2022 19th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON) | 978-1-6654-8643-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/SECON55815.2022.9918604

MAGDEFENDER: Detecting Eavesdropping on Mobile Devices using the Built-in Magnetometer

Hao Pan †, Feitong Tan#, Wenhao Li†, Yi-Chao Chen †, Lanqing Yang†, Guangtao Xue†*, Xiaoyu Ji‡
† Department of Computer Science and Engineering, Shanghai Jiao Tong University
# School of Computing Science, Simon Fraser University
‡ College of Electrical Engineering, Zhejiang University
panh09@sjtu.edu.cn, feitongt@sfu.ca, {fire1997ice, yichao, yanglanqing, gt_xue}@sjtu.edu.cn, xji@zju.edu.cn

***Abstract*—This study reveals that on-board hardware modules leak electromagnetic (EM) emissions whenever audio or camera data is accessed, and proposes MAGDEFENDER scheme that explores the possibility of using the magnetometer built into mobile devices to detect eavesdropping instances by malicious apps and even the unscrupulous phone vendors. However, the target EM signals generated by accessing multimedia data is weak and tends to be buried beneath other noisy EM signals from apps running in the foreground. It is also subject to the external interference from geomagnetic signals generated by the device movement. To cope with the challenges, we adopt a generative adversarial networks (GAN) based model to facilitate the extraction of target EM signals indicating the occurrence of eavesdropping from the overall magnetometer readings. We also develop a neural network-based classifier with triplet loss embedding to identify the EM signals from the camera and/or microphones. Empirical results demonstrate the efficacy of MAGDEFENDER in recognizing instances of eavesdropping on cameras/microphones data, with average accuracy of 97.3% when applied to the trained devices, and average 91.5% on unseen mobile devices.**

***Index Terms*—electromagnetic side-channel, eavesdropping detect, mobile security**

I. INTRODUCTION

The availability of high-fidelity multimedia sensors (*i.e.*, cameras and microphones) and ubiquitous internet connectivity have prompted the development of numerous mobile applications (apps) and services; however, many malicious apps are able to use sensors in ways that violate user expectations and privacy. The New York Times has reported on apps using code from a company called Alphonso to listen for audio signals indicative of user behavior and preferences [1].

Existing solutions: Current mainstream operating systems (OSes), *e.g.*Android and iOS, are equipped with mechanisms that control and monitor the access to hardware sensors. However, these restrictions or indicators may have no effect at all on actions performed by these manufacturers themselves. Note however that these software-based solutions cannot identify malicious eavesdropping instances from the OS itself. Furthermore, these solutions can be easily bypassed by attackers with some hacking tricks even on unrooted phones, *e.g.*, transplantation attacks [2].

EM side-channel: Side-channel sensing technologies provide another opportunity for anti-eavesdropping. Several papers have explored the use of magnetometers to detect EM activity indicative of specific apps or webpages [3]; however, these schemes require the collection of a app-/webpage-labeled EM training dataset, which is impractical in actual usage scenarios. Furthermore, the existing works can only identify the most active apps operating in the foreground; *i.e.*, they disregard foreground apps running in parallel as well as those running in the background.

Proposed scheme: Our proposed MAGDEFENDER identifies eavesdropping instances by analyzing EM signals generated during the acquisition of camera/mic data, as detected by the magnetometer built into the mobile device. This scheme can be implemented on any commercial off-the-shelf mobile device without the need for hardware/system kernel software modification, root/jailbreak, or additional system permissions.

Challenges: Development of the MAGDEFENDER system imposed a number of daunting challenges: **(1)** EM signals of interest that are associated with accessing to camera/mic data must be extracted from noisy magnetometer readings, which also contain noisy EM signals from legitimate running apps as well as magnetic field signals caused by device motion and other sources of interference. **(2)** A highly robust classification model is required to characterize EM signals, particularly when confronted by widely divergent EM emission patterns resulting from variations in hardware configuration, *e.g.*, camera/mic hardware modules.

Our solution: (1) Preliminary experiments were conducted to analyze the characteristics of EM signals generated while camera/mic data is accessed (target EM signal) as well as ambient magnetic field signals (interference). The target EM signals appeared stable and regular, compared to the external geomagnetic signals caused by device motion and the internal EM signals generated during the execution of other apps. We used mean-median filtering to remove signals with excessive fluctuations and developed a generative adversarial network (GAN) based model to separate target EM signals from noisy EM signals based on spectrogram analysis. **(2)** We developed a neural network-based classifier with triplet loss embedding to maximize the differences between EM signals from the camera and mic, while minimizing the differences between EM signals from different sensors modules of the same type. The resulting classification model is able to identify instances of eavesdropping, despite the diversity of mobile devices.

We implemented the MAGDEFENDER prototype on both

* Corresponding author.

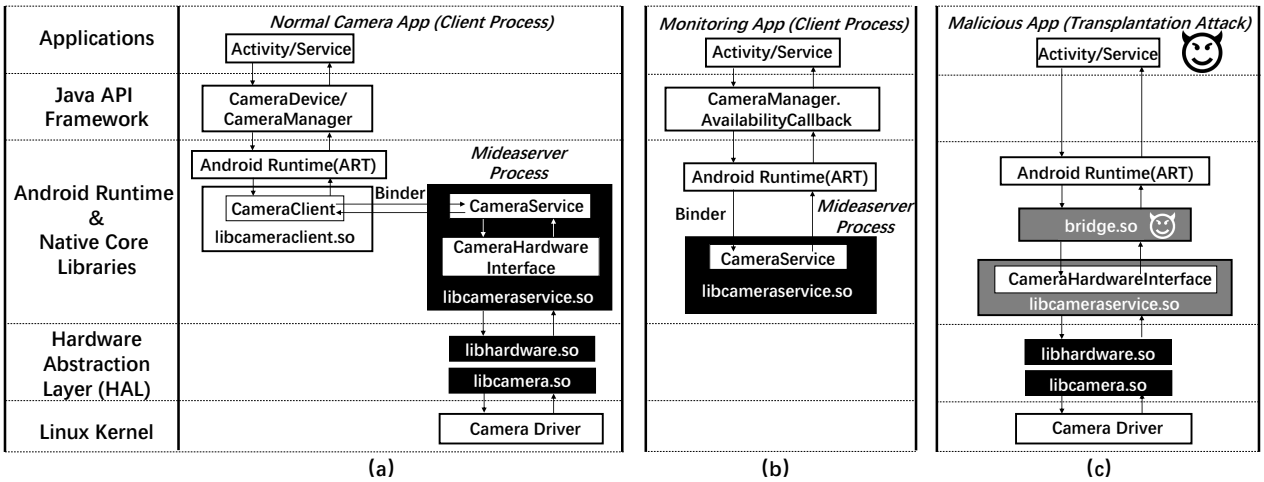


Fig. 1. (a) Workflow of camera apps when using the Android Camera APIs; (b) Process involved in monitoring app to detect instances of using the *CameraManager* to call camera clients; (c) Model of transplantation attack involving stealthy use of camera device by avoiding Android APIs [2]

