aimed at producing the probability that a given example is associated with the speaker.

During the training phase, a dropout parameter was set to 0.5 to avoid overfitting. In addition, we used the Root Mean Square Propagation (RMSProp) optimization algorithm with a decay rate set to 0.9 which is the default in the Keras API. The training process is stopped when the loss function reaches its minimum. The experiments were performed on a computer with an Intel Core i7 processor at 2.11GHz and 16GB RAM, the GPU was not used.

BiLSTM Classifier Architecture. To serve as a comparison to CNNs, we also used a Bidirectional Long Short-Term Memory (BiLSTM) Network. The structure of the network is presented in the right side of Figure 24. The proposed neural network contains an input layer followed by a BiLSTM layer with 1914 hidden units, a fully connected layer, a dropout layer, a softmax layer and a classification layer. Distinct to the CNN architecture, we noticed that the dropout parameter set to 0.25 gave better results than a dropout of 0.5 which resulted in a visible loss of accuracy. The optimization algorithm that we used was the stochastic gradient descent with momentum (SGDM) having the momentum value increased to 0.95, which represents the contribution from the previous step. The last layer of the network also returns the probability that the tested speaker belongs to one of the speakers that the network has learned. The maximum number of epochs to use for training was set to 100. These experiments were performed on a laptop with an Intel Core i7 processor at 2.6GHz and 16GB of RAM, the GPU was not used.

In what follows we analyze the performance of these two deep learning architectures in separating the 16 identical Samsung J5 speakers.

5.2 Results on clean recordings

Given our potential goal to apply this in authentication scenarios, each classifier's performance was evaluated in terms of False Rejection Rate (FRR) and False Acceptance Rate (FAR). FAR is the probability of unauthorized loudspeakers to be accepted as legitimate and FRR is the probability of authorized loudspeakers to be incorrectly rejected. We have computed the FAR and FRR as follows: $FAR = \frac{FP}{TN+FP}$; $FRR = \frac{FN}{TP+FN}$, where TP is true positive, FP is false negative, TN is true negative and FP is false positive.

First, in Figure 25 we present experiments for the CNN network in which we set $tr \in [30, 120, 210, 300, 390]$, and randomly pick tr training samples from the dataset of 500 recordings that are uniformly distributed upon speakers B, D and E which caused problems in the slope separation tests. We used the basic rule of the thumb and set 70% of these for training and 30% for validation. It can be seen that excellent results are gained when using more than 300 samples for training. The trained models were able to authenticate a smartphone with a high level of accuracy, i.e., close to 100%. We also mention that to account for possible environmental changes, the 500 recordings were collected over distinct days in chunks of 100 measurements each time for a speaker. For this reason, the 500 measurements dataset is more challenging. Figure 26 summarizes the FRR and FAR for the same number of training samples for the BiLSTM, i.e., $tr \in [30, 120, 210, 300, 390]$ out of which less than 30% were reserved for validation. For a small number of samples, i.e., 30 samples, the FRR is similar for both CNN and BiLSTM at around 30%. This is also close to the accuracy of the linear approximation technique in the previous section. The FAR is however better with the BiLSTM at around 15%. Interestingly, the BiLSTM performs much better than the CNN at 120-210 samples where the FRR and FAR quickly drop below 10%. At 300 and 390 samples the results are nearly identical with both CNNs and BiLSTMs. In particular for 390 samples, with both CNN and BiLSTM we achieve a FRR and FAR of 0-2% which we believe to be good enough for practical purposes.

We continued by testing the two proposed deep neural network architectures over all the 16 identical Samsung J5 speakers. For this, we collected a new set of 100 linear sweeps from each of the speakers from F to P. This time, all measurements were taken continuously for the same speaker. This makes the second dataset more stable. The 5 speakers from the previous experiment, i.e., labels A to E, were also used in this newer experiment but we relied on the first 100 measurements out of the previously taken 500 measurements. We split this dataset for each speaker into the following number of samples for training and validation: $tr \in \{15, 20, 35, 50, 55\}$, $v \in \{5, 10, 15, 20, 25\}$. The rest of the samples are used for testing, i.e., 80% down to 20% was used to compute the acceptance and rejection rates.

Figure 27 presents the results both as heatmaps and as numerical values for both the CNN and BiLSTM networks. Notably, for some of the speakers, e.g., F, G, H, I, J and K, the miss-identification is always 0 with the BiLSTM regardless of the size of the training set. For the CNN the situation is a bit different, there is limited confusion between these, but no confusion between

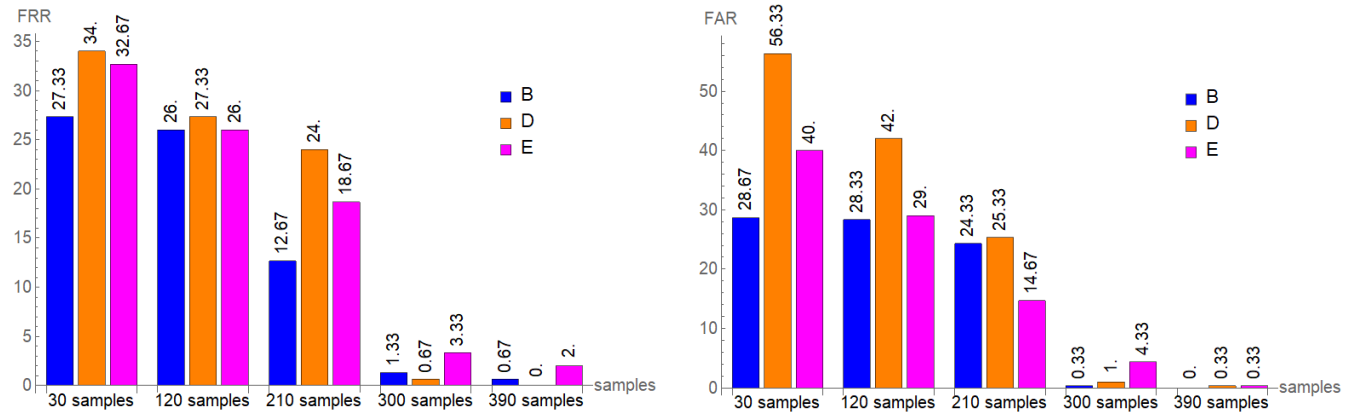


Fig. 25. FRR (left) and FAR (right) with CNN as a function of the training samples number

