Exploiting Contactless Side Channels in Wireless Charging Power Banks for User Privacy Inference via Few-shot Learning

Tao Ni*, Jianfeng Li†, Xiaokuan Zhang‡, Chaoshun Zuo¶, Wubing Wang‖, Weitao Xu*, Xiapu Luo§, Qingchuan Zhao*✉

*City University of Hong Kong, †Xi'an Jiaotong University, ‡George Mason University, ¶The Ohio State University, ‖DBAPPSecurity Co., Ltd, §The Hong Kong Polytechnic University

taoni2-c@my.cityu.edu.hk,jfli.xjtu@gmail.com,xiaokuan@gmu.edu,zuo.118@osu.edu

wubing.wang@dbappsecurity.com.cn,{weitaoxu,qizhao}@cityu.edu.hk,csxluo@comp.polyu.edu.hk

ABSTRACT

Recently, power banks for smartphones have begun to support wireless charging. Although these wireless charging power banks appear to be immune to most reported vulnerabilities in either power banks or wireless charging, we have found a new *contactless* wireless charging side channel in these power banks that leaks user privacy from their wireless charging smartphones without compromising either power banks or victim smartphones. We have proposed BankSnoop to demonstrate the practicality of the newly discovered wireless charging side channel in power banks. Specifically, it leverages the coil whine and magnetic field disturbance emitted by a power bank when wirelessly charging a smartphone and adopts the few-shot learning to recognize the app running on the smartphone and uncover keystrokes. We evaluate the effectiveness of BankSnoop using commodity wireless charging power banks and smartphones, and the results show it achieves over 90% accuracy on average in recognizing app launching and keystrokes. It also presents high adaptability when apply to different smartphone models, power banks, *etc.*, achieving over 85% accuracy with 10-shot learning.

✉The corresponding author.

ACM MobiCom '23, October 2–6, 2023, Madrid, Spain

ACM ISBN 978-1-4503-9990-6/23/10…$15.00

https://doi.org/10.1145/3570361.3613288

CCS CONCEPTS

• **Security and privacy → Mobile and wireless security**; **Side-channel analysis and countermeasures**.

KEYWORDS

Wireless charging power bank, Contactless side channel, Few-shot learning

ACM Reference Format:

Tao Ni*, Jianfeng Li†, Xiaokuan Zhang‡, Chaoshun Zuo¶, Wubing Wang‖,, Weitao Xu*, Xiapu Luo§, Qingchuan Zhao*. 2023. Exploiting Contactless Side Channels in Wireless Charging Power Banks for User Privacy Inference via Few-shot Learning. In *The 29th Annual International Conference on Mobile Computing and Networking (ACM MobiCom '23), October 2–6, 2023, Madrid, Spain.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3570361.3613288

1 INTRODUCTION

Today, power banks have almost become one of the must-carry-on devices for numerous people to charge their smartphones outdoors if the battery is about to die. Accordingly, we have witnessed the tremendous growth of power bank rental stations in various public spaces, *i.e.*, cafes and airports, making their global market exceed a value of 7.1 billion dollars worldwide (North America 54%, Asia Pacific 21%, Europe 10%, *etc.*) by the mid of 2022 [17]. Recently, many newly released power banks have begun to support wireless charging because of its growing popularity, and these power banks mostly follow the Qi [42] wireless charging standard that is widely supported by different smartphone models running different mobile operating systems (*e.g.*, iOS and Android).

While previous studies [12, 21, 24, 45, 47] have reported that either wireless charging or purely cable-based power banks could be leveraged to infer user privacy from the charging smartphones, these studies have not raised sufficient public awareness. This is because it seems plausible for wireless charging power banks to survive those vulnerabilities in daily

cases: *(i)* a wireless charging power bank does not require a USB cable that often connects cable-based power banks to smartphones for charging, where the cable can be used to collect side-channel information, *i.e.*, the charging current [12, 45] to eavesdrop on user privacy. *(ii)* unlike wireless chargers, wireless charging power banks are not connected to the power outlet via the power cable when charging a smartphone, and the power supply is dynamically adjusted to the power bank's battery level. Hence, reported attacks that collect traces (*e.g.*, current and voltage) in the power cable and assume the power supply is stable [21, 24] cannot apply to wireless charging power banks. *(iii)* similar to wireless chargers, heterogeneous wireless charging power banks also rely on well-trained models [12, 21, 24, 47], which are challenging to generalize across different power banks and make the attack cost economically unacceptable in practice.

However, wireless charging power banks are not as robust to privacy leakage as they may appear. In this paper, we report a new *contactless* wireless charging side channel in power banks that can be exploited to infer user privacy (*e.g.*, app usage, keystrokes) from their charging smartphones without compromising both the power bank and the smartphone in any way. This new side channel leverages two physical phenomena that are essentially rooted in the wireless charging process, *i.e.*, the emitted coil whine and the induced ambient magnetic field disturbance. These two physical phenomena are stemmed from the load changes [24] resulting from smartphone activities (*e.g.*, turning on the screen, receiving notifications), and these changes slightly vibrate the internal coil of a wireless charging power bank. An adversary could leverage the two physical phenomena to determine the device type and battery status of the charging devices, and infer users' activities on the charging smartphone from their unique and distinctive patterns.

We have designed and implemented BankSnoop to demonstrate the feasibility of leveraging our reported novel attack surface to launch a contactless, fine-grained, and domain-adaptive attack on wireless charging power banks for the first time. Table 1 summarizes a comparison with five state-of-the-art related works [12, 21, 24, 45, 47] from five metrics, which shows BankSnoop is more stealthy and practical: *(i)* It requires no prior knowledge of the smartphone and power bank, *(ii)* It has no need to compromise the power bank or install malware into the victim's smartphone, and *(iii)* it achieves good transferability across various attack scenarios. Specifically, BankSnoop detects the coil whine and measures the ambient magnetic field disturbance to recognize the content changes displayed on a smartphone's screen and uncover sensitive information through four steps: *First*, it detects the appearance of the coil whine as the indicator to trigger the attack because the coil whine can only be generated when a wireless charging power bank is attached to the smartphone and begins to charge its battery. *Then*, it depends on the power spectrum of the coil whine to recognize the type of power bank and smartphone and then leverages the magnetic field traces to infer their battery levels to specify the attacking conditions since battery levels determine the power consumption that significantly affects the strength and direction of ambient magnetic field disturbance. *Next*, it utilizes the magnetic field disturbance to recognize different user activities resulting from different displaying content on the screen. *Moreover*, it also adopts few-shot learning to quickly adapt pre-trained models to be deployed in new attack scenarios and environments, considering a relatively large number of practical factors in practice

Table 1: Comparison with related attacks from five metrics: (M1) contactless or not; (M2) no need to compromise devices; (M3) no prior knowledge of charging devices; (M4) fine-grained user privacy inference; and (M5) adaptive to various conditions. ✓: true, ✗: false.

Attacks	Attack surface	M1	M2	M3	M4	M5
Cour *et al.* [21]	Current in the power line	✗	✗	✗	✗	✗
Wu *et al.* [47]	Inductive current	✓	✓	✗	✗	✗
EM-Surfing [24]	Inductive voltage	✗	✗	✗	✓	✗
Charger-Surfing [12]	Current in the USB cable	✗	✗	✗	✓	✗
GhostTalk [45]	Current in the USB cable	✗	✗	✗	✓	✗
BankSnoop	Coil whine and magnetic field	✓	✓	✓	✓	✓

We evaluate the effectiveness of BankSnoop with a custom-built portable attacking device, which comprises commercial-off-the-shelf (COTS) electronic components, in uncovering three user privacy information, *i.e.*, app launching, in-app activities, and input keystrokes (*e.g.*, unlocking passcode, keyboard input) from different wireless charging power banks and smartphones configuring with various impact factors (different battery levels, users, screen brightness, *etc.*). Our evaluation shows that BankSnoop achieves high effectiveness in coil whine detection (99.0%), charging device fingerprinting (98.3%), battery level inference of both the power bank and the smartphone (99.8%), app launching recognition (93.1%), and keystroke uncovering from the unlocking numeric keyboard (94.9%) and the full-size QWERTY keyboard (86.9%) within ten attempts. In addition, BankSnoop also presents high performance and resilience when considering different practical impact factors by exploiting the few-shot learning module with 5-shot and 10-shot adaptation. On average, it achieves over 80% and 85% accuracy in 5-shot and 10-shot learning when adapting to different scenarios.

Contributions. We summarize the contributions as follows:

- **A novel side-channel attack.** We introduce a new side channel that can be exploited to attack wireless charging power banks in a contactless manner. It leverages the emitted coil whine and the induced magnetic field disturbance to reveal sensitive information about the smartphone when wirelessly charged by a power bank.

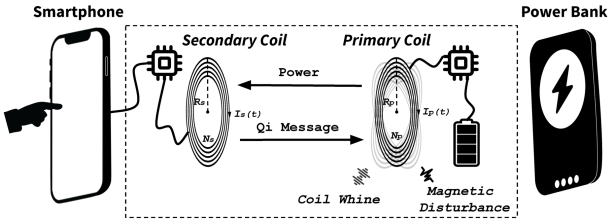


Figure 1: Wireless charging using a power bank.

- **A new attack framework.** We propose and implement a new attack framework, BankSnoop, to demonstrate the feasibility of the new side channel and address limitations in the previous wireless charging side-channel attacks.
- **Extensive evaluation.** We conduct an extensive evaluation to demonstrate the effectiveness of BankSnoop. The results indicate that it achieves high accuracy in uncovering user privacy and shows great potential for domain adaptation in various attack scenarios.

2 BACKGROUND

2.1 Wireless Charging Power Bank

Nowadays, almost all wireless charging power banks are designed to support the Qi wireless charging standard [42]. As shown in Figure 1, once a power bank is attached to a smartphone, it uses electromagnetic induction [46] to transfer power from its coil (*primary coil*) to the coil in the smartphone (*secondary coil*). First, the power bank generates inductive electromagnetic fields $\Phi_p(t)$ and $\Phi_s(t)$ in the primary coil and the secondary coil based on the Biot-Savart law (Equation 1). Then, according to Faraday's law, the inductive electromagnetic field $\Phi_s(t)$ generates an induced voltage $U_s(t)$ to charge the smartphone as shown in Equation 2:

$$\Phi_p(t) = \frac{\mu_0 N_p I_p(t)}{2R_p}, \Phi_s(t) = \eta \Phi_p(t) \tag{1}$$

$$U_s(t) = N_s \frac{\Delta \Phi_s(t)}{\Delta t} = \eta \frac{N_s}{N_p} \cdot \frac{\mu_0 \Delta I_p(t)}{2R_s \Delta t} \tag{2}$$

where $I_p(t)$ is the running current in the primary coil, N_p and R_p are the turns and radius of the primary coil, N_s and R_s are the turns and radius of the secondary coil, η is the energy transmission coefficient, and μ_0 is the magnetic constant.