Android and iOS devices, and conducted extensive experiments to demonstrate the effectiveness and generalization of our proposed MAGDEFENDER in detecting eavesdropping events. In the last, we also explore the limitations and outline a number of countermeasures to make our system failed.

The main contributions of this work are as follows:

- We developed a third-party app capable of monitoring the working status of the mics and cameras in mobile devices through analyzing magnetometer readings without using OS-related media APIs.
- We conducted a pilot study to verify that the media sensors and corresponding hardware modules in mobile devices generate unique and consistent EM signals detectable using the built-in magnetometer.
- We designed a general denoising scheme combined with a mean-media filtering method and a DNcGAN (DeNoising conditional Generative Adversarial Network) to extract the target EM signals (generated when accessing camera/mic data) from overall magnetometer readings. A neural network-based classifier with triplet loss embedding was also proposed to identify the eavesdropping types.

II. BACKGROUND AND VULNERABILITY ANALYSIS

In the Android, multimedia programming is based on the client/server architecture. The client process refers to any app using the Android Camera/Audio Service, whereas the server refers to *Mediaserver*. The workflow of an app accessing the camera data is presented in Fig. 1(a). Common attacks that involve in accessing camera/mic data on an Android device call multimedia-related APIs provided by an Android SDK. However, this type of spy-on-user attack is not stealthy, as it is unable to evade monitoring apps (Fig. 1(b)). Sophisticated attackers are able to take photos and record audio/video without calling Android APIs. These stealthy methods are referred to as transplantation attacks [2].

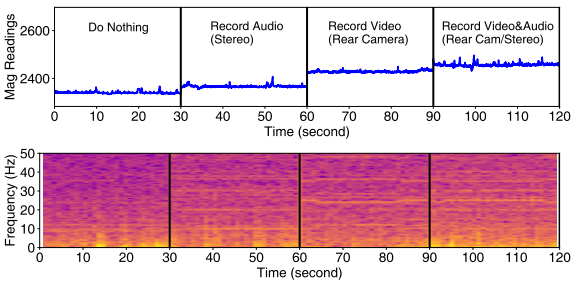
Repackaging the apps (*e.g.*, QRCode Scanner or Speech-to-Text) that uses `CAMERA/RECORD_AUDIO` permissions is straightforward for bypassing the permission restrictions. Moreover, the Checkmarx Security Research Team also found multiple concerning vulnerabilities stemming from permission bypass issues [4], which can also be utilized by the transplantation attack model. Fig. 1(c) illustrates the process after the malicious apps has been installed on the victim's mobile device. Basically, transplantation attacks involve loading the *Mediaserver* process into the address space of the malicious app, thereby allowing the Java code in the Application layer to pass through the Framework layer and access the ***libcameraservice_transplanted.so*** in the Native Core Libraries layer. Functions in the ***CameraHardwareInterface*** then call *.so* libraries in the HAL layer to talk with the Camera Driver in the Linux Kernel. Java Native Interface (JNI) programming is utilized as a bridge to connect Java code in the Application layer with transplanted *.so* libraries. The native code is compiled as an *.so* file (***bridge.so*** in Fig. 1(c)).

Unlike normal multimedia-related apps, this type of malicious app jumps over the Android API Framework and eliminates the Binder IPC between *Mediaserver* processes by deleting the *libcameraclient.so* library and Camera Service in the *libcameraservice.so*. Under these conditions, the API/Hook-based monitoring solutions mentioned above are unable to detect eavesdropping behavior that involves in accessing camera/mic data.

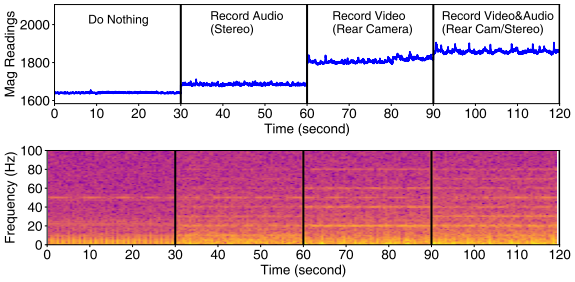
III. ATTACK AND DEFENSE MODELS

Threat Model: This study addresses two stealthy attack scenarios involving **(1)** unscrupulous mobile phone vendors access the camera/microphones in the background by interacting with kernel drivers directly; and **(2)** sophisticated attackers inject buggy code (or use malicious third-party libraries) for eavesdropping without using OS multimedia-related APIs.

We divide these sophisticated attackers' capabilities into three levels. The first level attackers can only use the APIs in the operating system. This type of attackers can run native code as well but without root privileges. The second level attackers can run native code with root privileges in the user space, however cannot run code with kernel privileges or secretly modify the system images. More specifically, in this case, we assume that the attacker cannot leverage the kernel

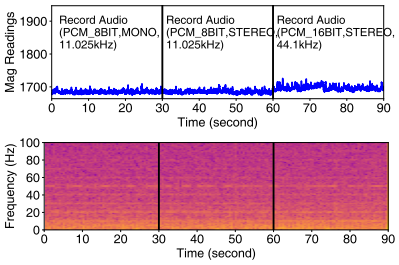


(a) iPhone 7Plus.

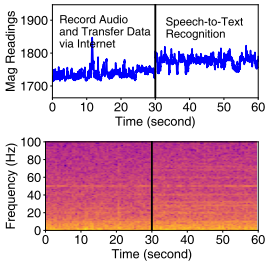


(b) Huawei P20Pro.