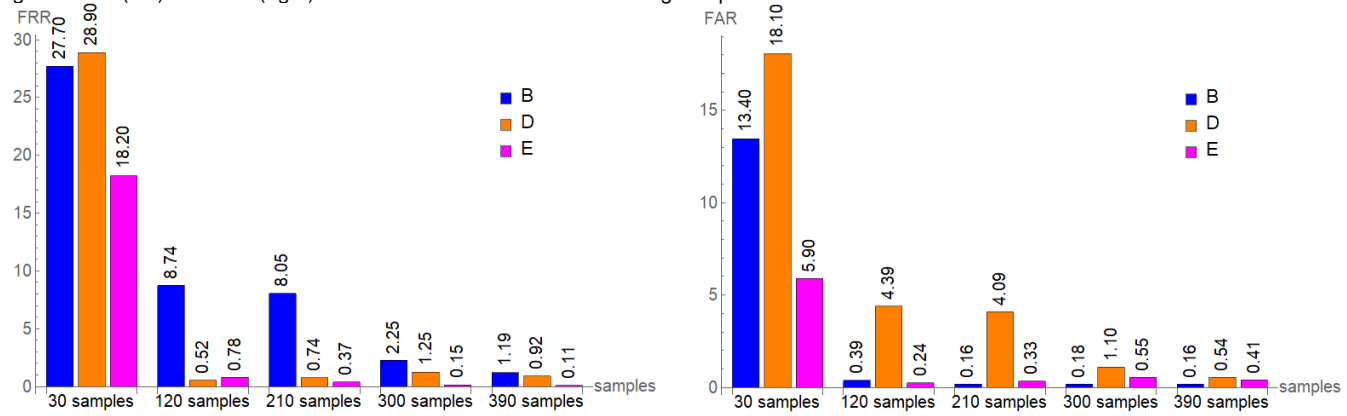
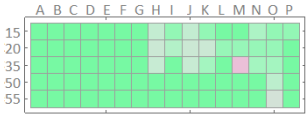
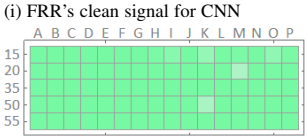


Fig. 26. FRR (left) and FAR (right) with BiLSTM as a function of the training samples number



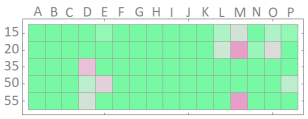
(i) FRR's clean signal for CNN

tr.	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
15	0.	0.	0.	0.	0.	0.	0.	0.0375	0.0125	0.0375	0.0125	0.	0.	0.0250	0.0125	0.0125
20	0.	0.	0.	0.	0.	0.	0.	0.0429	0.0286	0.0429	0.0429	0.0143	0.0143	0.0143	0.0143	0
35	0.	0.	0.	0.	0.	0.	0.	0.0400	0.	0.0400	0.0200	0.	0.0800	0.0200	0.0200	0
50	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.0333	0
55	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.0500	0



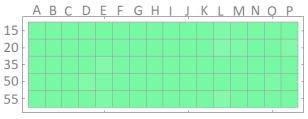
(ii) FAR's clean signal for CNN

tr.	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
15	0.	0.	0.	0.	0.	0.	0.	0.0025	0.0075	0.	0.0133	0.0008	0.0025	0.	0.0108	0.0017
20.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.0248	0.	0.0010	0
35	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.0013	0
50.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.0220	0.	0.	0.	0.	0
55	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0



(iii) FRR's clean signal for BiLSTM

tr.	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
15	0.	0.	0.	0.	0.0125	0.	0.	0.	0.	0.	0.	0.0250	0.0500	0.	0.0250	0.0125
20	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.0428	0.1000	0.0142	0.0571	0.
35	0.	0.	0.	0.0800	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
50	0.	0.	0.	0.0333	0.0666	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.0333
55	0.	0.	0.	0.0500	0.	0.	0.	0.	0.	0.	0.	0.	0.1000	0.	0.	0.



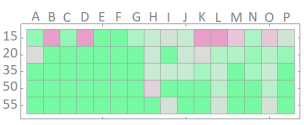
(iv) FAR's clean signal for BiLSTM

tr.	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
15	0.	0.	0.	0.0008	0.	0.	0.	0.	0.	0.	0.	0.0033	0.0016	0.0008	0.0008	0.0008
20	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.0066	0.0028	0.	0.	0.0047
35	0.	0.	0.	0.	0.0053	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
50	0.	0.	0.	0.0044	0.0022	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.0022	0.
55	0.	0.	0.	0.	0.0033	0.	0.	0.	0.	0.	0.	0.0066	0.	0.	0.	0.

Fig. 27. FRRs (up) and FARs (down) as heatmap (left) and numerical values (right) for the CNN and BiLSTM networks (100 samples, 16 speakers)

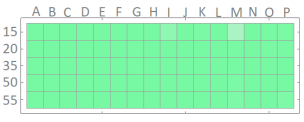
speakers A to G. It is always interesting to see that different neural network architectures behave slightly different. Speakers B, D, E are more easy to distinguish in this dataset which clearly points out that the experimental conditions may influence the results, though in all datasets high accuracy can be obtained. Finally, with both networks by increasing the size of the training set the FAR's and FRR's are generally kept at 0% or close with 2 exceptions in which the FRR gets to 10%. In all of the experiments, the accuracy computed over the entire dataset was 95-100%.