In the power transfer phase, the control circuit in the power bank continuously communicates with the control unit in the smartphone via Qi message and adjusts the current in the primary coil. That is, when the user is using mobile apps on the charging smartphone, the smartphone will increase the charging speed as the running app consumes more power [42, 47]. Hence, the smartphone will send a Qi message to the power bank to request more power supply, which changes the current running in the primary coil.

2.2 Two Physical Phenomena

In the charging process, the wireless charging power bank controls the running current in the coil $I_p(t)$ based on the level of its battery $B(t)$ (Equation 3) because it contains limited electricity storage and a continuous large discharging current will inevitably shorten the battery life.

$$I_p(t)\ (\text{A}) = \begin{cases} I_{p_1} & B_0 < B(t) \le B_1\ (\text{mAh}) \\ I_{p_2} & B_1 < B(t) \le B_2\ (\text{mAh}) \\ I_{p_3} & B_2 < B(t) \le B_3\ (\text{mAh}) \\ \dots & \dots \end{cases} \tag{3}$$

In particular, the adjusting mechanism of the charging voltage $U_s(t) \propto \Delta I_p(t)$ in the Qi wireless charging protocol [42, 47] and the consequent dynamically changed current ultimately result in slight vibrations of the coils, which incites two physical phenomena, *i.e.*, the coil whine and the disturbance of the ambient electromagnetic field.

Coil whine. Coil whine, *a.k.a.*, electromagnetically induced acoustic noise, is a microphonics phenomenon produced by the coils' vibration under the excitation of electromagnetic forces (*e.g.*, Maxwell stress tensor, magnetostriction, and Lorentz force) [6] based on the Ampere's force Law as:

$$F_p(t) = N_p \Phi_p(t) I_p(t) L_p \propto B^2(t) \tag{4}$$

where L_p is the circumference of the primary coil. These forces cause distortion and vibration of the coil, resulting in different coil whines. Specifically, coil whine exists in different frequency ranges, which makes it either human audible (frequency between 20Hz and 20kHz) or inaudible, while it can be captured by sound-recording microphone modules with sufficient sampling frequency [50].

Magnetic field disturbance. Different smartphone-user interactions cause changes in induced current during the charging process [21], which results in the disturbance of the ambient electromagnetic field. Specifically, power-intensive smartphone activities (*e.g.*, screen animation, pressing keyboard) changes the load by $\Delta r(t)$ on the secondary coil, which further results in the disturbance of the electromagnetic field $\Delta\Phi(t)$ as shown in Equation 5:

$$\Delta I(t) = \frac{U_s(t)}{\Delta r(t)}, \Delta \Phi(t) = \frac{\mu_0 N_s \Delta I(t)}{2R_s} = \frac{\mu_0 N_s U_s(t)}{2R_s \Delta r(t)} \propto \frac{\Delta B(t)}{\Delta r(t)} \tag{5}$$

As such, these changes can be measured by monitoring the electromagnetic field over a period of time. In practice, the electromagnetic field at a specific time point can be described as a 3-D (x, y, z) vector that can be captured by magnetometer modules, which can be utilized for inferring various user activities on smartphones [10, 34, 36].

3 MOTIVATION AND THREAT MODEL

3.1 A Motivating Example

This section presents a motivating example of launching our newly discovered side-channel attack in a real-life scenario. After attaching a wireless charging power bank to a smartphone, a user unlocks the screen with the password (*e.g.*, "1234"), taps the app icon to open the PayPal, and enters the password "abcde" to access financial functions (*e.g.*, pay,

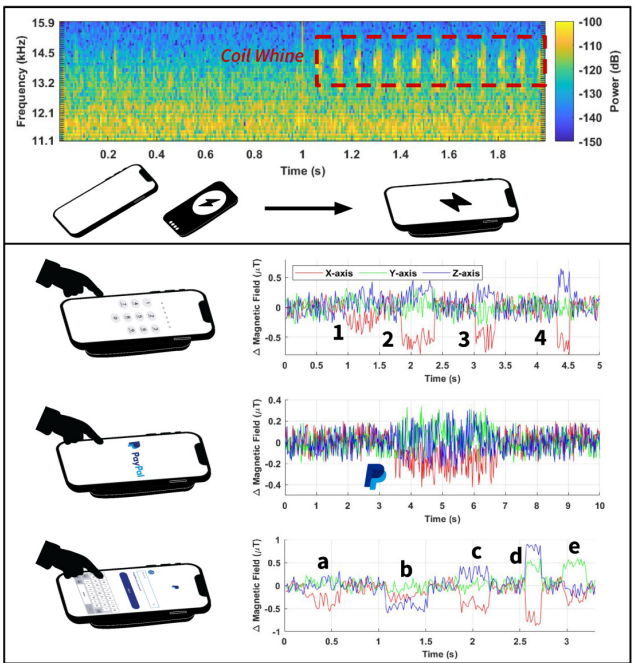


Figure 2: Motivating example scenario: a user charges the smartphone with a wireless charging power bank, unlocks the screen with password "1234", touches the app icon to open PayPal, and types password "abcde" to enter the financial account. Upper part: the power spectrum of the coil whine when the wireless charging process starts. Lower part: user activities and corresponding changes of the magnetic field.

transfer). These actions change the content displayed on the screen from one to another accordingly. As mentioned in §2.2, changes on the screen could impact the current in both the primary coil in the wireless charging power bank and the secondary coil in the smartphone, which further influences the ambient electromagnetic field, and these changes present detectable features that can be used for recognizing corresponding smartphone activities to infer user privacy.

In Figure 2, we present the changes in coil whine and the ambient magnetic field associated with displaying content changes resulting from different user interactions. Specifically, we show the power spectrum of the coil whine within the range from 13kHz to 15kHz and the three-axis magnetic field disturbance of unlocking keystrokes, app launching, and QWERTY keystrokes. As can be seen, the coil whine is detected right after attaching the wireless charging power bank to a smartphone. On the other hand, the magnetic field shows apparent disturbances when launching an app (*i.e.*, PayPal). Moreover, the magnetic field disturbance can also reflect and distinguish different keystrokes on the unlock numeric keyboard (*e.g.*, "1") and the full-size QWERTY keyboard (*e.g.*, "a"). Therefore, the mentioned two physical phenomena, coil whine and the magnetic field disturbance, can be exploited to develop a new contactless wireless charging side-channel attack to reveal the displaying content changes on the screen and uncover sensitive information (*e.g.*, running apps, in-app activities, keystrokes).

3.2 Threat Model

We consider a common scenario of using wireless charging power banks to charge smartphones where a victim attaches the wireless charging power bank (*e.g.*, his/her own or borrowed from shareable rental stations) to the smartphone. Then, the victim places the devices on a table and performs a series of interactions (*e.g.*, typing the keyboard on the screen, running apps). Such a scenario is prevalent in daily life in various public spaces like airports or cafes. The attackers can place an attacking device to record the leakage of physical traces in close proximity to the target power bank.

What an attacker cannot do. Unlike many relevant attacks [12, 21, 24, 45, 47], the attacker does not necessarily have prior knowledge of the charging smartphone and the power bank (*e.g.*, model type, battery status). We do not assume the attacker can compromise the power bank or install malware into the victim's smartphone to acquire current/voltage traces either. Also, the attacker has no LoS view of the two devices and does not know the specific time that the wireless charging power bank begins to charge the smartphone. Moreover, it is unlikely for an attacker to collect large amounts of data samples from different conditions to train multiple privacy inference models before the attack.

What an attacker can do. The attacker can place a small attacking device to record coil whine and measure the ambient magnetic field disturbance in close physical proximity to the target power bank (underneath the table or side-by-side, *e.g.*, 4in or 10cm, within a certain distance). The attacking device could be small to be hidden in a common electronic peripheral (*e.g.*, an earbud case) that can be attached beneath a table or put near the power bank without being notified.

4 ATTACK DESIGN

Figure 3 presents the overview of BankSnoop. An attacker first acquires coil whine and magnetic field signals to determine the charging status and trigger the attack to recognize the types and battery levels of both the smartphone and the power bank. Then, the triggered attacking device utilizes magnetic signals for fine-grained activity recognition using pre-trained deep neural network models. Moreover, BankSnoop also incorporates a few-shot learning module for quickly adapting to various attack scenarios (*e.g.*, different smartphone models, power banks). Finally, the attacker can infer fine-grained user activities and privacy such as unlocking passcode, sensitive keystrokes, and in-app activities.

4.1 Attack Triggering Recognition

To launch this side-channel attack, we need to recognize the triggering condition where a power bank wirelessly charges

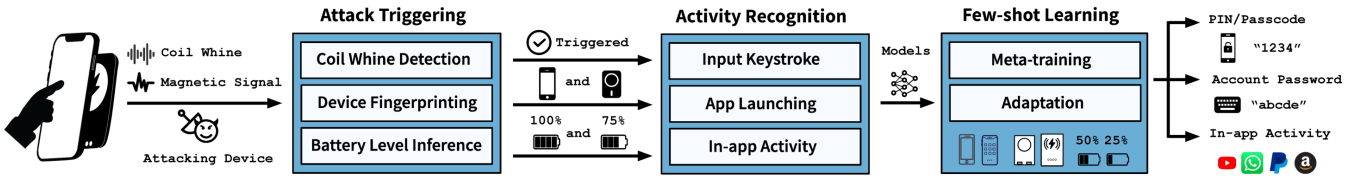


Figure 3: Overview of BankSnoop.

a smartphone. As mentioned above (§2), it will generate two physical phenomena, the coil whine and the magnetic field disturbance at such a condition. Specifically, the coil whine appears when the wireless charging starts and performs as an indicator to trigger the attack. Furthermore, the magnetic field disturbance can be used to infer the status of the charging devices. As such, we use them together to detect the charging status by monitoring the coil whine and infer the battery levels of both the power bank and the charging smartphone by measuring the magnetic field.