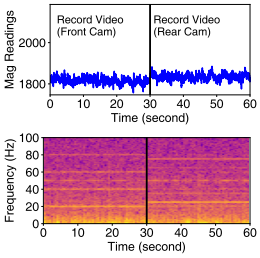
Fig. 2. Magnetometer readings and corresponding spectrograms when recording video and/or audio on different mobile phones



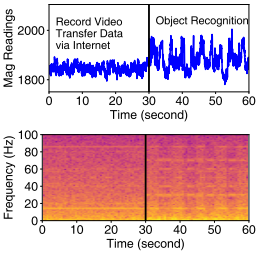
(a) Recording audio with microphone and different parameters settings



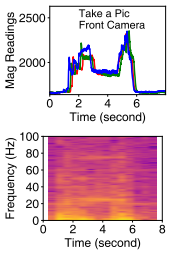
(b) Executing different audio-related tasks



(c) Recording video with front/rear cameras



(d) Executing different video-realted tasks



(e) Taking a picture (3 times).

Fig. 3. Magnetometer readings and corresponding spectrograms obtained while executing various audio-/camera-related tasks as a proxy for accessing microphone/camera data.

vulnerabilities to inject code and assume that the kernel is configured to prevent a root user from easily modifying the running kernel memory. The third level attackers can leverage the vulnerabilities of the kernel to compromise it and hence can run code with kernel privileges. Moreover, he can also rewrite the kernel images, have physical access to the device and can manipulate the hardware directly. In the real attack scenario, the vast majority of attackers can only reach up to the second level of attacking capability.

Defense Model: The fact that the above mentioned attack strategies can easily evade the software-based monitoring methods. In this study, we developed a novel defense model, which uses the on-board magnetometer in mobile devices to detect electromagnetic (EM) emissions generated whenever the built-in microphone and/or cameras are being used, even surreptitiously. The attackers may know the means of auditing multimedia sensor access via EM side channel and could invalidate it by tampering with the magnetometer readings. However, only the Level-3 attackers could achieve tampering with the magnetometer readings by modifying the sensor driver. Once the attacker has extremely high hardware access and control over the victim's device, then all auditing methods will be equally invalidated. Therefore, here we assume that the built-in magnetometer readings are trustworthy and can not be tampered with attackers in the real attack scene.

IV. Preliminary Analysis

Preliminary experiments were conducted to answer three fundamental questions: **(i)** When apps access camera/mic data, does the mobile device generate EM emission signals that could be captured using the built-in magnetometer? **(ii)** If so, what are the characteristics of these EM signals generated by apps that perform eavesdropping via the built-in camera/mic? **(iii)** What other factors affect magnetometer readings, and what are the characteristics of noisy signals? Our answers to these questions demonstrated the potential of using EM side-channel sensing to detect instances of eavesdropping via the built-in camera/mic and helped to elucidate the challenges involved in developing this technology.

Proprietary apps were installed on two representative smartphones to enable the continuous recording the built-in magnetometer at maximum sampling rates: iPhone 7 Plus ($100Hz$) and Huawei P20Pro ($200Hz$). The magnetometer readings in this paper mean the raw sensor data (*e.g.*, 12/16 bit), and the EM signals emitted from the device were calculated using the total magnetic field intensity from the three-axis readings as follows: $mag_t(t) = \sqrt{mag_x(t)^2 + mag_y(t)^2 + mag_z(t)^2}$.

A. EM Signals When Accessing Multimedia Data

To minimize interference during the first experiment, the phones were placed on a table (static) and no other active app tasks were executed. Fig. 2 presents readings from the built-in magnetometer in the time and frequency domains. The magnetometer patterns generated while the device was recording video and/or audio were regular and easily differentiated from those obtained while the device was "Doing Nothing". The results of this first preliminary test demonstrated that the built-in magnetometer is well-suited to capturing EM emission signals generated by mobile devices while accessing camera/mic data. Meanwhile, a cursory comparison of Figs. 2(a) and 2(b) revealed that different devices generated different EM signals, when performing the same task. This can no doubt be attributed to the fact that manufacturers differ in their selection of camera/mic modules and codec chips, and corresponding codes (*e.g.*, hardware drivers, interface libraries) in the HAL layer and Linux Kernel layer. Widely divergent EM emission patterns resulting from variations in multimedia

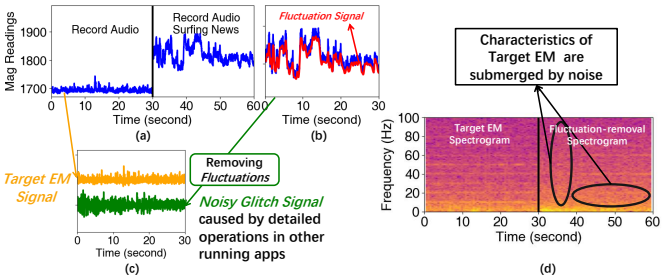


Fig. 4. The target EM signals (*e.g.*, accessing mic data) can be interfered by those from other concurrent running app tasks (*e.g.*, surfing news)

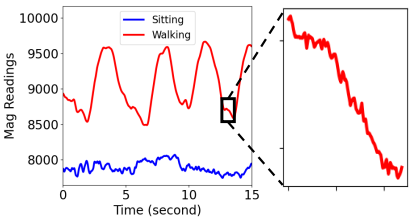


Fig. 5. Magnetometer readings caused by surrounding geomagnetic signals due to the smartphone motion

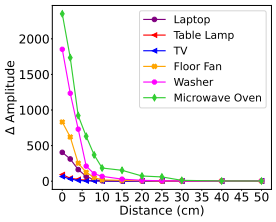


Fig. 6. Changes in EM amplitude at various distances from household appliances.

hardware will enhance difficulty to identify eavesdropping behaviors on untrained devices, we will discuss in Sec. V-C3.

B. EM Signals When Executing Eavesdropping Apps

As mentioned in Sec. III, the characteristics of EM signals differ according to which eavesdropping app tasks are running in the background. Thus, the second experiment was meant to characterize the EM signals generated by the smartphone, while executing a variety of app tasks that involve accessing camera/mic data.

1) Executing Audio-related App Tasks: We first programmed the devices to access audio data under a variety of parameter settings (`sampleRateInHz`, `channelConfig`, `audioFormat`) and collected their EM signals. We also examined the EM signals generated while executing apps that use audio data in conjunction with other functions, such as speech recognition and apps that transmit audio data over the Internet. As shown in Figs. 3(a) and (b), the magnetometer readings revealed the following important points: (i) EM signals generated by the device while recording audio data did not vary obviously, despite changes in parameter settings; (ii) EM signals generated by the device while performing complex audio-related tasks seem have more interference noises which superimposed over the target EM signals, which conceal the characteristic EM signals generated by accessing audio signal to a certain extent.