To further improve on usability we considered reducing the number of frequencies used in the sweep signal. According to recent results in the field of psychoacoustics, frequencies in the range of 2kHz-7kHz are considered annoying for human years. Concretely, the work in [51] shows the sounds with high unpleas-



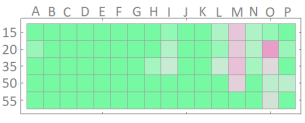
tr.	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
15	0.0125	0.675	0.0125	0.5375	0.	0.	0.0250	0.0375	0.0500	0.0375	0.3125	0.1625	0.0750	0.0250	0.0750	0.0500
20	0.0571	0.	0.	0.	0.	0.	0.	0.0429	0.	0.0429	0.0571	0.0286	0.0286	0.0143	0.0286	0.0143
35	0.	0.	0.	0.	0.	0.	0.	0.0400	0.0400	0.0400	0.0200	0.0400	0.	0.0200	0.0400	0.
50	0.	0.	0.	0.	0.	0.	0.	0.0667	0.	0.	0.	0.0333	0.	0.	0.0333	0.
55	0.	0.	0.	0.	0.	0.	0.	0.	0.0500	0.	0.	0.0500	0.	0.	0.0500	0.

(i) FRR's with 2-7kHz removed for CNN



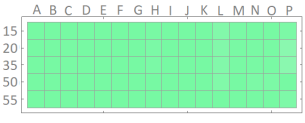
tr.	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
15	0.	0.	0.	0.	0.	0.	0.	0.	0.0100	0.	0.0017	0.	0.0242	0.	0.0008	0.0017
20	0.	0.	0.	0.	0.	0.	0.0010	0.	0.	0.	0.	0.	0.0010	0.	0.0010	0.0010
35	0.	0.	0.	0.	0.	0.	0.0013	0.	0.	0.	0.	0.	0.	0.	0.	0.
50	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.0022	0.	0.	0.
55	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.0033	0.	0.	0.0033

(ii) FAR's with 2-7kHz removed for CNN



tr.	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
15	0.	0.	0.	0.	0.	0.	0.	0.	0.0125	0.	0.	0.0250	0.0750	0.0250	0.0125	0.
20	0.0142	0.	0.	0.	0.	0.	0.	0.	0.0285	0.	0.	0.0142	0.0714	0.0285	0.1142	0.0142
35	0.	0.	0.	0.	0.	0.	0.	0.0200	0.0400	0.	0.	0.0400	0.0800	0.0200	0.0600	0.
50	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.0666	0.	0.0333	0.0333
55	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.0500	0.

(iii) FRR's with 2-7kHz removed for BiLSTM



tr.	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
15	0.	0.	0.	0.	0.	0.	0.	0.0008	0.	0.	0.	0.0050	0.0016	0.	0.	0.0025
20	0.	0.	0.0009	0.	0.	0.	0.	0.0019	0.	0.	0.	0.0047	0.0009	0.0009	0.0009	0.0085
35	0.	0.	0.0009	0.	0.	0.	0.	0.0019	0.	0.	0.	0.0047	0.0009	0.0009	0.0009	0.0085
50	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.0044	0.	0.0022	0.0022	0.
55	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.0033

(iv) FAR's with 2-7kHz removed for BiLSTM

Fig. 28. FRRs (up) and FARs (down) as heatmap (left) and numerical values (right) for the CNN and BiLSTM networks with 2-7kHz band removed

antness, e.g., knife or fork on bottle, chalk on blackboard, to be in the range of 2kHz and up to 7kHz. Also, the authors in [52] point to the range of 2kHz-4kHz as being annoying for human years. For this reason, we choose to cut the frequencies in the range of 2kHz-7kHz and present the results in Figure 28. It can be easily seen that the FARs and FRRs are only slightly affected by the frequency removal and in general it is the same speakers that caused minor confusions.

Besides generating just part of the sweep signal, which will reduce authentication time and potentially cause less hearing discomfort to humans, the sweep signal can be used only to complement traditional authentication methods in order to re-enforce security with physical tokens in special circumstances, e.g., when authorizing a secondary device to pair on Bluetooth with the headunit or to give access to more critical functionalities.

5.3 Results on recordings influenced by noise

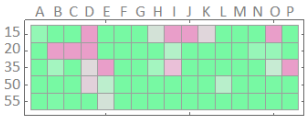
We now carry more tests on the deep neural network classifiers by providing them with sample datasets that are affected by synthetic AWGN. As mentioned in an earlier section, AWGN is not a perfect replacement for environmental noise, but this type of synthetic noise was also used by related works, e.g., [14], when fingerprinting microphones. Clearly, this type of noise will negatively influence the accuracy of the classifiers. Fortunately, as we show below, the effects are only small when AWGN with an equal power to the original signal is used.

Figure 29 presents the results in terms of heatmaps and numerical values when using the 16 speaker dataset (consisting in 100 measurements for each speaker) after applying the synthetic noise to each of the measurements. The performance degradation is visible when comparing to the same values obtained for the clean signal in Figure 27. But both networks still manage to classify the speakers with very high accuracy. The more significant difference when adding noise, is that the BiLSTM appears to struggle a bit with distinguishing between speakers L and M. The CNN also has some problems with L and M, but it improves with the number of samples, while the BiLSTM doesn't. This may be also due to the lower number of layers in the BiLSTM which may lead to a lower representation power. The overall loss in accuracy between the clean signals and the ones affected by noise was in the range of 1-4% as only a small percentage of the speakers caused classifications keeping the overall accuracy always above 95%. For the CNN all the FARs and FRRs are kept below 5% with the larger training sets. The fact that neural network classifiers can recognize speakers with high accuracy even when the recordings are affected by noise seems very promising for practical deployments.