Coil whine detection. We find the acoustic phenomenon, coil whine, which exists during the wireless charging process. Therefore, we can use the coil whine effect as the attack trigger of BankSnoop. In practice, the microphone module on the attack prototype first detects the coil whine, and a high-pass filter is applied to remove noises caused by low-frequency sounds (*e.g.*, human speaking, touchscreen tapping). Next, we obtain the power spectrum of the filtered audio by utilizing Short-time Fourier Transform (STFT) using a periodic Hann window. Then, we extract acoustic features Mel-frequency cepstral coefficients (MFCCs) [41] from the power spectrum and a pre-trained Decision Tree [38] classifier determines the charging status at present (*non-charging* or *in-charging*). In practice, we leverage MATLAB Audio Toolbox (version 3.0) to extract MFCC features to train the Decision Tree classifier with 10-fold cross-validation.

Device fingerprinting. Aforementioned wireless charging side-channel attacks [21, 24, 47] usually assume the attacker knows the type of the victim's device, whereas it is impractical and increases the difficulties of launching such an attack in a real-life scenario. Based on Equation 4, we know the electromagnetic forces that cause coil whine, are related to the turns and circumference of the coils. Therefore, it is reasonable to exploit the coil whine to fingerprint different power banks and smartphones since their coils have varied characteristics. Following the same procedure, another pre-trained Decision Tree classifier is implemented to determine the device type of both the power bank and the smartphone.

Battery level inference. Unlike wireless chargers with cables that provide a stable charging voltage, wireless charging power banks contain limited electrical energy in the battery. As mentioned in §2.2, the power bank adjusts the charging voltage based on its battery status to prevent excessive discharge. Therefore, two battery levels are involved in the wireless charging process supported by the power bank, *i.e.*, the battery level of the smartphone and the battery level of the power bank. As such, BankSnoop depends on the inference of the two battery levels for two reasons. *First*, it recognizes the power bank's battery level to understand whether it has power left or not (yes or no) to be sufficient to launch the attack. *Second*, the smartphone's battery level is an essential factor impacting model performance in previous studies of wireless charging side channels such as [21], whose models can only work when the smartphone's battery exceeds 80%. Hence, to enhance BankSnoop with the practicality to launch attacks at any battery levels, we leverage the magnetic field disturbance to infer the exact battery level of the smartphone and the power bank to facilitate procedures in the following steps. Specifically, we leverage the magnetometer to collect three-axis magnetic signals and measure the strength of captured 3D magnetic field $Mag_s(t)$ at a specific time point t as shown in Equation 6:

$$Mag_s(t) = \sqrt{Mag_x^2(t) + Mag_y^2(t) + Mag_z^2(t)}, \tag{6}$$

where Mag_x, Mag_y, Mag_z represent the magnetic field on x, y, z axis, respectively. We obtain the strength differences by deducting the magnetic field when no charging device is presented and then calculate the difference's cumulative distribution function (CDF). Next, we use the CDF values to train a Decision Tree classifier to infer the battery level of the power bank and the charging smartphone.

To demonstrate the feasibility of battery level inference in the charging process, we conduct a preliminary investigation to answer three research questions as follows:

- **RQ1:** Do different power banks present different battery levels in a wireless charging process?
- **RQ2:** Does the initial battery percentage of the smartphone impacts the inductive charging current?
- **RQ3:** Can CDFs of magnetic field strength differences distinguish the battery levels of a power bank?

To answer research questions *RQ1* and *RQ2*, we use a commodity app, *i.e.*, Amperes 4 [26], to record statistics during the charging status of iPhone 13 Pro starting from 10% charging with four wireless charging power banks: EGO MAGPOWER 2 [27], Anker MagGo [2], Apple MagSafe Battery Pack [4], and Belkin BOOSTCHARGE [7]. We present the current curves of these power banks in Figure 4a and notice that Apple MagSafe Battery Pack and Belkin BOOSTCHARGE show stable current during the charging process, whereas EGO MAGPOWER 2 and Anker MagGo present ladder-like

current changes. Additionally, we find the current levels of these two wireless charging power banks correspond to the battery levels that are normally displayed as the number of LED lights on the power banks.

> **Answer to RQ1:** Different power banks present different charging patterns, and some (*e.g.*, EGO MAGPOWER 2, Anker MagGo) present ladder-like battery levels.

Furthermore, we measure the inductive charging current in the *secondary coil* when charging an iPhone 13 Pro with EGO MAGPOWER 2 at different initial battery percentages. Figure 4b shows that the inductive currents present similar ladder-like patterns regardless of the initial battery percentage of the smartphone. Although the initial battery percentage of the smartphone has no impact on the inductive charging current, it still influences the current patterns incurred by smartphone activities, as a prior work [21] has demonstrated. Therefore, in BankSnoop, we design the battery level inference module by recognizing the power bank's battery level and the percentage of the smartphone's battery.

> **Answer to RQ2:** The inductive charging current in the *secondary coil* depends on the battery level of the power bank regardless of the smartphone's initial battery percentage.

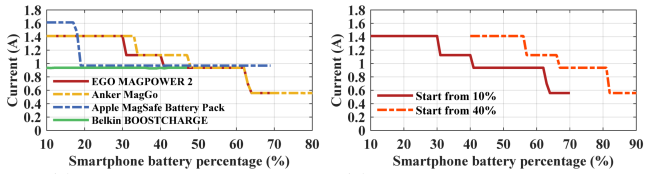
To answer *RQ3*, we measure the battery levels of two wireless charging power banks (*i.e.*, EGO MAGPOWER 2 and Anker MagGo) that demonstrate ladder-like charging current curves by obtaining the cumulative distribution of the strength differences. Figure 5a and Figure 5b separately present the cumulative distribution plots of EGO MAGPOWER 2 and Anker MagGo, they all show that strength difference patterns are distinctive at different battery levels. Therefore, our proposed CDF-based method is feasible to distinguish the different battery levels of a power bank.

> **Answer to RQ3:** We can use CDFs of the magnetic field strength differences as the measurement to distribute different battery levels of a wireless charging power bank.

4.2 Magnetic-based Activity Recognition

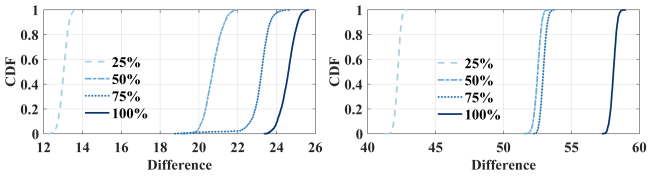
Having recognized the attack triggering condition and determined the devices' type as well as the battery levels, BankSnoop next exploits the captured 3D magnetic field signals with pre-trained deep learning models to recognize various user activities on the charging smartphone.

Pre-processing. After obtaining the raw magnetic field signals, we first apply a Savitzky–Golay (S-G) filter to remove noises in the collected sequential magnetic field signals without distorting the signal shapes [8, 32, 33]. Then, we calculate the average values of the first one-second data as the static magnetic field values on three axes, deduce this offset value



(a) Different power banks. (b) Different battery percentages.

Figure 4: Charging curves measured from iPhone 13 Pro. (a) Charging with different power banks. (b) Charging at different smartphone battery percentages.



(a) EGO MAGPOWER 2. (b) Anker MagGo.

Figure 5: CDFs of magnetic field strength differences at battery levels 25%, 50%, 75%, and 100% of two power banks.

as the starting coordinate, and further obtain the disturbance resulting from different user-smartphone interactions.

Since each activity has a different length of time in every attempt (*e.g.*, an app launching takes 1-5 seconds [36], a single key-press takes 0.05-0.2 seconds [48]), BankSnoop normalizes the processed signals of each activity attempt to the same length of time (*e.g.*, 0.1 seconds) by utilizing up-sampling (*e.g.*, interpolation [35]) or down-sampling (*e.g.*, decimation factor [20]) algorithms.

Activity recognition. As the processed magnetic signals are three-axis one-dimensional time series, we adopt a one-dimensional convolutional neural network (CNN) to build a classifier for activity inference (*e.g.*, app fingerprinting, in-app activity recognition, and single key-press recognition). Specifically, CNN-based deep learning models are utilized in previous side-channel attacks using one-dimensional signals [21, 34] because they can capture temporal (*e.g.*, movements) and spatial (*e.g.*, position) features that reflect user-smartphone interactions from time series and achieve a high classification accuracy [12, 18]. In the CNN-based network, we utilize two convolutional layers to extract temporal and spatial features from the input time series and two batch-normalization layers to standardize the data and stabilize the learning process. Then, two max-pooling layers can reduce the dimension by half, and a dropout layer has been added to prevent overfitting. Finally, the flatten layer converts feature maps to one-dimensional, and the last fully-connected layers output the predicted class with the highest probability.

Implementations. We implement the CNN-based neural networks in Keras 2.3 on the Tensorflow 2.0 framework. We apply the ReLU activation function for two convolution layers and set the pool size as two for each max-pooling layer. In the training stage, we set the batch size as 32 and use the cross-entropy loss and Adam optimizer with an initial learning rate of 0.01 and epoch of 100. The output shape of

the last fully-connected layer depends on the corresponding task (*e.g.*, the number of apps and keys on the keyboard).

In the case of keystroke inference, as users often type passwords or sensitive keystrokes in sequences of various lengths, we define each interval between two adjacent key presses as a new key class and add it to the training process of the mentioned two soft keyboards. Furthermore, the *softmax* function produces an array that contains the probability of each class and outputs the predicted label with the highest probability value (*a.k.a.*, *argmax*). Hence, we use the output of the *softmax* function and generate predicted sequences with top k (*e.g.*, $k = 5, 10, \dots$) highest probabilities, which are also denoted as top-k prediction candidates. We utilize the top-k candidates to evaluate the keystroke inference performance of BankSnoop as it is reasonable that an attacker can surmise the correct passwords or keystrokes in a few attempts [19].

4.3 Adaptation via Few-shot Learning

Although the CNN-based magnetic signal classifier achieves promising accuracy, its performance can be impacted by shifting conditions. Therefore, previous side-channel attacks try to restrict prerequisites (*e.g.*, smartphone battery percentage over 80% [21]) or train multiple deep learning models for different configurations (*e.g.*, smartphone models [12]). However, these methods not only require large-scale datasets to ensure good performance but also limit attack scenarios. Therefore, considering various attack scenarios in practice, we design a few-shot learning module in BankSnoop based on the concept of model-agnostic meta-learning (MAML) [15]. Below, we illustrate our proposed algorithm in two stages: *meta-training* and *adaptation*.