2) Executing Camera-related App Tasks: Our analysis included a comparison of EM signals generated while recording video using the rear or front camera modules (see Fig. 3(c)), as well as those generated while executing apps that involved the recording of video data in conjunction with other functions, such as face recognition (see Fig. 3(d)). We found that the EM signals differed according to which data stream was being accessed (front or rear camera), and EM signals generated while taking photos are characterized by sequential changes (Fig. 3(e)) which concealed the target EM signals (accessing camera stream data) in the form of noise. Also, we discovered that the EM signals generated during video data processing operations are similarly superimposed over the target EM signals, which means that it should be possible to identify eavesdropping instances based on target EM signals emitted whenever camera data is accessed. In this study, we treat all EM signals generated by other operations (*e.g.*, data processing and transmission) that are not related to multimedia data acquisition as noise, and systematically discuss this kind of interference noises in Sec. IV-C1.

C. Other Factors Affecting Magnetometer Readings

1) Internal EM Interference Signal: The execution of other operations in eavesdropping apps and other app tasks/services also produce EM interference signals; therefore, we sought to characterize the EM signals generated while simultaneously running eavesdropping operations in the background and other legitimate apps in the foreground. The magnetometer readings in Fig. 4(a) revealed that when surfing news/posts on Twitter, the device generated non-negligible interference signals superimposed over the target EM signals generated by accessing audio data in the background.

In-depth analysis allowed us to divide the noisy EM signals into two parts. One part referred to as a "fluctuation" signal (red line in Fig. 4(b)) presents coarse-grained changes in the amplitude of the signal over a medium-to-long time span. This type of noise is caused mainly by changes of intensity of tasks executed by the CPU. The other type of is referred to as glitch noise, which is caused by the CPU executing fine-grained operations associated with app tasks. As indicated by the green line in Fig. 4(c), "glitch" noise becomes apparent after the removal of fluctuation signals (see Sec. V-C1) and appears far more random than the target EM signal (orange line in Fig. 4(c)). The corresponding spectrograms (Fig. 4(d)) also revealed that the "glitch" noisy signal submerged the spectral characteristics of the target EM signals.

2) External Interference Signals: Any movement and location changes of device the user makes while using a mobile device could alter the position and/or orientation of the device, resulting in changes in the geomagnetic signals. Fig. 5 respectively present signals generated while the user was sitting and walking around. Clearly, motion-induced variations in geomagnetic signals also presented as "fluctuation" signals, with a larger amplitude and lower frequency than the EM signals generated from the mobile devices. The variations in geomagnetic flux cannot be disregarded in the EM analysis.

Other electrical devices (*e.g.*, household appliances) also leak EM emissions at levels that cannot be disregarded out of hand. We conducted a series of experiments to determine the actual attenuation of EM emission signals as a function of distance by collecting EM signals emitted from various electrical devices using the magnetometer built into the Huawei P20Pro at various distances from the appliances. We then drew up a

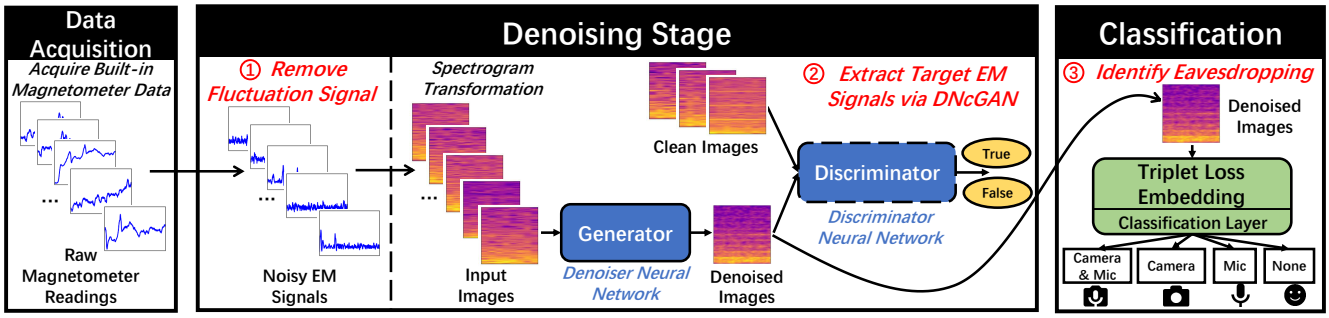


Fig. 7. Overview architecture and processing pipeline of the MAGDEFENDER system.

diagram plotting the attenuation of EM intensity with distance (see Fig. 6). When the mobile device was $\geq 25cm$ from large household appliances (washing machine and microwave oven), the built-in magnetometer was unable to detect the EM signals. In the case of the appliances with low EM emission (table lamp and TV), $5cm$ was sufficient to eliminate any interference.

V. MAGDEFENDER DESIGN

A. Denoising task

Our preliminary experiments in the above section revealed three components in the overall magnetometer readings: i) medium-to-long fluctuation signals (caused by geomagnetic interference and other running apps' task intensity), ii) fine-grained glitch signals (caused by the detailed operations in other running apps), and iii) target EM signals (caused by accessing camera/mic data). Here, we introduce the denoising method developed for the extraction of our target EM signals associated with the acquisition of camera/mic data from the overall built-in magnetometer readings.

B. Data Collection

We begin with the assembly of a dataset comprising magnetometer readings recorded from mobile devices. Specifically, the dataset comprised exemplar (clean) signals and instance (noisy) signals, which were subsequently paired for training.

Exemplar Data. We first collected exemplar signals generated when accessing camera/mic data. With the mobile device stationary on a table and all other app tasks disabled, we recorded magnetometer readings in the background. Note that the readings obtained without multimedia-related tasks were labeled **"none"**. We then ran a proprietary app that requested access to the audio data stream and labeled the corresponding data **"audio"**. Using the same setup, we also requested access to video data streams without preview (w/ or w/o audio) respectively from the rear and front camera modules and labeled the corresponding data using the term **"camera"/"audio-camera"**. Note that we did not differentiate between the front and rear cameras because victims are primarily concerned about the fact of eavesdropping rather than which camera is used.