Given the nature of the scenario that we target, i.e., an in-vehicle authentication setup, we also consider to analyze a more challenging scenario: the effect of human speech over the identification of each smartphone. To achieve this, we use the recordings from a phone in the Mobiphone dataset [53] which records the speech of 24 persons from a public database on smartphones. The smartphone recorded voice of the 24 persons in the Mobiphone dataset was evenly distributed over the 100 samples that we collected. The results can be seen in Figure 30. In this case it can be easily seen that the results are affected to a higher extent. This indicates human speech to be more harmful for phone recognition. Still, the FAR is generally well below 2% and the FRR occasionally reaches 20% at 55 samples which will lead to a higher rejection rate for legitimate devices (this is expected because of the noise). Also, the results with the CNN have higher false rejection rates and lower false acceptance rates than for the BiLSTM, but this is explainable as we used distinct separation thresholds in the two implementations, i.e., 0.5 for the CNN and 0.1 for the BiLSTM.

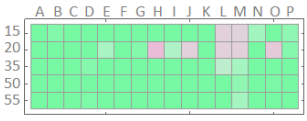
The influence of other environmental factors such as temperature on speakers has been poorly studied so far. There are only a limited number of research works that have considered the influence of temperature on voice coils [54] and nano-materials in speakers [55]. We may consider experiments involving such environmental changes as future work.

To serve as additional evidence for the advantage in using deep learning neural networks, we also add three classical machine learning classifiers KNN, SVMs and Random Forests to serve as reference in our analysis. As can be seen in the barcharts plots with the FAR and FRR in Figures 31 and 32 the problem with



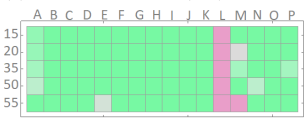
(i) FRR's with AWGN (0db) for CNN

tr.	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
15	0.0125	0.	0.	0.1250	0.	0.	0.	0.0500	0.9900	0.1250	0.0625	0.	0.	0.	0.9900	0.
20	0.	0.8571	0.1000	0.1857	0.	0.	0.	0.	0.0286	0.	0.	0.	0.	0.0143	0.0143	0
35	0.	0.0200	0.	0.0600	0.1200	0.	0.	0.0200	0.0800	0.	0.	0.	0.	0.	0.0400	0.1200
50	0.	0.	0.	0.0667	0.0333	0.	0.	0.	0.	0.	0.	0.0310	0.	0.	0.0010	0.
55	0.	0.0033	0.	0.	0.0500	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.



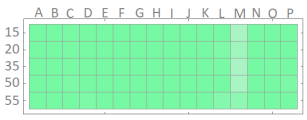
(ii) FAR's with AWGN (0db) for CNN

tr.	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
15	0.	0.0050	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.0667	0.0650	0.0192	0.	0.0100
20	0.	0.	0.	0.	0.0219	0.	0.0057	0.0819	0.0267	0.0667	0.	0.0667	0.0667	0.	0.0714	0.0067
35	0.	0.	0.	0.0067	0.	0.	0.	0.	0.	0.	0.	0.0347	0.0200	0.	0.	0
50	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.0110	0.	0.0020	0
55	0.	0.	0.	0.0033	0.	0.	0.	0.	0.	0.	0.	0.	0.0200	0.	0.	0



(iii) FRR's with AWGN (0db) for BiLSTM

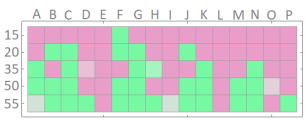
tr.	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
15	0.0125	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.3500	0.	0.	0.	0.
20	0.0142	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.2571	0.0571	0.	0.	0.
35	0.0200	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.3600	0.0200	0.	0.	0.0200
50	0.0333	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.3333	0.	0.0333	0.	0.
55	0.	0.	0.	0.	0.0500	0.	0.	0.	0.	0.	0.	0.1500	0.1000	0.	0.	0.



(iv) FAR's with AWGN (0db) for BiLSTM

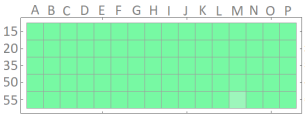
tr.	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
15	0.	0.	0.0008	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.0233	0.	0.	0.
20	0.	0.	0.0009	0.	0.	0.	0.	0.	0.	0.	0.	0.0038	0.0171	0.	0.	0.
35	0.	0.	0.0013	0.	0.	0.	0.	0.	0.	0.	0.	0.0013	0.0240	0.	0.0013	0.
50	0.	0.	0.0022	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.0222	0.	0.	0.0022
55	0.	0.	0.	0.0033	0.	0.	0.	0.	0.	0.	0.	0.0066	0.0100	0.	0.	0.

Fig. 29. FRRs (up) and FARs (down) as heatmap (left) and numerical values (right) for the CNN and BiLSTM networks with AWGN affected signal



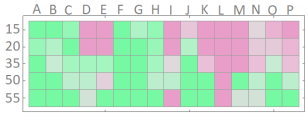
(i) FRR's with speech for CNN

tr.	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
15	1.	1.	1.	1.	1.	0.	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.
20	1.	0.	0.	1.	0.1286	0.	0.	1.	1.	0.	1.	0.9429	1.	1.	1.	1.
35	0.	0.4400	0.	0.0800	1.	0.1000	0.	0.0200	1.	1.	0.	0.9400	0.2600	0.	1.	1.
50	0.	0.	0.	0.1333	0.1667	0.	0.	1.	1.	0.	0.	0.2667	0.	0.	0.0667	0.9000
55	0.0500	0.	0.	0.	0.2000	0.	0.	0.	0.0500	0.	0.	0.2000	0.	0.	0.1000	0.