Meta-training. We present the meta-training algorithm for the magnetic signal classifier in Algorithm 1. In the meta-training step, we denote the magnetic signal classifier as f and network parameters as θ. A set of tasks $\mathcal{T}$ are generated from the source dataset $\mathcal{D}_S$ that contains magnetic signal samples collected from various conditions (*e.g.*, different wireless power banks). For each task $\mathcal{T}_i \in \mathcal{T}$, the classifier learns to recognize N classes by using a small number of K (*e.g.*, five or ten) labeled samples of each class, which is also known as K-shot N-way classification task. Furthermore, each task $\mathcal{T}_i$ involves a support set $\mathcal{S}_{\mathcal{T}_i}$ and a query set $\mathcal{S}_{\mathcal{Q}_i}$, where $\mathcal{S}_{\mathcal{T}_i}$ disjoints with $\mathcal{S}_{\mathcal{Q}_i}$ ($\mathcal{S}_{\mathcal{T}_i} \cap \mathcal{S}_{\mathcal{Q}_i} = \phi$) and each set contains $K \times N$ samples. The classifier f is initialized with random parameters θ_0 and then being trained by the associated support set $\mathcal{S}_{\mathcal{T}_i}$ of each task $\mathcal{T}_i$. Then, the classifier learns a new task-specific parameters $\theta'_{\mathcal{T}_i}$ which are tuned from the initial parameters θ_0 via updating the gradient descent:

$$\theta'_{\mathcal{T}_i} = \theta_0 - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_{\theta_0}, \mathcal{S}_{\mathcal{T}_i}), \tag{7}$$

where α is a preset learning rate of individual tasks and $\mathcal{L}_{\mathcal{T}_i}(f_{\theta_0}, \mathcal{S}_{\mathcal{T}_i})$ is the task-specific cross-entropy loss of f on the support set $\mathcal{S}_{\mathcal{T}_i}$ which is given as follows:

Algorithm 1: Meta-training for magnetic classifier

Input: $\mathcal{D}_\mathcal{S}$: source dataset. f: magnetic signal classifier. α and β: learning rate hyperparameters.
Output: f_{θ^*}: trained magnetic signal classifier with optimized parameters θ^*.

```
θ ← θ_0, f_θ ← f_θ0 ▷ random initialize f_θ with parameters θ_0
while not finished do
    T ← generate a batch of tasks from D_S
    for each task T_i ∈ T do
        S_Ti ← K × N support samples from T_i
        S_Qi ← K × N query samples from T_i (S_Ti ∩ S_Qi = ϕ)
        Evaluate ∇_θ L_Ti(f_θ) with S_Ti and loss L_Ti(f_θ, S_Ti)
        θ'_Ti ← θ_0 − α∇_θ L_Ti(f_θ0, S_Ti) ▷ obtain task-specific
          parameters θ'_Ti of T_i using gradient descent.
        Evaluate L_Ti(f_θ'_Ti) with query set S_Qi.
    θ* ← θ_0 − β∇_θ Σ_{T_i∈T} L_Ti(f_θ'_Ti, S_Qi)   ▷ obtain the
      optimized parameters θ* that minimizes all task losses
Output classifier f_θ* with optimized parameters θ*
```

$$\mathcal{L}_{\mathcal{T}_i}(f_{\theta_0}, \mathcal{S}_{\mathcal{T}_i}) = \sum_{(\boldsymbol{x}_j, \boldsymbol{y}_j) \in \mathcal{S}_{\mathcal{T}_i}} \boldsymbol{y}_j \log f_\theta(\boldsymbol{x}_j) + (1 - \boldsymbol{y}_j) \log f_\theta(1 - \boldsymbol{x}_j), \tag{8}$$

where $(\boldsymbol{x}_j, \boldsymbol{y}_j)$ is the jth sample in $\mathcal{S}_{\mathcal{T}_i}$. With the task-specific parameters $\theta'_{\mathcal{T}_i}$ of all tasks $\mathcal{T}_i$ in $\mathcal{T}$, we can define a meta-objective function presented as follows:

$$\underset{\theta}{\operatorname{argmin}} \sum_{\mathcal{T}_i \in \mathcal{T}} \mathcal{L}_{\mathcal{T}_i}(f_{\theta'_{\mathcal{T}_i}}, \mathcal{S}_{\mathcal{Q}_i}). \tag{9}$$

The objection function is proposed to find parameters θ^* that can minimize the sum of task losses in $\mathcal{T}$. We obtain the testing loss of task $\mathcal{T}_i$ by evaluating the performance of the task-specific classifier on the query set $\mathcal{S}_{\mathcal{Q}_i}$. Finally, we obtain θ^* by applying stochastic gradient descent (SGD) [15]:

$$\theta^* \leftarrow \theta_0 - \beta \nabla_\theta \sum_{\mathcal{T}_i \in \mathcal{T}} \mathcal{L}_{\mathcal{T}_i}(f_{\theta'_{\mathcal{T}_i}}, \mathcal{S}_{\mathcal{Q}_i}), \tag{10}$$

where β is another preset learning rate for SGD optimization. The final outputs of meta-training step is the classifier f_{θ^*} with the optimized parameters θ^*.

Adaptation. After obtaining the optimized initialization parameters θ^*, the magnetic signal classifier can realize fast adaptation to various attack scenarios (*e.g.*, different wireless charging power banks, battery levels, *etc.*) with only $K \times N$ labeled training samples collected from the new scenario to fine-tune the pre-trained model. For example, when a new target dataset $\mathcal{D}_{new}$ that is collected from a different wireless power bank ($\mathcal{D}_{new} \cap \mathcal{D}_\mathcal{S} = \phi$), the optimized classifier f_{θ^*} can quickly adapt to this new task $\mathcal{T}_{new}$ and obtain the new parameters θ_{new} in a few gradient descent updates as follows:

$$\theta_{new} = \theta^* - \alpha \nabla_\theta \mathcal{L}_{\mathcal{D}_{new}}(f_{\theta^*}), \tag{11}$$

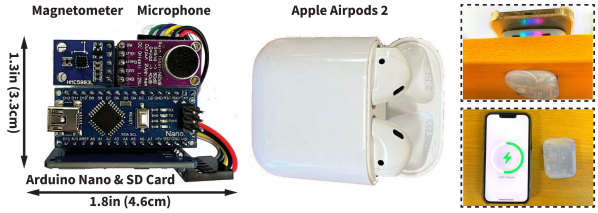


Figure 6: Custom-built attacking device (almost the same size as Apple AirPods 2) and attack scenarios.

where the α is same to the hyperparameter denoted in Equation 7. After the adaptation stage, we obtain the magnetic signal classifier $f_{\theta_{new}}$ with fine-tuned parameters θ_{new} towards the new task $\mathcal{T}_{new}$. In practice, we adopt 5-shot and 10-shot with $N = 120$ for app fingerprinting, $N = 11$ and $N = 33$ for the unlocking and the QWERTY key-pressing recognition, respectively. We set the hyperparameter learning rate α and β as 0.01 and 0.001, respectively. Then, we apply ten gradient descent updates for generating the magnetic signal classifier with cross-task optimized parameters θ^* and ten gradient steps to fine-tune the θ^* to obtain parameters θ_{new} for target datasets in new scenarios.

4.4 Portable Attacking Device

As mentioned above, we have implemented a portable attacking device, which consists of four commercial-off-the-shelf (*COTS*) components: an Arduino Nano microcontroller unit (MCU) [31], a microphone module to capture coil whine, a three-axis magnetometer module to capture magnetic signals, and a micro SD card shield to record collected data. Specifically, we use Adafruit MAX9814 microphone amplifier [1] and HMC5883L magnetometer module [14]. The total size of the attack prototype is approximately 1.8×1.3in (4.6×3.3cm), which is close to the size of an Apple AirPods case. The total cost is approximately 32.5 dollars.

5 EVALUATION

5.1 Experiment Setup

In the primary settings of the experiment[1], we use a full-battery (100%) EGO MAGPOWER2 power bank to charge an iPhone 13 Pro at 80% battery percentage. Then we place the in-charging smartphone on an oak table with a thickness of 0.94in (2.4cm), and stick the attack prototype underneath the table. In addition, the preset sampling frequencies of the microphone and magnetometer are 40kHz and 100Hz, respectively. Moreover, all data processing is conducted on a desktop running Windows 10 with 32GB memory, Intel i7-9700K CPU, and an NVIDIA GeForce RTX 2080Ti GPU. Note that our experiments are conducted in an uncontrolled environment, and low-frequency noises (*e.g.*, human speaking: 50–300Hz; button pressing: 1–10Hz) have little impact since the coil whine has a high-frequency range (*e.g.*, 13–15kHz).

[1]**Ethical consideration**: This work takes ethical considerations seriously, and it has been approved by our IRB to collect data from human participants. More experimental details (*e.g.*, full list of testing sequences) are available at https://github.com/taoni0718/BankSnoop

5.2 Datasets

We build six different datasets on commodity devices in different conditions to demonstrate its effectiveness in §5.3 from four commodity wireless charging power banks (P_1–P_4: EGO MAGPOWER 2, Anker MagGo, Apple MagSafe Battery Pack, and Belkin BOOSTCHARGE) and four smartphones (S_1–S_4: iPhone 13 Pro, iPhone 12, iPhone 11, and Samsung S10) to train different models in BankSnoop and evaluate their performance in detecting coil whine and devices' type, inferring the battery levels, recognizing app/in-app activities, uncovering keystrokes, and adapting to different attack scenarios with the few-shot learning module.