Instance Data. The collection and labeling of noisy magnetometer readings involved turning off permissions to access the camera and microphone (for all apps on the mobile devices) and allowing users to operate the mobile devices at will (without transplantation attacks), the corresponding magnetometer readings were labeled using the term **"none"**. Similarly, we ran our proprietary apps with various media-related types of tasks in the background, while respectively running the above programs, which were labeled using the terms **"audio"**, **"camera"**, and **"camera-audio"** respectively. We also let our apps take pictures in the background, and labeled the signals with **"camera"**.

C. System Design

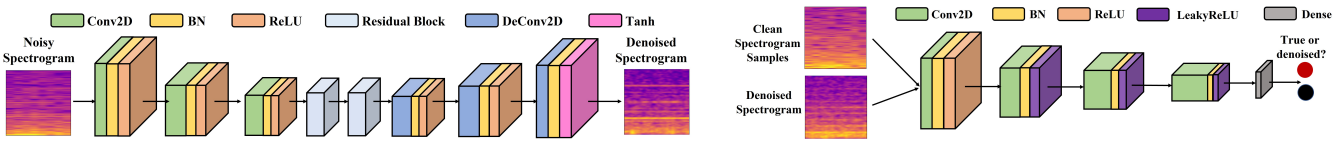
Our objective in this work was to determine from built-in magnetometer readings whether an app task/service was accessing camera/mic data. We also sought to identify which multimedia sensor was being accessed (*e.g.*, mic and/or camera). Fig. 7 presents an overview architecture of the proposed MAGDEFENDER. In the denoising stage, the fluctuation signals and glitch noisy signals are removed respectively (Step ① and ② in Fig. 7) to facilitate the extraction of the target EM signals, which are then fed into a classification model to identify instances of eavesdropping (Step ③ in Fig. 7).

1) Fluctuation Removal: As shown in Fig. 4(b) and 5, unpredictable changes in the characteristics of signal fluctuations over time cannot be tracked using conventional fitting methods (*e.g.*, polynomial fitting (PF) and iterative restricted least square (IRLS)) . The limited sampling rates of the built-in magnetometers ($100 \sim 200Hz$) can lead to aliasing in the low-frequency region of target EM signals (*i.e.*, undersampling). Utilizing filtering methods in the frequency-domain (*e.g.*, wavelet transform) to filter out the fluctuation signals, will also tend to erase the spectral characteristics of target EM signals.

Thus, we employed Mean-Median Filtering (MMF) to eliminate the effect of fluctuations over time. Mathematically, this process can be described as the decomposition of a given time series $M(t)$ (the built-in magnetometer readings) into a medium-to-long term baseline $b(t)$ and a short-term cycle $c(t)$, as follows:$M(t) = b(t) + c(t)$. MMF utilizes the convex combination of the sample median and sample mean of the signal $M(t)$ as:

$$b^{'}(t) = (1-\alpha) \times mean(M(t)) + \alpha \times median(M(t)) \quad (1)$$

where, $\alpha \in [0,1]$ is the 'contamination factor'. We then smoothed the $b^{'}(t)$ with a Gaussian function to remove sharp discontinuities, and obtain the fluctuation-free signals as follows: $c^{'}(t) = M(t) - Gaussian(b^{'}(t))$.



(a) Generator neural network. (b) Discriminator neural network.

Fig. 8. Illustration of proposed DeNoising conditional Generative Adversarial Network (DNcGAN).

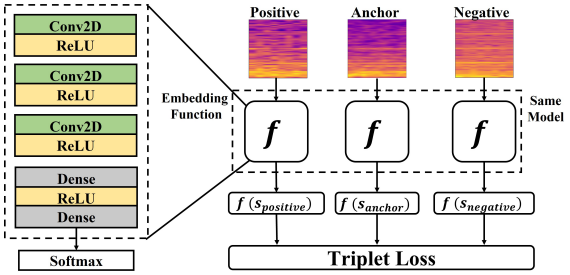


Fig. 9. Classification network with triplet loss training.

2) Denoising conditional GAN: After removing the fluctuation noisy signals, extracting the steady-state and periodic characteristic of our target EM signals from accessing cameras/mic data requires the removal of the glitch noisy signals. Instead of processing data in the time-domain, we transformed time-series data ($c^{'}(t)$) into spectrograms, due to the fact that spectral-image contains more information on the underlying periodicity.

In order to solve image-to-image translation problem from the noisy spectrogram to the clean spectrogram, there are some statistics-based methods can be applied to achieve denoising, but the major drawback is that there are many tunable parameters needed to be defined, and the fact that their performances rely on the models with assumptions. In this study, we proposed a denoising neural network to obtain a clean spectrogram presenting only the target EM signals using a Generative Adversarial Network (GAN) [5]. The proposed model is referred to as DeNoising conditional GAN (DNcGAN), which is highly effective in mapping from a source data distribution to a target data distribution. Taking a noisy spectrogram image as its input, generator G is trained to create a clean spectrogram by creating an output distribution with characteristics similar to those found in actual clean spectrograms. Discriminator D is used to produce an adversarial loss by detecting whether the input is from G or an actual clean spectrogram.

DNcGAN loss. The aim of DNcGAN is to solve a min-max problem between generator G and discriminator D.

$$\min_{G}\max_{D}\mathbb{E}_{x\sim\mathbb{P}_r(x)}[logD(x)]+\mathbb{E}_{\tilde{x}\sim\mathbb{P}_g}[log(1-D(G(\tilde{x}))] \quad (2)$$

where $\mathbb{P}_g$ is the denoised spectrogram distribution implicitly defined by $\tilde{x}=G(z)$, input z is the noisy spectrogram, and $\mathbb{P}_r$ indicates the distributions of actual clean spectrograms. Unfortunately, the adversarial loss equation (Eq. 2) above is insufficient to constrain the generator to produce an output close to the distribution of an actual clean spectrogram, thereby undermining the stability of model training. Thus, we adopted an alternative objective function referred to as Wasserstein GAN (WGAN) [6], which utilizes the Wasserstein distance to measure the difference between two distributions. We further enhanced the stability of training by combining WGAN with a gradient penalty, as follows:

$$\begin{aligned}\mathcal{L}(GAN)=\mathbb{E}_{\tilde{x}\sim\mathbb{P}_g}[D(\tilde{x})]-\mathbb{E}_{x\sim\mathbb{P}_r}[D(x)]+\\ \lambda\mathbb{E}_{\hat{x}\sim\mathbb{P}_{\hat{x}}}[(||\nabla_{\hat{x}}D(\hat{x})||_2-1)^2]\end{aligned} \quad (3)$$

where $\mathbb{P}_{\hat{x}}$ is defined as uniform sampling along straight lines between pairs of points sampled from $\mathbb{P}_r$ and $\mathbb{P}_g$. In our experiments, we set λ at 10 with all any other hyper-parameters matching those of WGAN-GP.