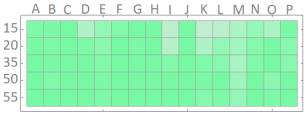
(ii) FAR's with speech for CNN

tr.	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
15	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
20	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
35	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
50	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.0022	0.	0.	0.
55	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.0167	0.	0.	0.



(iii) FRR's with speech for BiLSTM

tr.	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
15	0.	0.	0.0125	0.1875	0.4500	0.	0.0250	0.0125	0.4875	0.0750	0.5375	0.3750	0.5125	0.0625	0.0875	0.3750
20	0.0142	0.0285	0.	0.1000	0.1000	0.	0.	0.0142	0.4000	0.0285	0.4428	0.2000	0.1857	0.0714	0.0714	0.0857
35	0.	0.0400	0.0200	0.0200	0.0200	0.	0.	0.0200	0.0600	0.	0.0800	0.2400	0.1200	0.0800	0.0400	0.0800
50	0.	0.	0.0333	0.0333	0.0666	0.	0.	0.0333	0.0333	0.	0.	0.2666	0.	0.0333	0.	0.0333
55	0.	0.	0.	0.0500	0.	0.	0.	0.	0.1000	0.	0.	0.2000	0.0500	0.0500	0.	0.



(iv) FAR's with speech for BiLSTM

tr.	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
15	0.0033	0.0025	0.	0.0275	0.0125	0.0041	0.	0.0016	0.0333	0.	0.0358	0.0325	0.0233	0.0125	0.0233	0.0008
20	0.0009	0.0038	0.	0.0028	0.0085	0.	0.0019	0.	0.0295	0.	0.0247	0.0123	0.0180	0.0057	0.	0.0076
35	0.	0.0013	0.	0.	0.0040	0.	0.	0.	0.0053	0.	0.0040	0.0080	0.0213	0.0040	0.	0.0066
50	0.	0.0044	0.	0.	0.0022	0.	0.	0.	0.	0.	0.0022	0.	0.0177	0.	0.0088	0.
55	0.	0.	0.	0.	0.0033	0.	0.	0.	0.	0.	0.0066	0.0033	0.0133	0.	0.0033	0.

Fig. 30. FRRs (up) and FARs (down) as heatmap (left) and numerical values (right) for the CNN and BiLSTM networks with speech affected datasets (using MOBIPHONE speech)

these classical classifiers is that they do not enough improvements after increasing the number of training samples from 15% to 55% while the carefully designed deep neural networks did significantly improve once the number of samples has been increased. With a lower number of samples, KNN performed worst in terms of the false acceptance rate, while RF and SVM did somewhat similar to the BiLSTM. The CNN needed a higher number of training samples since for 15% training all speakers were rejected (this is due to the aforementioned 0.5 separation threshold used for the CNN while for the BiLSTM it was set to and 0.1). Notably however, even with 55% training rate all three classical algorithms KNN, SVM and RF have serious problems with speakers F, G (visible as pink bars in Figure 32 under the respective classifier) resulting in 70-80% false rejections. Such problems do not occur with the deep learning alternatives where both the FRRs and FARs drop by increasing the number of training samples. Whether this problem of traditional classifiers can be tackled by careful tuning of various parameters would be out of scope for the current communication and we leave it as future work.

6 Conclusion

We explored an efficient fingerprinting methodology that can be easily implemented to recognize smartphones based on *speaker roll-off characteristics*. Our results show that speaker roll-offs provide a good fingerprint that is also more resilient to changes in volume levels. In contrast, it seems that the volume level may be misleading in case of other approaches. While the slope of the roll-offs alone was sufficient to distinguish between distinct smartphones, for speakers coming from identical smartphone models a more careful analysis with deep-learning algorithms was

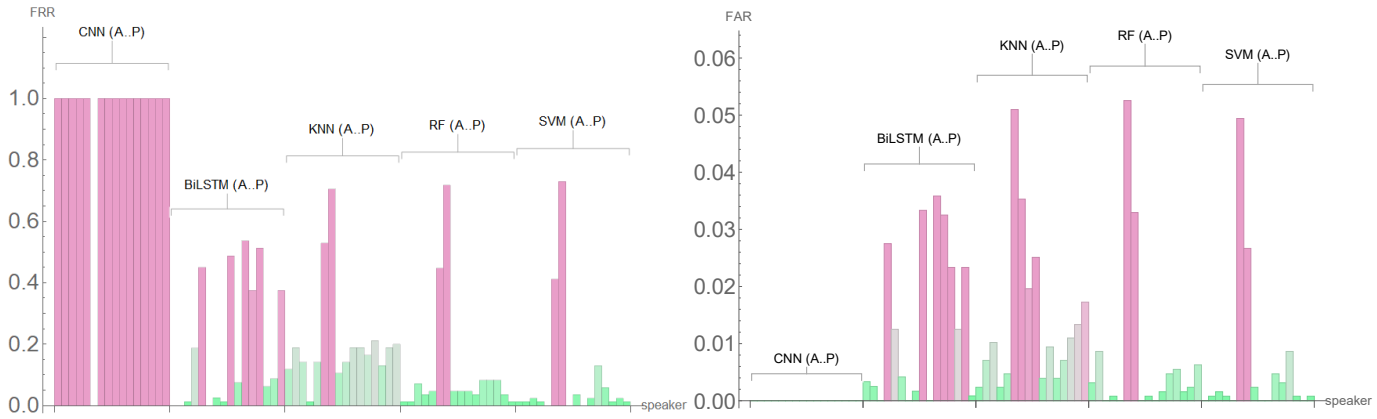


Fig. 31. FRR (left) and FAR (right) with CNN, BiLSTM, KNN, RF and SVM for 15% training