- $\mathcal{D}_{CW}$: the coil whine dataset contains one-second audio clips in two cases: the smartphone is in-charging and non-charging. In the in-charging condition, we collect samples in both screen-off and screen-on status. This procedure is repeated 50 times and then a 0.1 seconds sliding window is applied to perform STFT ($2 \times 50 \times 10 \times 4$ traces).
- $\mathcal{D}_{DF}$: the device fingerprinting dataset follows the same data collection procedure from four smartphones that are being charged by four power banks ($4 \times 4 \times 50 \times 10$ traces).
- $\mathcal{D}_{BL}$: the battery level dataset is collected from two wireless charging power banks (more details in §5.3) that present different battery levels (25%, 50%, 75%, and 100%) when charging the four smartphones at four battery percentages (20%, 40%, 60%, and 80%), each charging combination 500 samples ($4 \times 4 \times 500 \times 8$ traces).
- $\mathcal{D}_{App}$: the mobile app dataset is collected from a total of 120 apps from the official iOS and Android store, which contains the top 5 popular free apps from 24 app categories based on the statistics provided by *appfigures* [3] by the end of 2021. We collect the first 0.1 seconds of launching each app and repeat it 100 times ($120 \times 100 \times 8$ traces). Moreover, we select the most popular five apps (*e.g.*, YouTube, PayPal) and collect data when performing five application-specific activities for 100 times to train the classifier for in-app activity recognition ($5 \times 5 \times 100$ traces).
- $\mathcal{D}_{UK}$ and $\mathcal{D}_{QWERTY}$: the two keystroke datasets are collected from two common soft keyboards: unlocking keyboard ($\mathcal{D}_{UK}$) and full-size QWERTY keyboard ($\mathcal{D}_{QWERTY}$). Each key (including backspace, space, *etc.*), as well as the interval key for segmentation, is pressed 100 times ($11 \times 100 \times 8$ traces in $\mathcal{D}_{UK}$ and $33 \times 100 \times 8$ traces in $\mathcal{D}_{QWERTY}$).

5.3 Effectiveness

We use accuracy and confusion matrix as the metrics to evaluate BankSnoop in coil whine detection, device fingerprinting, battery level inference, app launching/in-app activity recognition, and keystroke uncovering.

Effectiveness of coil whine detection. Based on $\mathcal{D}_{CW}$, we train a Decision Tree classifier to determine the presence of a power bank wireless charging a smartphone. On the testing set (200 samples for each wireless power bank), the Decision Tree classifier achieves an overall accuracy of 99% (EGO MAGPOWER 2: 99.5%, Anker MagGo: 98.5%, Apple MagSafe Battery Pack: 99%, and Belkin BOOSTCHARGE: 100%).

Effectiveness of device fingerprinting. Similarly, we use the captured coil whine in the dataset $\mathcal{D}_{DF}$ to train a Decision Tree classifier to recognize the type of smartphone and the power bank. Figure 7 presents the confusion matrix of the device fingerprinting results of the 16 evaluated combinations (*e.g.*, $S_1 \times P_1$: iPhone 13 Pro charged by EGO MAGPOWER 2, $S_2 \times P_3$: iPhone 12 charged by Apple MagSafe Battery Pack). The results show that BankSnoop achieves an accuracy of 98.3% in recognizing the type of charging devices.

Effectiveness of battery level inference. Based on the results of the preliminary study (§4.1), we, therefore, utilize the MATLAB Statistics Toolbox to generate the CDFs of magnetic field strength differences from $\mathcal{D}_{BL}$ to develop a Decision Tree classifier to recognize the combination of both the battery level/percentage of the power bank and the in-charging smartphone. Figure 8 presents the confusion matrix of battery level inference results of using the EGO MAGPOWER 2 to charge the iPhone 13 Pro at 16 different battery level combinations (*e.g.*, $S20 \times P100$: smartphone battery at 20%, power bank battery at 100%). It shows that BankSnoop achieves 99.8% accuracy in battery level inference.

Effectiveness of app launching recognition. Figure 9 presents the effectiveness of BankSnoop in recognizing 120 mobile apps at the app launching stages. We utilize 80% data of each app from $\mathcal{D}_{App}$ to train the recognition model and evaluate its performance with the remaining 20% data. Overall, the recognition model achieves 93.1 ± 2.9% accuracy on the testing set of traces from 120 apps. Specifically, BankSnoop performs the best in identifying apps in categories such as "Books" and "Education" (accuracy 100.0%), and it performs worst in the category "Social Network" (accuracy 89.8 ± 2.2%). We found that the launching stage of most apps in the category "Social Network" involves fewer animations, which makes them more difficult to be recognized compared with apps that perform launching animations from other categories (*e.g.*, B&N NOOK, Duolingo).

Effectiveness of in-app activity recognition. In $\mathcal{D}_{App}$, we also collect magnetic traces when performing five application-specific activities in five popular apps (YouTube, PayPal, WhatsApp, Facebook, and Spotify) and implement the CNN-based classification model for in-app activity recognition. For example, in YouTube, we evaluate BankSnoop in recognizing activities including *play video*, *forward*, *backward*, *pause video*, and *next video*. Figure 10a–Figure 10e present the confusion matrices of the in-app activity recognition results, where we find BankSnoop achieves an accuracy of 85.2%, 84.0%, 86.2%, 82.2%, and 81.4% in recognizing the five application-specific activities of the evaluated five mobile apps, respectively. Therefore, we demonstrate the effectiveness of BankSnoop, which accurately recognizes not only the launching app but also the fine-grained in-app activities.

Effectiveness of keystroke uncovering. The evaluation of BankSnoop's performance on keystroke uncovering is conducted on two datasets,*i.e.*, $\mathcal{D}_{UK}$ and $\mathcal{D}_{QWERTY}$, that are collected from the unlocking keyboard and the QWERTY keyboard. We randomly generate three sequences of numbers and characteristics for each keyboard with lengths ranging from one to ten[1]. Each randomly generated sequence is repeated for 100 times (*e.g.*, three examples of testing cases in length one of the QWERTY keyboard are "c", "o", and "e"). Figure 11 shows the evaluation results on the random sequence testing set. We generate the top-10 candidates of the predicted sequence and obtain the corresponding accuracy if one of the top-10 candidates is correct. The overall accuracy of uncovering sequences in the length of one and ten are 94.9%, and 86.9%, respectively. The top-10 accuracy decreases as the sequence length increases, whereas the keystroke uncovering accuracy is still comparable to other works [19, 22, 48].

5.4 Few-shot Learning Evaluation

Baselines. We compare our proposed few-shot learning module with three baselines: *(i)* Source-only (*SO*): we use the model trained from only the source dataset $\mathcal{D}_{\mathcal{S}}$ and evaluate its performance on the target dataset $\mathcal{D}_{\mathcal{T}}$ directly with no adaptation, *(ii)* Target-only (*TO*): we use the few samples (*e.g.*, five or ten) from $\mathcal{D}_{\mathcal{T}}$ to train the CNN-based neural network and evaluate it with the rest samples of $\mathcal{D}_{\mathcal{T}}$, and *(iii)* Transfer-convolutional (*TrC*): Transfer convolutional [40] is one of the most state-of-the-art transfer learning methods for domain adaptation, which assumes that the upper layers' representations of similar problems are transferable [30, 44]. In practice, we freeze the convolutional layers of the trained model based on $\mathcal{D}_{\mathcal{S}}$ and then fine-tune the fully-connected layers only with the few-shot samples from $\mathcal{D}_{\mathcal{T}}$. Below, we evaluate the adaptation of BankSnoop in eight different scenarios.

Scenario 1: Different battery levels of a power bank. To evaluate the performance of BankSnoop in different battery levels of the power bank, we collect datasets $\mathcal{D}_{UK}$, $\mathcal{D}_{QWERTY}$ and $\mathcal{D}_{App}$ at the battery level 25%, 50%, 75%, and 100%, and then use datasets collected at the battery level of 100% as the $\mathcal{D}_{\mathcal{S}}$ to build the base model and use the rest samples as $\mathcal{D}_{\mathcal{T}}$. Figure 12a presents the performance of the few-shot learning module at the battery levels of 25%, 50%, and 75%. We can observe that the performance of app launching and keystroke recognition under *SO* is the worst (lower than 25%). The

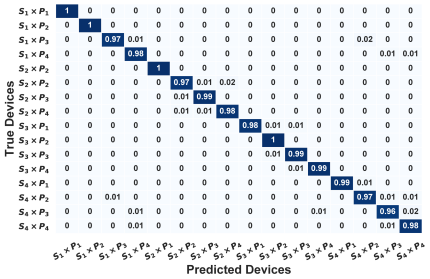


Figure 7: Device fingerprinting results. $S_i \times P_j$: **Smartphone** S_i **charged by the power bank** P_j $(i, j = 1, 2, 3, 4)$.

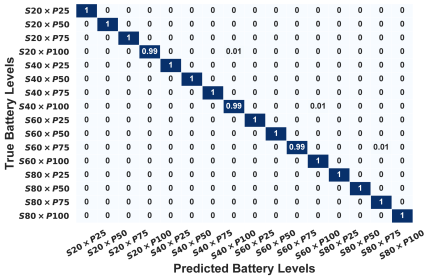


Figure 8: Battery level inference results. $S_{l_1} \times P_{l_2}$: **Smartphone battery at** l_1%, **power bank battery at** l_2% $(l_1, l_2 = 25, 50, 75, 100)$.

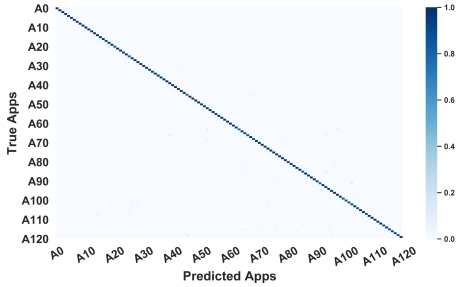


Figure 9: App launching recognition results. A_n: **the** n**th app of the most popular** 120 **mobile apps** $(n = 1, 2, \ldots, 120)$.

(a) YouTube (b) PayPal (c) WhatsApp (d) Facebook (e) Spotify

Figure 10: In-app activity recognition results.Evaluated activities of and : ▶–Play, ❚❚–Pause, ▶▶–Forward, ◀◀–Backward, ▶❙–Next; : –Scan QR code, $ –Pay bills, –Request money, –Get invoices, –Call wallet; : T –Texting, –Send images, –Send videos, –Video call, –Voice messages; : –Thumb-up, –Comment, –Refresh, –Share, –Repost.