Generator G in Fig. 8(a) has network architecture similar to that of the "encoder-decoder" network. The "encoder" has a three-layer convolutional structure denoted as $Conv(32)-Conv(64)-Conv(64)$. Each layer is followed by a ReLU activation function. To enable the extraction and retention of more information, we increased the number of weighted layers between the encoder and decoder. We also used two residual blocks as this allows the skipping of connections to concatenate all channels between layers. To capture image details, the decoder includes three deconvolution layers denoted as $DeConv(64)-DeConv(64)-DeConv(32)$ to upsample the image in order to magnify slight differences.

Discriminator D (see Fig. 8(b)) is a network of "encoder" structure that contains four convolution layers, denoted as $Conv(32)-Conv(64)-Conv(64)-Conv(128)-Dense$. The other three convolutions use LeakyReLU and the first layer uses the ReLU activation function. The discriminator determines whether an image from the generator is similar to sufficiently similar to actual clean spectrogram samples. The binary decision generated by discriminator is then sent to an optimizer to enhance the performance of generator.

3) Classification with Automatic Adaption to Mobile Devices: After denoising the given EM spectrogram, our final objective is to determine whether the denoised spectrogram includes signals characteristic of tasks involved in accessing camera/mic data (presumably by eavesdropping apps). Unlike the noisy EM signals generated by typical app tasks/ services (unvarying across devices) [3]; target EM signals tend to vary with the mobile devices (see Fig. 2). There is no practical way to collect exemplar EM signals for every mobile device on the market just to create a training dataset. Even if this were attempted, it is unlikely that the classifier would perform well when applied to new (unseen) devices. We therefore implemented a classification neural network with triplet loss embedding, to assist in discriminating among EM spectrograms. Compared with the conventional contrastive loss that focuses on gathering as many positive examples as feasible, the triplet loss merely requires positive samples to be closer than

negative samples. Thus, the triple loss embedding can adjust to various degrees of intra-class variance for distinct classes and be utilized to train the classification model to find common features of EM signals generated by accessing multimedia data across various mobile devices.

Fig. 9 summarizes the concept of triplet loss training. The triplet embedding module generates (s_a, s_p, s_n) triplets, where s_a is an anchor sample (*e.g.*, "audio"), s_p is a positive sample in the same category (*e.g.*, "audio") as s_a, and s_n is a negative sample from any other category (*e.g.*, "camera"). All three samples are passed through the NN for embedding function f. Finally, triple loss is used to minimize the distance between embeddings ($f(\cdot)$) of the anchor and positive samples, and thereby maximize the distance between the anchor and negative samples. Let $\mathcal{T}$ be the set of all possible triplets $\tau = (s_a^\tau, s_p^\tau, s_n^\tau)$. The triplet loss is meant to enhance separation between positive and negative pairs by adding a safety margin $m \in \mathbb{R}^+$. For any triplet τ, the aim is to achieve $\triangle_\tau + m < 0$ where

$$\triangle_\tau = ||f(s_a^\tau) - f(s_p^\tau)||_2^2 - ||f(s_a^\tau) - f(s_n^\tau)||_2^2 \quad (4)$$

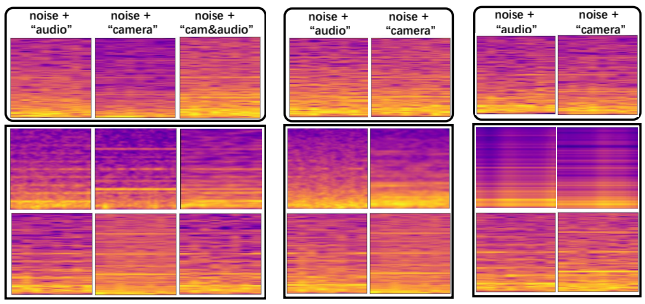
More precisely, loss can be defined as follows:

$$\mathcal{L}(\mathcal{T}) = \sum_{\mathcal{T}} \max(0, \triangle_\tau + m) \quad (5)$$

The neural network classification (embedding function f) is denoted as $Conv(32) - Conv(64) - Conv(64) - Dense(128) - Dense(32)$, and the details are presented on the left of Fig. 9. After embedding, we add a softmax layer to identify instances of eavesdropping.

4) Overall Training: Training the DNcGAN and the subsequent classification model involved slicing labeled instance and exemplar magnetometer readings into time windows of 5 seconds with overlap of 1 second (default settings). We then implemented MMF to filter out fluctuation signals and transform the resulting signals into spectrogram images (with a unified pixel size: $64 \times 64 \times 1$).

After preparing the training dataset, it is necessary to train the two neural networks: DNcGAN and classification NN. If DNcGAN were trained independently, then the discriminator could only determine whether the denoised spectrogram is "generated" or "real", which would result in a larger number of candidate denoising results. For example, the generator cleaned the noisy spectrogram labeled "audio" to create a clean spectrogram with a distribution similar to that of "none"; however, the entire DNcGAN will not be punished because that the discriminator cannot distinguish them. Therefore, we concatenate the subsequent classification NN using DNcGAN to enhance discriminative ability. Note that during training, the positive and negative samples in each triple τ were randomly sampled from clean spectrogram images which are transformed from the exemplar signals. In the experiment, we employed the Adam optimizer in the training and the overall loss function was denoted as: $\mathcal{L} = \mathcal{L}(GAN) + \mathcal{L}(\mathcal{T})$.