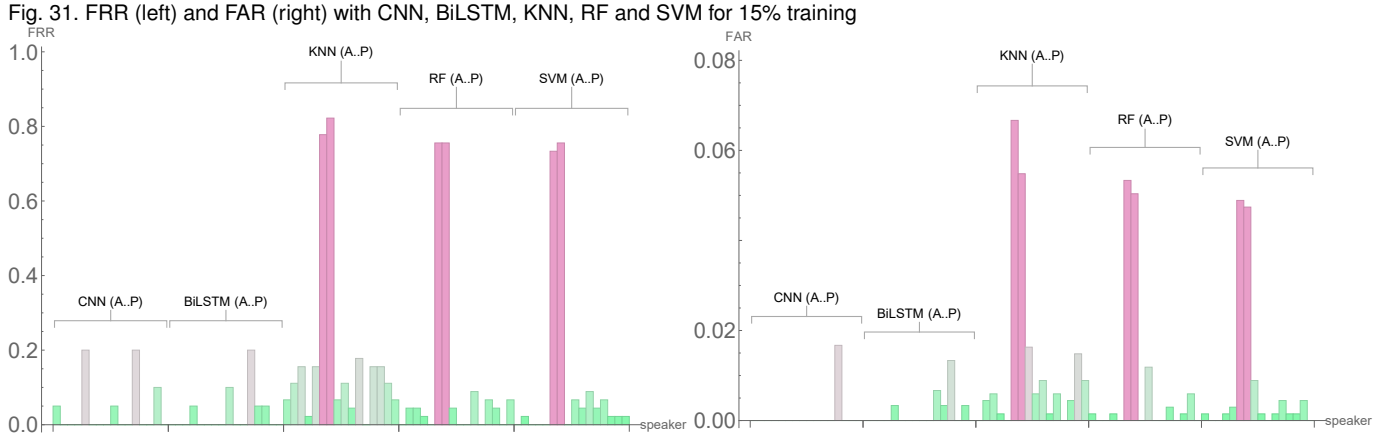


Fig. 32. FRR (left) and FAR (right) with CNN, BiLSTM, KNN, RF and SVM for 55% training

necessary. The CNN and BiLSTM neural network architectures allowed us to distinguish between such identical speakers with an accuracy of 95-100%. The described techniques along with existing methodologies from the literature can be used to fingerprint smartphones and further enable their use as smart keys. One such particular scenario is their use inside vehicles, a reason for which most of the experiments that we carried used an in-vehicle headunit to record the audio output of the smartphones (a calibrated microphone was occasionally used as a reference). The identification has high success rates regardless of the recorder which suggests that in-vehicle headunits are practical for this scenario. Further experiments addressing various in-vehicle settings, e.g., passenger/phone locations, as well as various environmental noises, e.g., speech or traffic sounds, may be future work for us. Clearly, practical deployment in cars calls for more experiments but these are out of reach for us in the current communication.

Acknowledgement. This work was supported by a grant of the Romanian National Authority for Scientific Research and Innovation, CNCS-UEFISCDI, project number PN-III-P1-1.1.-TE-2016-1317 (2018-2020).

REFERENCES

[1] C. Busold, A. Taha, C. Wachsmann, A. Dmitrienko, H. Seudié, M. Sobhani, and A.-R. Sadeghi, "Smart keys for cyber-cars: secure smartphone-based nfc-enabled car immobilizer," in *Proceedings of the third ACM conference on Data and application security and privacy*. ACM, 2013, pp. 233–242.

[2] A. Dmitrienko, A.-R. Sadeghi, S. Tamrakar, and C. Wachsmann, "Smarttokens: Delegable access control with nfc-enabled smartphones," in *International Conference on Trust and Trustworthy Computing*. Springer, 2012, pp. 219–238.

[3] I. Symeonidis, A. Aly, M. A. Mustafa, B. Mennink, S. Dhooghe, and B. Preneel, "Sepcar: A secure and privacy-enhancing protocol for car access provision," in *European Symposium on Research in Computer Security*. Springer, 2017, pp. 475–493.

[4] B. Groza, T. Andreica, A. Berdich, P. Murvay, and E. H. Gurban, "Prestvo: Privacy enabled smartphone based access to vehicle on-board units," *IEEE Access*, vol. 8, pp. 119 105–119 122, 2020.

[5] M. Asim, J. Guajardo, S. S. Kumar, and P. Tuyls, "Physical unclonable functions and their applications to vehicle system security," in *VTC Spring 2009-IEEE 69th Vehicular Technology Conference*. IEEE, 2009, pp. 1–5.

[6] S. Doleva, Ł. Krzywieckib, N. Panwara, and M. Segalc, "Optical puf for non-forwardable vehicle authentication."

[7] J. Yang, Z. Duan, M. Wang, J. Mahmood, Y. Xiao, and Y. Yang, "An authentication mechanism for autonomous vehicle ecu utilizing a novel slice-based puf design," *Journal of New Media*, vol. 2, no. 4, p. 157, 2020.

[8] M. Fomichev, J. Hesse, L. Almon, T. Lippert, J. Han, and M. Hollick, "Fastzip: Faster and more secure zero-interaction pairing," *arXiv preprint arXiv:2106.04907*, 2021.

[9] Z. Zhou, W. Diao, X. Liu, and K. Zhang, "Acoustic fingerprinting revisited: Generate stable device id stealthily with inaudible sound," in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2014, pp. 429–440.

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

[11] ——, "Do you hear what i hear?: Fingerprinting smart devices through embedded acoustic components," in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2014, pp. 441–452.

[12] 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.

[13] G. Baldini and I. Amerini, "Smartphones identification through the built-in microphones with convolutional neural network," *IEEE Access*, vol. 7, pp. 158 685–158 696, 2019.