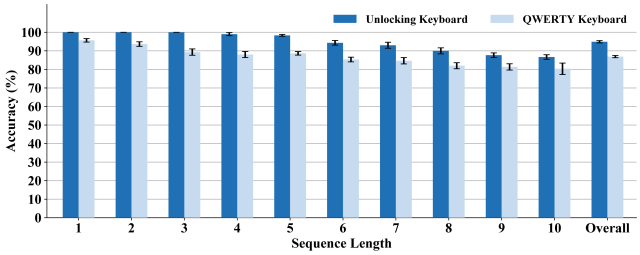


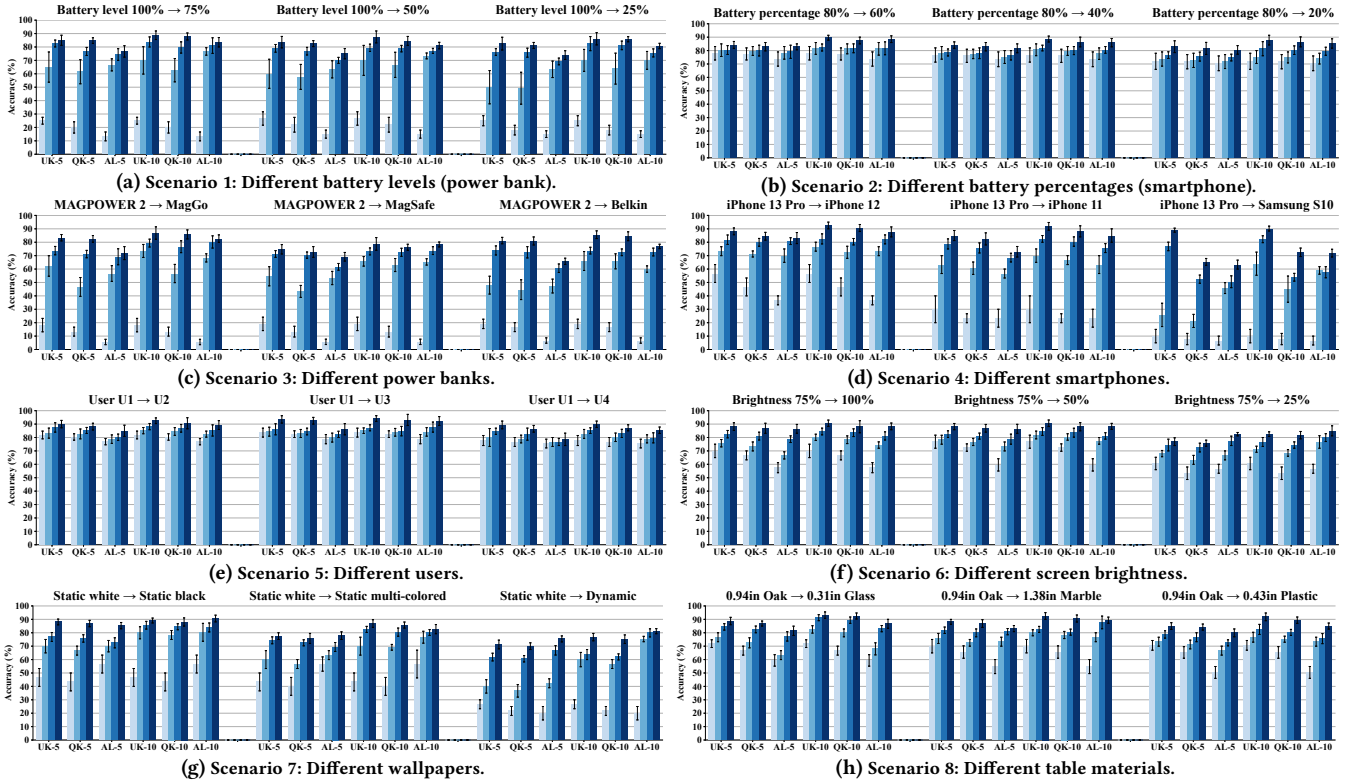
Figure 11: Keystroke uncovering results of the unlocking and the QWERTY keyboards with top-10 **candidates.**

target-only (*TO*) and transfer-convolutional (*TrC*) methods individually improve accuracy by approximately 35% and 45%, but their performance is lower than 80% in most of the cases. Among all scenarios, our proposed few-shot learning approach achieves the best performance. Specifically, it improves the recognition accuracy of app launching, unlocking and QWERTY key pressing to 75.2%, 83.7% and 82.8% in the 5-shot cases, and 81.8%, 87.5%, and 86.1% in the 10-shot cases. The results demonstrate that *BankSnoop can quickly adapt to different battery levels of the wireless charging power bank while maintaining a high accuracy with few samples.*

Scenario 2: Different battery percentages of a smartphone. Previous wireless charging side-channel attacks [12, 21] show that the smartphone battery percentage can impact the model performance. We evaluate the adaptation ability of BankSnoop across different smartphone battery percentages by using the datasets collect from the iPhone 13 Pro charged by the EGO MAGPOWER 2 at battery percentage 80% as $\mathcal{D}_S$ and datasets collected from 60%, 40%, and 20% as $\mathcal{D}_T$. Figure 12b presents the evaluation results, where we find the overall activity recognition accuracy decreases to 76.4%, 75.4% and 71.4% when applying models trained from smartphone battery percentage 80% to 60%, 40%, and 20%. Our few-shot learning module improves the accuracy to 83.3%, 82.9%, and 81.7% in the 5-shot cases, and 88.5%, 87.0%, and 86.5% in the 10-shot cases, which also performs better than the three baselines. The results demonstrate the *practicality of deploying BankSnoop to launch attacks at different smartphone battery percentages with few-shot learning.*

Scenario 3: Different wireless charging power banks. Different wireless charging power banks may present dissimilar coil whine and magnetic signal patterns of a similar task due to the different coil parameters (*e.g.*, coil turns, materials). We evaluate BankSnoop's domain adaptation between different power banks by utilizing datasets of P_1 as $\mathcal{D}_S$ and datasets collected from the other three commodity power banks (P_2–P_4) as $\mathcal{D}_T$. Figure 12c shows the evaluation results of the few-shot learning module with the three power banks. The recognition accuracy of app launching, unlocking passcode and the QWERTY keystroke has been enhanced from lower than 20% (*SO*) to 68.9%, 79.8%, and 78.7% in the 5-shot cases, and 79.3%, 83.7%, and 82.2% in the 10-shot cases, which outperforms about 25% and 8% than the *TO* and *TrC* methods. The results indicate that *it is practical for BankSnoop to attack different power banks and achieve promising accuracy.*

Scenario 4: Different smartphone models. Different smartphones have different configurations of the secondary coil parameters and battery volumes (*e.g.*, iPhone 12: 2815mAh, iPhone 13 Pro: 3095mAh), which results in different patterns of the induced current changes by user activities. Therefore, prior attacks on smartphones [12, 21, 24, 47] usually trained multiple deep learning models to ensure model performance across different smartphones. Instead, we implement the



(a) Scenario 1: Different battery levels (power bank).
(b) Scenario 2: Different battery percentages (smartphone).
(c) Scenario 3: Different power banks.
(d) Scenario 4: Different smartphones.
(e) Scenario 5: Different users.
(f) Scenario 6: Different screen brightness.
(g) Scenario 7: Different wallpapers.
(h) Scenario 8: Different table materials.

Figure 12: Few-shot learning module evaluation results (5-shot and 10-shot) in different scenarios. UK: unlocking keyboard keystroke accuracy. QK: QWERTY keyboard keystroke accuracy. AL: app launching recognition accuracy. -K: with K-shot learning, *e.g.*, UK-5 means 5-shot accuracy of unlocking keystroke recognition. – SO, – TO, – TrC, – Our method.

proposed few-shot learning method by selecting samples collected from S_1 as the $\mathcal{D}_S$ and other datasets collected from another three smartphones (S_2–S_4) as $\mathcal{D}_T$ (charging by EGO MAGPOWER 2). Figure 12d shows the results of adaptation from iPhone 13 Pro (S_1) to other two iPhone models (S_2, S_3), and our few-shot learning method improves the recognition accuracy of app launching, and keystrokes of two keyboards to 78.0%, 86.6%, and 83.5% in the 5-shot cases, and 86.4%, 92.2%, and 89.4% in the 10-shot cases. In particular, the accuracy decreases drastically when we directly apply the model (*SO*) trained for iPhone 13 Pro (S_1) to a Samsung S10 (S_4), which has totally different layouts of soft keyboards. Nevertheless, our method also achieves an accuracy of 72.5% in app launching recognition, 90.2%, and 71.88% in unlocking and QWERTY keyboards' keystrokes recognition. The results demonstrate that *BankSnoop achieves fast adaptation to smartphones of different platforms (iOS and Android).*

Scenario 5: Different users. Since smartphone users may have distinctive typing patterns (*e.g.*, speed and movement), we recruit a total of four volunteers (note as U_1, U_2, U_3, and U_4) to join this study (IRB approved) and collect data for evaluation (iPhone 13 Pro charging with EGO MAGPOWER 2) to investigate the impact of different users. Then, we use the dataset U_1 as the $\mathcal{D}_S$ and other three datasets (U_2, U_3, and U_4) as $\mathcal{D}_T$. Figure 12e shows the results of cross-user evaluations with different approaches. We find the overall activity recognition accuracy decreases to 79.7%, 81.6%, and 76.6% when applying the trained models of $U1$ to $U2$, $U3$, and $U4$. The few-shot learning approach improves the accuracy of app launching, unlocking, and QWERTY keystrokes recognition to 83.3%, 91.0%, and 88.9% in the 5-shot cases, and 89.0%, 92.2%, and 90.3% in the 10-shot cases. The results show that *the few-shot learning method improves the models' domain adaptation performance in cross-user evaluations.*

Scenario 6: Different screen brightness. Recent studies have revealed that the brightness of the touchscreen dominates most of the battery consumption [9, 13]. Hence, the brightness might impact the performance of recognizing user activities, especially when the screen brightness varies greatly (*e.g.*, 75% → 25%). We collected data from four different brightness (25%, 50%, 75% and 100%) when an iPhone 13 Pro is charging by the EGO MAGPOWER 2. Then we set the dataset of brightness 75% as the $\mathcal{D}_S$ and other three brightness datasets as the $\mathcal{D}_T$. Figure 12f presents the evaluations of few-shot learning performance in different screen brightness. The accuracy of app launching, unlocking, and QWERTY keystrokes recognition can be enhanced to 83.8%, 84.0%, and 82.5% in the 5-shot cases, and 86.9%, 87.5%, and

85.8% in the 10-shot cases. The results show that *BankSnoop can quickly adapt to varying brightness conditions.*