(a) GNcGAN (overall training) (b) GNcGAN (trained alone) (c) Low-rank matrix decomposition

Fig. 10. Comparisons between denoised spectrograms generated by several denoising methods and ground truth. First row is the noisy images, second row is the denoised results, and third row is the ground truth

VI. Evaluation

A. Experiment Setup

Mobile Devices and Participants: We selected 16 testing smartphones, and ten of the devices were used to train the proposed MagDefender system, and six were used to evaluate the performance when applied to unseen devices. We also recruited 8 university students from different majors as our participants, including 5 males and 3 males and ranging from 22 to 27 years old. Eight participants were each allocated two mobile devices pre-installed with our customized app.

Dataset Collection: During each participant's use of the mobile device, the customized app periodically requested camera/mic data in the background with the user's permission, and recorded the corresponding starting and ending timestamp of accessing multimedia sensor data for subsequent EM signal labeling. Also, users could used the audio/camera-related apps and they need to record the related app usage timestamp artificially. Throughout the whole process, the customized app continuously recorded the built-in magnetometer readings and saved them to local storage. The EM dataset was labeled using the same methods outlined in Sec. V-B.

Model Training: We utilized the $10-$fold cross-validation procedure for determining the best parameters of the MagDefender model (*e.g.*, the denosing GAN model and the classification NN). In details, the 9 folds (the selected 9 mobile devices) of the training dataset were used to trained the MagDefender, and the fold left out was used for test. By applying grid search techniques, we can determine the final model parameters with the best performance.

B. Methodologies Evaluation

1) Denoising Performance: We first evaluated the performance of the DNcGAN generator (denoiser) when trained separately and trained in conjunction with the classification NN with the training dataset. We also assessed a conventional low-rank matrix decomposition method (RLRM) [7] for the sake of comparison. The results in Fig. 10 show that the output of the generator after training DNcGAN alone confused information related to multimedia sensors, and the conventional matrix decomposition method resulted in the loss of many spectral features. As expected, training DNcGAN in conjunction with

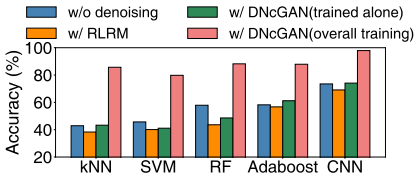


Fig. 11. Performances between different denoising methods. RLRM is a matrix decomposition method in [7].

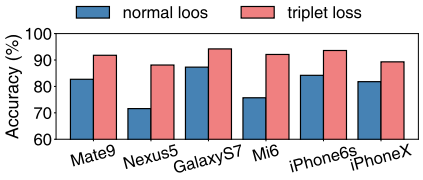


Fig. 12. Performance of the classification NN models on the unseen devices with different loss functions.

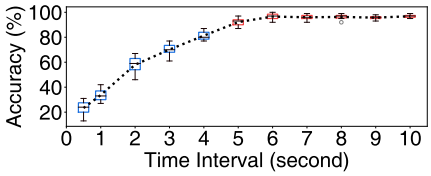
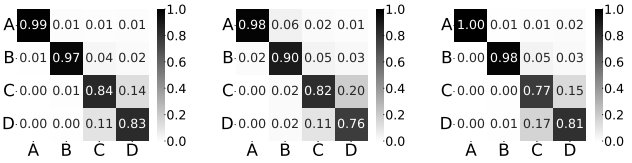


Fig. 13. Time interval lengths vs. eavesdropping behaviors classification performance.



(a) Huawei Mate9 (b) LG Nexus5 (c) iPhoneX

Fig. 14. Confusion matrix while classifying eavesdropping behaviors, where A-D denote "none", "audio", "camera", and "camera-audio".

the classification NN enabled the generator to eliminate noise while retaining distributions corresponding to the operation of multimedia sensors.

The efficacy of denoising was assessed via comparisons with popular classification models on denoised spectrograms and raw noisy spectrograms. In details, we implemented conventional spectral features (*e.g.*, mean value, standard deviation, kurtosis, crest factor, flatness) in conjunction with four conventional machine learning methods (kNN, SVM, Randoom Forest, and adaboost), and a CNN (with the same model in the left of Fig. 9) for classification. From the results in Fig.11, we can draw the following conclusions: (1) Direct classification of raw noisy spectrograms resulted in very poor performance, with the best performance of 73.5% accuracy when using a CNN. (2) Conventional matrix decomposition and DNcGAN (trained alone) both led to the loss of important information, thereby negating any improvement in accuracy. (3) DNcGAN (overall training) presented satisfactory performance (97.3%) in identifying instances of eavesdropping.

2) Triplet Loss Embedding: We sought to verify the effectiveness of triplet loss embedding by assessing the classification performance when applied to unseen devices. This was achieved by designing another classification NN with the same architecture (in the left of Fig. 9) but a normal contrastive loss function instead of triplet loss. Classification performance was assessed using six unseen devices with the testing dataset. The classification results in Fig. 12 demonstrated that triplet loss training enhanced classification performance when applied to unknown devices. And confusion matrices of classifying cam/mic working status on the selected three smartphones (whose performances were not relatively well) were presented in Fig. 14, we found that there existed the misclassification of "camera" and "camera-audio". Overall our proposed system still obtained average 91.5% of accuracy, and 93.6% recall when detecting eavesdropping on unseen devices.

3) Eavesdropping Time Interval Selection: MAGDEFENDER was tasked with the extraction of usable information from magnetometer readings over fixed time intervals of various lengths. Assigning a short time interval would enable fine-grained differentiation between eavesdropping behaviors; however, this would greatly hinder classification performance, due to a lack of time series information for feature extraction. After the removal of time-dependent fluctuations, we divided the magnetometer readings into segments matching the assigned time intervals to generate spectrogram images ($64\times64\times1$ pixels). As shown in Fig. 13, setting a time interval of greater than five seconds improved the average accuracy to roughly 97% with negligible standard deviation.

C. Power Consumption

MAGDEFENDER depends on continuous magnetometer readings to detect eavesdropping As a result, our proposed MAGDEFENDER consumes more power than software-based solutions. We measured the power consumption on the 16 mobile devices by running the MAGDEFENDER app for 1 hour and comparing the results when the phones were left to idle and when the phones were running a typical software solution, Access dots [8]. The average power consumption results showed that continuously running MAGDEFENDER for 1 hour consumed only 11% more power than idling and only 7% more than running Access dots on average. This is acceptable because our system can achieve more powerful anti-eavesdropping at the expense of a little energy consumption.