[14] G. Baldini, I. Amerini, and C. Gentile, "Microphone identification using convolutional neural networks," *IEEE Sensors Letters*, 2019.

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

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

[17] D. Luo, P. Korus, and J. Huang, "Band energy difference for source attribution in audio forensics," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 9, pp. 2179–2189, 2018.

[18] D. Bykhovsky, "Recording device identification by enf harmonics power analysis," *Forensic Science International*, vol. 307, p. 110100, 2020.

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

[20] V. A. Hadoltikar, V. R. Ratnaparkhe, and R. Kumar, "Optimization of mfcc parameters for mobile phone recognition from audio recordings," in *2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA)*. IEEE, 2019, pp. 777–780.

[21] Z. Ba, S. Piao, and K. Ren, "Defending against speaker fingerprinting based device tracking for smartphones," in *2017 IEEE Symposium on Privacy-Aware Computing (PAC)*. IEEE, 2017, pp. 188–189.

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

[23] ——, "Exploring ways to mitigate sensor-based smartphone fingerprinting," *arXiv preprint arXiv:1503.01874*, 2015.

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

[25] J. Zhang, A. R. Beresford, and I. Sheret, "Sensorid: Sensor calibration fingerprinting for smartphones," in *Proceedings of the 40th IEEE Symposium on Security and Privacy (SP)*. IEEE, May 2019.

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

[27] 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 Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1761–1789, 2017.

[28] M. Fomichev, F. Álvarez, D. Steinmetzer, P. Gardner-Stephen, and M. Hollick, "Survey and systematization of secure device pairing," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 1, pp. 517–550, 2017.

[29] D. Chen, N. Zhang, Z. Qin, X. Mao, Z. Qin, X. Shen, and X.-Y. Li, "S2m: A lightweight acoustic fingerprints-based wireless device authentication protocol," *IEEE Internet of Things Journal*, vol. 4, no. 1, pp. 88–100, 2016.

[30] P. Xie, J. Feng, Z. Cao, and J. Wang, "Genewave: Fast authentication and key agreement on commodity mobile devices," *IEEE/ACM Transactions on Networking (TON)*, vol. 26, no. 4, pp. 1688–1700, 2018.

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

[32] E. Novak, Z. Tang, and Q. Li, "Ultrasound proximity networking on smart mobile devices for iot applications," *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 399–409, 2018.

[33] L. Zhang, X. Zhu, and X. Wu, "No more free riders: Sharing wifi secrets with acoustic signals," in *2019 28th International Conference on Computer Communication and Networks (ICCCN)*. IEEE, 2019, pp. 1–8.

[34] B. Zhang, Q. Zhan, S. Chen, M. Li, K. Ren, C. Wang, and D. Ma, "Priwhisper: Enabling keyless secure acoustic communication for smartphones," *IEEE internet of things journal*, vol. 1, no. 1, pp. 33–45, 2014.

[35] M. T. Goodrich, M. Sirivianos, J. Solis, C. Soriente, G. Tsudik, and E. Uzun, "Using audio in secure device pairing," *International Journal of Security and Networks*, vol. 4, no. 1-2, pp. 57–68, 2009.

[36] S. Mathur, R. Miller, A. Varshavsky, W. Trappe, and N. Mandayam, "Proximate: proximity-based secure pairing using ambient wireless signals," in *Proceedings of the 9th international conference on Mobile systems, applications, and services*, 2011, pp. 211–224.

[37] M. Wang, W.-T. Zhu, S. Yan, and Q. Wang, "Soundauth: Secure zero-effort two-factor authentication based on audio signals," in *2018 IEEE Conference on Communications and Network Security (CNS)*. IEEE, 2018, pp. 1–9.

[38] S. Sigg, Y. Ji, N. Nguyen, and A. Huynh, "Adhocpairing: Spontaneous audio based secure device pairing for android mobile devices," in *Proceedings of the 4th International Workshop on Security and Privacy in Spontaneous Interaction and Mobile Phone Use, IWSSI/SPMU*, vol. 12, 2012.

[39] D. Schürmann and S. Sigg, "Secure communication based on ambient audio," *IEEE Transactions on mobile computing*, vol. 12, no. 2, pp. 358–370, 2011.

[40] N. Nguyen, S. Sigg, A. Huynh, and Y. Ji, "Using ambient audio in secure mobile phone communication," in *2012 IEEE International Conference on Pervasive Computing and Communications Workshops*. IEEE, 2012, pp. 431–434.

[41] S. Žiaran, "Low frequency noise and its assessment and evaluation," *Archives of Acoustics*, vol. 38, no. 2, pp. 265–270, 2013.

[42] V. Cevher, R. Chellappa, and J. H. McClellan, "Joint acoustic-video fingerprinting of vehicles, part i," in *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07*, vol. 2. IEEE, 2007, pp. II–745.

[43] J. Yang, S. Sidhom, G. Chandrasekaran, T. Vu, H. Liu, N. Cecan, Y. Chen, M. Gruteser, and R. P. Martin, "Sensing driver phone use with acoustic ranging through car speakers," *IEEE Transactions on Mobile Computing*, vol. 11, no. 9, pp. 1426–1440, 2012.

[44] X. Xu, J. Yu, Y. Chen, Y. Zhu, S. Qian, and M. Li, "Leveraging audio signals for early recognition of inattentive driving with smartphones," *IEEE Transactions on Mobile Computing*, vol. 17, no. 7, pp. 1553–1567, 2017.

[45] J. Han, Y.-H. Lin, A. Perrig, and F. Bai, "Mvsec: Secure and easy-to-use pairing of mobile devices with vehicles (cmu-cylab-14-006)," 2014.