Scenario 7: Different wallpapers. Commodity mobile devices' screens are typically buttons with blurred backgrounds. The background picture (*a.k.a.* wallpaper) is displayed by numerous RGB pixels on the OLED touch screen that can exhibit different colors, which induce different power consumption. We consider reducing the wallpapers' impact on BankSnoop while using fewer data samples and we also explore not only static but also dynamic wallpapers. In practice, we use the data collected from pure white wallpapers as $\mathcal{D}_{\mathcal{S}}$ and other datasets collected from pure black, multi-colored, and dynamic wallpapers as $\mathcal{D}_{\mathcal{T}}$ ($P1 \times S_1$). Figure 12g shows the results of adapting the trained model of static white wallpapers to static black and multi-colored wallpapers, where the recognition accuracy rates are 81.6%, 82.8%, and 81.3% in the 5-shot cases, and 86.7%, 88.0%, 86.9% in the 10-shot cases. Regarding the dynamic wallpaper cases, the dynamic animation adds extra noise to the power consumption of the touchscreen, which results in the worst adaptation performance (*SO* lower than 30%). In addition, our method still achieves an average 77.7% accuracy in the dynamic wallpaper cases, which makes BankSnoop *adaptive to the vary of different wallpapers, which can realize higher performance with better screen-noise cancellation methods or more shots.*

Scenario 8: Different table surfaces. In our experiment settings, the properties of the table surface (*e.g.*, thickness, materials) may impact the performance of BankSnoop. Hence, we collect data by placing devices ($P_1 \times S_1$) on three other table surfaces for evaluation: 0.31in (0.8cm) glass, 1.38in (3.5cm) marble, and 0.43in (1.1cm) plastic. Similarly, we use data collected from the oak table as $\mathcal{D}_{\mathcal{S}}$ and evaluate the adaptation performance on datasets of other table surfaces ($\mathcal{D}_{\mathcal{T}}$). Figure 12h shows the evaluation results in three different table surfaces. Activity recognition accuracy, such as unlocking key-press decreases around 25% due to the attenuation of the inductive electromagnetic field. By utilizing few-shot learning, the accuracy of the aforementioned three activities reaches 81.5%, 87.3%, and 86.0% in 5-shot cases, and 87.0%, 92.5%, 90.8% in 10-shot cases. The results show that *table surface matters in such a contactless attack, whereas the proposed few-shot learning method still performs well.*

6 DISCUSSION

6.1 Analysis of Other Impact Factors

Impact of environmental noise. To investigate the impact of environmental noise, we further collect samples (iPhone 13 Pro charged by EGO MAGPOWER 2) with high-frequency environmental noise (*e.g.*, Gaussian white noise [47]) at different signal-to-ratio (SNR) levels. Figure 13 presents the results of coil whine detection and device fingerprinting under noise SNR ranging from 10^{-6} to 50, where we find BankSnoop's performance decreases as we enhance the strength of the environmental noise. In particular, when the strength of the environmental noise is over $10^4\times$ of the coil whine (SNR = 10^{-4}), the performance of BankSnoop degraded drastically (*i.e.*, device fingerprinting accuracy $<$ 40%) as such high-frequency noise dominates the captured signals.

Impact of position and distance. In practice, an attacker can place the disguised attacking device near the victim's charging smartphone at different distances. To understand the impact of the position and distance, we conducted experiments by placing the attacking device near the targeted charging devices at different distances. Figure 14 presents the evaluations at a distance ranging from 0.98in (2.5cm) to 7.88in (20cm). Although BankSnoop achieves similar promising accuracy in different positions (underneath the table or near the smartphone), the overall performance decreases as the changes in the magnetic field can be difficult to monitor when the distance increases. In particular, when the distance is 20cm, BankSnoop's performance decreases to lower than 20% insufficient strength of the magnetic field disturbance, which remains undetectable by the attacking device.

6.2 Countermeasures

C1: Shielding magnetic field. One countermeasure to defend against attacks from BankSnoop is to prevent the magnetic traces from being eavesdropped. For example, manufacturers could add thicker cases to commodity wireless charging power bank products to shield the magnetic field to an undetectable degree [28]. Hence, the attacker needs to put the attacking device much closer to the victim or use more sensitive sensors to capture the magnetic traces, which inevitably increases the difficulties and costs of using BankSnoop to launch side-channel attacks.

C2: Signal obfuscation. Another countermeasure is to apply signal obfuscation mechanisms in the charging coils to generate indistinguishable current patterns so that the attacker cannot use the collected magnetic traces for user privacy inference. In practice, one can add random current noises (*e.g.*, Gaussian white noises [47]) to the primary coil or utilize a different charging protocol that dynamically switches the frequency and amplitude of the coil current [49] to obfuscate the captured signals. In addition, placing other charging devices in the vicinity can create extra magnetic fields and obfuscate the captured magnetic field disturbances.

Implementation. We implement $C1$ by wrapping a 0.5 cm thick insulation shield [37] to the power bank and implement $C2$ by leveraging the RIGOL DS 1052E signal generator [39] and then measure the unlocking key-press accuracy as well as the charging efficiency (start from 10%). Figure 14a and Figure 14b show the result, where we know even $C1$ and

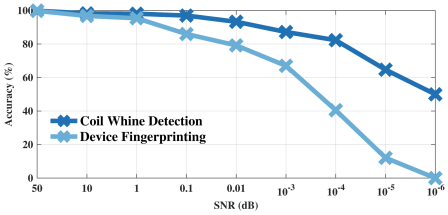


Figure 13: Impact of environmental noise at different signal-to-noise ratios (SNR).

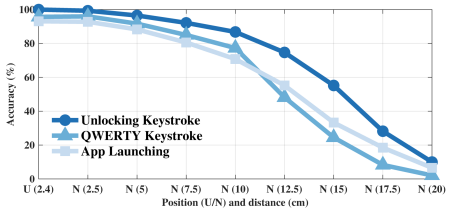
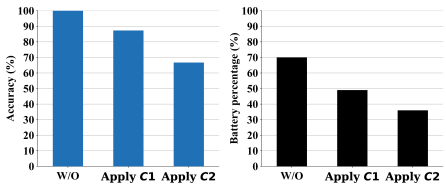


Figure 14: Impact of position and distance (U–underneath, N–near).



(a) Recognition accuracy. (b) Battery after charging.

Figure 15: Results of applying $C1$ and $C2$ to BankSnoop.

$C2$ could defend against BankSnoop, they also impact the charging efficiency. That is, *C1* also increases the distance of the charging coils and *C2* results in fluctuated coil currents, which both reduce the power transmission efficiency.

6.3 Limitations and Future Works

We have implemented BankSnoop to demonstrate the feasibility of the reported contactless side channel. While the results are promising, there still exist several limitations in the current work. First, BankSnoop is evaluated by attaching the attacking device underneath the table or putting it next to the target power bank for the proof of concept. Our work has not evaluated its performance in other possible scenarios, such as users holding the charging devices in their hands and performing activities on the run. Theoretically, BankSnoop is feasible to apply to those scenarios by adjusting the position and distance of the attacking device within a certain range to capture traces, whereas it inevitably increases the difficulty of launching attacks. Second, we consider a close and practical attacking distance in BankSnoop to demonstrate the feasibility because the two physical phenomena will attenuate as the distance increase, which requires tuning the models to adapt to a longer distance. We push these analyses to our future works.

7 RELATED WORKS

Wireless charging attacks. Qi protocol has become the de-facto wireless charging standard for mobile devices [42]. Nevertheless, recent researches reveal security vulnerabilities of Qi-certified wireless charging systems. Cour *et al.* [21] presented a website fingerprinting attack on wireless chargers from its current traces in the power line, which requires a stable charging voltage and a high battery level of the smartphone (*e.g.*, > 80%). Wu *et al.* [47] used a hidden coil to obtain induced current for hijacking the battery and identifying app activities. Moreover, EM-Surfing [24] utilized the induced voltage of an external resistor to monitor privacy leakages, *e.g.*, app usage and keystrokes. BankSnoop addresses the limitations in these prior works to launch contactless and end-to-end side-channel attacks that achieve fine-grained user privacy inference and realize fast adaptation.

Magnetic side-channel attacks. Recent years have witnessed the development of studies relevant to magnetic-based side-channel attacks. For instance, MagEar [23] utilizes the magnetic flux from the victim's earphone speaker to perform audio eavesdropping attacks. MagSnoop [11] injects malware to capture the sounds in a magnetic secure transmission (MST) process (*e.g.*, Samsung Pay) to recover the tokens of a credit card. In addition, electromagnetic (EM) emanation can be exploited to extract secret keys [16], reconstruct model architectures [5, 29], uncover screen messages [25] and keystrokes [19, 43]. Likewise, BankSnoop has demonstrated the feasibility of exploiting two magnetic-induced phenomena to attack wireless charging power banks.

8 CONCLUSION

In this paper, we report a new side channel in wireless charging power banks that can be exploited to launch contactless attacks to infer sensitive information from the charging smartphone, which leverages the coil whine and the magnetic field disturbance stemming from the wireless charging process. We have designed and implemented BankSnoop, an attack framework to demonstrate the feasibility of this new side-channel attack. To the best of our knowledge, it is the first attack on wireless charging power banks. Our extensive evaluation suggests that BankSnoop is effective in recognizing app launching/in-app activities and uncovering user keystrokes, and the few-shot learning module enables it to adapt to different scenarios while maintaining high accuracy.

ACKNOWLEDGMENTS

We sincerely thank our shepherd and all anonymous reviewers for their constructive feedback. This work was supported by CityU APRC grant 9610563, Hong Kong RGC (CityU 21219223, C1029-22G, CityU 21201420, CityU 11201422), The China Postdoctoral Science Foundation 2023M732791, NSFC (62101471), NSF of Shandong province (ZR2021LZH010), Shenzhen Science and Technology Funding Fundamental Research Program (2021Szvup126), and Hong Kong RGC GRF PolyU 15224121. Any opinions, findings, and conclusions in this paper are those of the authors and do not necessarily of supported organizations.

REFERENCES

[1] Adafruit. 2021. Electret Microphone Amplifier - MAX9814 with Auto Gain Control. (2021). https://www.adafruit.com/product/1713.

[2] ANKER. 2020. Anker MagGo. (2020). https://us.anker.com/pages/maggo.

[3] appfigures. 2021. Top Ranked iOS App Store Apps. (2021). https://appfigures.com/top-apps/ios-app-store/united-states/iphone/top-overall.

[4] Apple. 2022. MagSafe Battery Pack. (2022). https://support.apple.com/kb/SP846?viewlocale=en_US&locale=en_US.

[5] Lejla Batina, Shivam Bhasin, Dirmanto Jap, and Stjepan Picek. 2019. CSI NN: Reverse engineering of neural network architectures through electromagnetic side channel. In *Proceedings of the 28th USENIX Security Symposium*. 515–532.

[6] Anouar Belahcen et al. 2004. *Magnetoelasticity, magnetic forces and magnetostriction in electrical machines*. Helsinki University of Technology.