VII. RELATED WORK

A. Eavesdropping Detection Techniques

Hook-based solutions: Hooking is a technique for inserting codes into a system call for alteration, and it can be used to intercept the applications' requests, thus to realize the sensor-related behavior check. FireDroid [9] intercepted the system calls to identify if an application is executing dangerous actions at runtime. DeepDroid [10] dynamically hooked system processes in order to find details of applications' requests. However, the attackers can also apply the similar interception to bypass these hook-based solutions either by substituting the sensor-related system calls for its own implementation, or by breaking down them completely.

Trustzone-based solutions: Some researchers have proposed techniques based on Trustzone that support for isolated execution of secure applications and for secure peripherals [11], [12]. However, utilizing the above Trustzone-based techniques requires modifications to various parts of the system software and hence is not easily deployable by ordinary users on their mobile systems. Meanwhile, device and TEE vendors lock down the TEE in commercial phones

before they are shipped, so normal app developers (other than the vendors) can hardly changes to the code inside the secure world. By contrast, the MAGDEFENDER system is able to detect any operations of accessing to multimedia data directly from magnetometer readings, which is also easy deployable, *e.g.*installing an app. MAGDEFENDER can also detect the eavesdropping behaviors from the OS itself and the malicious apps in means of transplantation attacks.

B. Side-channel Sensing on Mobile Devices

Shmatikov [13] illustrated how the memory footprint left by browsers can be used for website fingerprinting. Several researchers have shown that power consumption traces [14] can be used to infer the opening of apps and websites. In [15], the authors designed learning systems to automatically fingerprint apps using encrypted network traffic. Note however that it is very difficult for third-party apps to acquire the system kernel data (*i.e.*, CPU/memory usage, power consumption) with high sampling rates ($\leq 20Hz$).

The EM side channel has been exploited for attacking electronic devices. [16], [17] created a novel near-field communication system between mobile devices. In [18], mobile devices were characterized based on their near-field EM signals. Several researches exploited the reaction of the built-in magnetometer to EM activity to infer apps and webpages opened on victim's laptop/phones [3], [19]. In this work, we prove that the built-in magnetometer readings accurately capture the EM emission signal generated by the app accessing the multimedia sensor data, and also show the feasibility of identifying the eavesdropping behaviours with the elaborate deep learning network.

VIII. CONCLUSION

In this paper, we proposed a novel EM-side-channel sensing technology-based model, MAGDEFENDER, that utilized the magnetometer built into mobile devices to detect eavesdropping via on-board mics and cameras. Various experiments have proved the performance of MAGDEFENDER, proving its effectiveness to detect the whole eavesdropping behaviors that access to the multimedia sensors (*i.e.*, cameras and/or microphones) on the mobile devices.

ACKNOWLEDGEMENT

This work is supported by NSFC (62072306 and 61936015) and Program of Shanghai Academic Research Leader (20XD1402100).

REFERENCES

[1] S. Maheshwari, "That game on your phone may be tracking what you're watching on tv," https://www.nytimes.com/2017/12/28/business/media/alphonso-app-tracking.html, 2017.

[2] Z. Zhang, P. Liu, J. Xiang, J. Jing, and L. Lei, "How your phone camera can be used to stealthily spy on you: Transplantation attacks against android camera service," in *ACM CODASPY*, 2015.

[3] Y. Cheng, X. Ji, W. Xu, H. Pan, Z. Zhu, C. You, Y. Chen, and L. Qiu, "Magattack: Guessing application launching and operation via smartphone," *ACM AsiaCCS*, 2019.

[4] P. Umbelino, "How attackers could hijack your android camera to spy on you." https://www.checkmarx.com/blog/how-attackers-could-hijack-your-android-camera/, 2019.

[5] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *IEEE CVPR*, 2017.

[6] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein gan," *arXiv preprint arXiv:1701.07875*, 2017.

[7] A. Sobral, T. Bouwmans, and E.-h. Zahzah, "Lrslibrary: Low-rank and sparse tools for background modeling and subtraction in videos," in *Robust Low-Rank and Sparse Matrix Decomposition: Applications in Image and Video Processing*. CRC Press, Taylor and Francis Group., 2015.

[8] T. Mehta, "Access dots, ios 14 cam/mic access indicators!" https://play.google.com/store/apps/details?id=you.in.spark.access.dots, 2020.

[9] G. Russello, A. B. Jimenez, H. Naderi, and W. van der Mark, "Firedroid: Hardening security in almost-stock android," in *IEEE ACSAC*, 2013.

[10] X. Wang, K. Sun, Y. Wang, and J. Jing, "Deepdroid: Dynamically enforcing enterprise policy on android devices." in *NDSS*, 2015.

[11] S. Mirzamohammadi and A. A. Sani, "Viola: Trustworthy sensor notifications for enhanced privacy on mobile systems," *IEEE TMC*, 2018.

[12] M. Lentz, R. Sen, P. Druschel, and B. Bhattacharjee, "Secloak: Arm trustzone-based mobile peripheral control," in *ACM MobiSys*, 2018.

[13] S. Jana and V. Shmatikov, "Memento: Learning secrets from process footprints," 2012.

[14] Y. Chen, X. Jin, J. Sun, R. Zhang, and Y. Zhang, "Powerful: Mobile app fingerprinting via power analysis," *IEEE INFOCOM*, 2017.

[15] V. F. Taylor, R. Spolaor, M. Conti, and I. Martinovic, "Robust smartphone app identification via encrypted network traffic analysis," *IEEE TIFS*, 2018.

[16] H. Pan, Y.-C. Chen, G. Xue, and X. Ji, "Magnecomm: Magnetometer-based near-field communication," in *ACM MobiCom*, 2017.

[17] G. Xue, H. Pan, Y.-C. Chen, X. Ji, and J. Yu, "Magnecomm+: Near-field electromagnetic induction communication with magnetometer," *IEEE Transactions on Mobile Computing*, 2021.

[18] E. J. Wang, T.-J. Lee, A. Mariakakis, M. Goel, S. Gupta, and S. N. Patel, "Magnifisense: Inferring device interaction using wrist-worn passive magneto-inductive sensors," in *ACM UbiComp*, 2015.

[19] H. Pan, L. Yang, H. Li, C.-W. You, X. Ji, Y.-C. Chen, Z. Hu, and G. Xue, "Magthief: Stealing private app usage data on mobile devices via built-in magnetometer," in *2021 18th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*. IEEE, 2021, pp. 1–9.