[46] W. Choi, M. Seo, and D. H. Lee, "Sound-proximity: 2-factor authentication against relay attack on passive keyless entry and start system," *Journal of Advanced Transportation*, vol. 2018, 2018.

[47] T.-K. Hon, L. Wang, J. D. Reiss, and A. Cavallaro, "Audio fingerprinting for multi-device self-localization," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 23, no. 10, pp. 1623–1636, 2015.

[48] S. Yun, Y.-C. Chen, H. Zheng, L. Qiu, and W. Mao, "Strata: Fine-grained acoustic-based device-free tracking," in *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services*. ACM, 2017, pp. 15–28.

[49] C. Liu, S. Jiang, S. Zhao, and Z. Guo, "Infrastructure-free indoor pedestrian tracking with smartphone acoustic-based enhancement," *Sensors*, vol. 19, no. 11, p. 2458, 2019.

[50] N. Kim, J. Lee, J. J. Whang, and J. Lee, "Smartgrip: grip sensing system for commodity mobile devices through sound signals," *Personal and Ubiquitous Computing*, pp. 1–12, 2019.

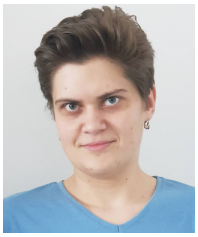
[51] S. Kumar, K. von Kriegstein, K. Friston, and T. D. Griffiths, "Features versus feelings: dissociable representations of the acoustic features and valence of aversive sounds," *Journal of Neuroscience*, vol. 32, no. 41, pp. 14 184–14 192, 2012.

[52] C. Reuter and M. Oehler, "Psychoacoustics of chalkboard squeaking," *The Journal of the Acoustical Society of America*, vol. 130, no. 4, pp. 2545–2545, 2011.

[53] C. Kotropoulos and S. Samaras, "Mobile phone identification using recorded speech signals," in *2014 19th International Conference on Digital Signal Processing*. IEEE, 2014, pp. 586–591.

[54] P. J. Chapman, "Thermal simulation of loudspeakers," in *Audio Engineering Society Convention 104*. Audio Engineering Society, 1998.

[55] P. L. Torraca, M. Bobinger, M. Servadio, P. Pavan, M. Becherer, P. Lugli, and L. Larcher, "On the frequency response of nanostructured thermoacoustic loudspeakers," *Nanomaterials*, vol. 8, no. 10, p. 833, 2018.



Adriana Berdich is a PhD student at Politehnica University of Timisoara (UPT). She received the Engineer title in 2017 and MsC. degree in 2019, both from UPT. Between 2015-2018 she has been working as a software developer in the automotive industry for Continental Corporation in Timisoara. She was a research student in the PRESENCE project focusing on environment-based device association (2019-2020). Currently, she continues as a function developer in the automotive industry for Vitesco Technologies focusing on power-train applications.

Bogdan Groza is Professor at Politehnica University of Timisoara (UPT). He received his Dipl.Ing. and Ph.D. degree from UPT in 2004 and 2008 respectively. In 2016 he successfully defended his habilitation thesis having as core subject the design of cryptographic security for automotive embedded devices and networks. He has been actively involved inside UPT with the development of laboratories by Continental Automotive and Vector Informatik. Besides regular participation in national and international research projects in information security, he lead the CSEAMAN (2015-2017) and PRESENCE (2018-2020) projects, two research programs dedicated to the security of vehicular ecosystems funded by the Romanian National Authority for Scientific Research and Innovation.

René Mayrhofer is head of the Institute of Networks and Security at Johannes Kepler University Linz (JKU), Austria and continues to be involved with Android platform security as a domain expert. Previously, he held a full professorship for Mobile Computing at Upper Austria University of Applied Sciences, Campus Hagenberg, a guest professorship for Mobile Computing at University of Vienna, and a Marie Curie Fellowship at Lancaster University, UK. His research interests include computer security, mobile devices, network communication, and machine learning, which he currently brings together in his research on securing mobile devices and digital identity. Within the scope of u'smile, the Josef Ressel Center for User-friendly Secure Mobile Environments, his research group looked into full-stack security of mobile devices from hardware through firmware up to user interaction aspect. One particular outcome was a prototype for a privacy conscious Austrian mobile Driving License (AmDL) on Android smartphones supported by tamper-resistant hardware. René has contributed to over 80 peer-reviewed publications and is a reviewer for numerous journals and conferences. He received Dipl.-Ing. (MSc) and Dr. techn. (PhD) degrees from Johannes Kepler University Linz, Austria and his Venia Docendi for Applied Computer Science from University of Vienna, Austria.

Efrat Levy is a senior security researcher at Intel Corporation and a Ph.D. student at Ben-Gurion University of the Negev (BGU). She has been actively leading significant security innovative projects in the industry and academia for more than a decade. She holds an M.Sc. degree in the field of Quantum Computing from the Hebrew University of Jerusalem, Israel. Her primary research interests are computer and network security, machine learning, cryptography and side-channel attacks.

Asaf Shabtai is a Professor in the Department of Software and Information Systems Engineering at Ben-Gurion University of the Negev. His main areas of interest are computer and network security, machine learning, security of the IoT and smart mobile devices, and security of avionic and operational technology systems.

Yuval Elovici is the director of the Telekom Innovation Laboratories at Ben-Gurion University of the Negev (BGU), head of BGU Cyber Security Research Center, Professor in the Department of Software and Information Systems Engineering at BGU. He holds B.Sc. and M.Sc. degrees in Computer and Electrical Engineering from BGU and a Ph.D. in Information Systems from Tel-Aviv University. His primary research interests are computer and network security, cyber security, web intelligence, information warfare, social network analysis, and machine learning. Prof. Elovici also consults professionally in the area of cyber security and is the co-founder of Morphisec, startup company that develop innovative cyber-security mechanisms that relate to moving target defense.