[7] Belkin. 2022. Magnetic Wireless Power Bank 2.5K. (2022). https://www.belkin.com/us/chargers/wireless/boost-charge-magnetic-wireless-power-bank-2-5k/p/p-bpd002/.

[8] Jin Chen, Per Jönsson, Masayuki Tamura, Zhihui Gu, Bunkei Matsushita, and Lars Eklundh. 2004. A simple method for reconstructing a high-quality NDVI time-series data set based on the Savitzky–Golay filter. *Remote sensing of Environment* 91, 3-4 (2004), 332–344.

[9] Xiang Chen, Yiran Chen, Zhan Ma, and Felix CA Fernandes. 2013. How is energy consumed in smartphone display applications?. In *Proceedings of the 14th Workshop on Mobile Computing Systems and Applications*. 1–6.

[10] Yushi Cheng, Xiaoyu Ji, Wenyuan Xu, Hao Pan, Zhuangdi Zhu, Chuang-Wen You, Yi-Chao Chen, and Lili Qiu. 2019. Magattack: Guessing application launching and operation via smartphone. In *Proceedings of the ACM Asia Conference on Computer and Communications Security (AsiaCCS)*. 283–294.

[11] Myeongwon Choi, Sangeun Oh, Insu Kim, and Hyosu Kim. 2022. MagSnoop: listening to sounds induced by magnetic field fluctuations to infer mobile payment tokens. In *Proceedings of the 20th Annual International Conference on Mobile Systems, Applications and Services (MobiSys)*. 409–421.

[12] Patrick Cronin, Xing Gao, Chengmo Yang, and Haining Wang. 2021. Charger-Surfing: Exploiting a Power Line Side-Channel for Smartphone Information Leakage. In *Proceedings of the 30th USENIX Security Symposium*. 681–698.

[13] Mian Dong and Lin Zhong. 2011. Chameleon: A color-adaptive web browser for mobile OLED displays. In *Proceedings of the 9th International Conference on Mobile systems, Applications, and Services*. 85–98.

[14] ElectronicWings. 2022. HMC5883L Magnetometer Module. (2022). https://www.electronicwings.com/sensors-modules/hmc5883l-magnetometer-module.

[15] Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the International Conference on Machine Learning (ICML)*. 1126–1135.

[16] Daniel Genkin, Lev Pachmanov, Itamar Pipman, Eran Tromer, and Yuval Yarom. 2016. ECDSA key extraction from mobile devices via nonintrusive physical side channels. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*. 1626–1638.

[17] Spherical Insights. 2022. Global Power Bank Rental Services Market Size, Share and Trends, Analysis and Forecast 2021 – 2030. (2022). https://www.sphericalinsights.com/reports/power-bank-rental-services-market.

[18] Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, and Pierre-Alain Muller. 2019. Deep learning for time series classification: a review. *Data mining and knowledge discovery* 33, 4 (2019), 917–963.

[19] Wenqiang Jin, Srinivasan Murali, Huadi Zhu, and Ming Li. [n. d.]. Periscope: A Keystroke Inference Attack Using Human Coupled Electromagnetic Emanations. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*.

[20] David M Kreindler and Charles J Lumsden. 2016. The effects of the irregular sample and missing data in time series analysis. In *Nonlinear Dynamical Systems Analysis for the Behavioral Sciences Using Real Data*. CRC Press, 149–172.

[21] Alexander S La Cour, Khurram K Afridi, and G Edward Suh. 2021. Wireless Charging Power Side-Channel Attacks. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*. 651–665.

[22] Mengyuan Li, Yan Meng, Junyi Liu, Haojin Zhu, Xiaohui Liang, Yao Liu, and Na Ruan. 2016. When CSI meets public WiFi: inferring your mobile phone password via WiFi signals. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*. 1068–1079.

[23] Qianru Liao, Yongzhi Huang, Yandao Huang, Yuheng Zhong, Huitong Jin, and Kaishun Wu. 2022. MagEar: Eavesdropping via audio recovery using magnetic side channel. In *Proceedings of the 20th Annual International Conference on Mobile Systems, Applications and Services (MobiSys)*. 371–383.

[24] Jianwei Liu, Xiang Zou, Leqi Zhao, Yusheng Tao, Sideng Hu, Jinsong Han, and Kui Ren. 2022. Privacy Leakage in Wireless Charging. *IEEE Transactions on Dependable and Secure Computing* (2022).

[25] Zhuoran Liu, Niels Samwel, Léo Weissbart, Zhengyu Zhao, Dirk Lauret, Lejla Batina, and Martha Larson. 2020. Screen gleaning: A screen reading TEMPEST attack on mobile devices exploiting an electromagnetic side channel. *Proceedings of the Network and Distributed System Security Symposium (NDSS)*.

[26] CrioSoft LLC. 2022. Amperes - battery charge info. (2022). https://apps.apple.com/us/app/amperes-battery-charge-info/id1245475416.

[27] EGO INNOVATION LTD. 2021. EGO MAGPOWER Gen.2 6000mAh 15W magsafe powerbank. (2021). https://www.egoshop.co/en/products/ego-magpower-15w-magsafe-6000mah-powerbank-1.

[28] Danyue Ma, Jixi Lu, Xiujie Fang, Ke Yang, Kun Wang, Ning Zhang, Bangcheng Han, and Ming Ding. 2021. Parameter modeling analysis of a cylindrical ferrite magnetic shield to reduce magnetic noise. *IEEE Transactions on Industrial Electronics* 69, 1 (2021), 991–998.

[29] Henrique Teles Maia, Chang Xiao, Dingzeyu Li, Eitan Grinspun, and Changxi Zheng. 2021. Can one hear the shape of a neural network?: Snooping the GPU via Magnetic Side Channel. (2021).

[30] Francisco Javier Ordóñez Morales and Daniel Roggen. 2016. Deep convolutional feature transfer across mobile activity recognition domains, sensor modalities and locations. In *Proceedings of the 2016 ACM International Symposium on Wearable Computers*. 92–99.

[31] Arduino Nano. 2022. Arduino Nano Document. (2022). https://docs.arduino.cc/hardware/nano.

[32] Tao Ni, Yongliang Chen, Keqi Song, and Weitao Xu. 2021. A Simple and Fast Human Activity Recognition System Using Radio Frequency Energy Harvesting. In *Adjunct Proceedings of the 2021 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2021 ACM International Symposium on Wearable Computers*. 666–671.

[33] Tao Ni, Guohao Lan, Jia Wang, Qingchuan Zhao, and Weitao Xu. 2023. Eavesdropping Mobile App Activity via Radio-Frequency Energy Harvesting. In *Proceedings of the 32nd USENIX Security Symposium*.

[34] Rui Ning, Cong Wang, ChunSheng Xin, Jiang Li, and Hongyi Wu. 2018. Deepmag: Sniffing mobile apps in magnetic field through deep convolutional neural networks. In *Proceedings of the IEEE International Conference on Pervasive Computing and Communications (PerCom)*. IEEE, 1–10.

[35] Rukundo Olivier and Cao Hanqiang. 2012. Nearest Neighbor Value Interpolation. *International Journal of Advanced Computer Science and Applications* 3, 4 (2012). https://doi.org/10.14569/ijacsa.2012.030405

[36] Hao Pan, Lanqing Yang, Honglu Li, Chuang-Wen You, Xiaoyu Ji, Yi-Chao Chen, Zhenxian Hu, and Guangtao Xue. 2021. MagThief: Stealing private app usage data on mobile devices via built-in magnetometer. In *Proceedings of the 18th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*. IEEE, 1–9.

[37] US Energy Products. 2023. US Energy Products (AD3) Reflective Foam Insulation Shield. (2023). https://www.amazon.com/US-Energy-Products-Reflective-Insulation/dp/B07R1S669V.

[38] J Ross Quinlan. 1996. Learning decision tree classifiers. *ACM Computing Surveys (CSUR)* 28, 1 (1996), 71–72.

[39] RIGOL. 2022. Rigol DS1052E. (2022). https://www.batronix.com/shop/oscilloscopes/Rigol-DS1052E.html.

[40] Seyed Ali Rokni, Marjan Nourollahi, and Hassan Ghasemzadeh. 2018. Personalized human activity recognition using convolutional neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32.

[41] Md Sahidullah and Goutam Saha. 2012. Design, analysis and experimental evaluation of block based transformation in MFCC computation for speaker recognition. *Speech communication* 54, 4 (2012), 543–565.

[42] Dries Van Wageningen and Toine Staring. 2010. The Qi wireless power standard. In *Proceedings of 14th International Power Electronics and Motion Control Conference (EPE-PEMC)*. IEEE, S15–25.

[43] Martin Vuagnoux and Sylvain Pasini. 2009. Compromising electromagnetic emanations of wired and wireless keyboards.. In *Proceedings of the USENIX Security Symposium*, Vol. 8. 1–16.

[44] Jindong Wang, Vincent W Zheng, Yiqiang Chen, and Meiyu Huang. 2018. Deep transfer learning for cross-domain activity recognition. In *Proceedings of the 3rd International Conference on Crowd Science and Engineering*. 1–8.

[45] Yuanda Wang, Hanqing Guo, and Qiben Yan. 2022. GhostTalk: Interactive Attack on Smartphone Voice System Through Power Line. In *Proceedings of the Network and Distributed System Security Symposium (NDSS)*.

[46] Wikipedia. 2022. Inductive charging. (2022). https://en.wikipedia.org/wiki/Inductive_charging.

[47] Yi Wu, Zhuohang Li, Nicholas Van Nostrand, and Jian Liu. 2021. Time to Rethink the Design of Qi Standard? Security and Privacy Vulnerability Analysis of Qi Wireless Charging. In *Proceedings of the Annual Computer Security Applications Conference (ACSAC)*. 916–929.

[48] Boyuan Yang, Ruirong Chen, Kai Huang, Jun Yang, and Wei Gao. 2022. Eavesdropping user credentials via GPU side channels on smartphones. In *Proceedings of the 27th ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*. 285–299.

[49] Shengqi Yang, Wayne Wolf, Narayanan Vijaykrishnan, Dimitrios N Serpanos, and Yuan Xie. 2005. Power attack resistant cryptosystem design: A dynamic voltage and frequency switching approach. In *Design, Automation and Test in Europe*. IEEE, 64–69.

[50] Guoming Zhang, Chen Yan, Xiaoyu Ji, Tianchen Zhang, Taimin Zhang, and Wenyuan Xu. 2017. Dolphinattack: Inaudible voice commands. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*. 103–117.