Feature Reconstruction: Far Field EM Side-Channel Attacks in Complex Environment

Huanyu Wang, Dalin He, Deng Tuo, and Junnian Wang

***Abstract*—Far Field EM Side-Channel Attacks (FEM-SCAs) have emerged as a realistic security threat to widely deployed RF-integrated IoT edge devices. In mixed-signal chips, side-channel leakage may unintentionally couple with transmission signals and be emitted via the on-chip antenna, potentially allowing adversaries to extract sensitive information from the victim at long distances. However, in practical scenarios, far field EM traces captured at long distances usually suffer from noise and interference, which makes the attack less efficient or sometimes even unfeasible. In this paper, we propose a Domain-Adversarial ReFeature Nueral Network (DAR-NN) to facilitate "noisy-clean" adaptation for far field EM traces captured at long distances. By integrating a DAE model with two deep-learning classifiers as regularization terms, the proposed DAR-NN model can reconstruct features of traces obtained remotely in complex environments, thereby achieving a more efficient FEM-SCA. We first test our model by using a publicly available dataset and show that it is feasible to extract the AES key from 141 traces captured at 15 m distance to the victim, which is 58.7% more efficient than existing methods with 80% less profiling data. Afterwards, we set up a more complex experimental environment with a HackRF radio serving as an interference source. We show that the proposed model can still extract the key by using around 2K traces at 15 m even in the presence of 25% active interference, while the state-of-the-art model fails under same conditions.**

***Index Terms*—Far-field EM side-channel attack, complex environment, domain-adversarial neural network, feature reconstruction, AES.**

I. Introduction

INTERNET of Things (IoT) edge devices are revolutionizing our society by facilitating connections ubiquitously. However, their widespread deployment also makes them attractive victims for adversaries' malicious activities. Among various threats, Side-Channel Attacks (SCAs) have emerged as an effective way to compromise physical implementations of cryptographic algorithms. In contrast to traditional cryptoanalysis, SCAs aim to bypass the inherent theoretical strength of algorithm designs and extract the sensitive information from their physical implementations. At present, different types of side channels have been successfully exploited to compromise cryptographic implementations [1], [2], reverse neural networks' architecture [3], [4], monitor user-browser information [5], [6], steal the password from the keystroke [7], [8] and track the code execution [9]. However, most commonly exploited SCAs require close physical proximity to the victim, which may lead to the attack less practical in numerous scenarios.

In 2018, a new type of side channel has been discovered, called far field electromagnetic (EM) emanations or **screaming channels** [10]. In the novel Far Field EM Side-Channel Attacks (FEM-SCAs), by exploring the coupling effect between EM emissions from the CPU core and radio carrier signals on mixed-signal chips, adversaries are feasible to eliminate the requirement of the direct physical proximity to the victim device, which makes the attack increasingly convert. This method of attack capitalizes on radio transmitters integrated into mixed-signal chips, permitting sensitive data to inadvertently modulate radio frequency signals, which are subsequently disseminated through the airwaves. In [10], the secret key of a Bluetooth device implementation of AES is recovered from 1.4K traces captured at 10 m distance (in an anechoic chamber), with each trace derived from the average of 500 measurements of the same encryption. In the follow up template attack [11], the key is recovered from 5K traces captured at 15 m with each trace derived from the average of 1K measurements of the same encryption, while the key enumeration is still up to 2^{23}. Afterwards, with the help of deep-learning techniques, [12] successfully extracts the AES key from 10K traces without repetitions at 15 m distance to the victim device. In 2021, [13] goes one step further to use "clean" traces captured via a coaxial cable with controlled amount of additive noise for constructing neural networks. This approach enhances the model's generalization capacity and achieves to extract the key from around 350 traces captured at 15 m distance without repeating each encryption more than once under the same conditions. Next, Zhang et al. [14] employed a wireless repeater positioned between the adversary and the victim to amplify the captured signals. However, this method is highly depends on the strategic placement of the repeater, a factor that can potentially compromise the attacker's ability to remain undetected. In 2023, their subsequent work [15] proposes a multi-spectrum capturing method to acquire traces with an enhanced Signal-to-Noise Ratio (SNR). Table I summarizes existing FEM-SCAs on nRF52 SoC implementations of AES.

However, traces captured at long distances often encounter substantial interference and noise, making them challenging for recognition by deep-learning models and leading to a marked decrease in the efficiency of the attack. In [13], it is shown that when traces are captured at 15 m distance with

Received 18 February 2025; revised 14 July 2025 and 18 August 2025; accepted 15 September 2025. Date of publication 18 September 2025; date of current version 30 September 2025. This work was supported by the Key Research and Development Program of Hunan Province, Department of Science and Technology of Hunan Province, through the Research Grant 2025AQ2024. The associate editor coordinating the review of this article and approving it for publication was Prof. Fengwei Zhang. *(Corresponding author: Junnian Wang.)*

Huanyu Wang is with the School of Computer Science and Engineering, Hunan University of Science and Technology, Xiangtan 411199, China.

Dalin He, Deng Tuo, and Junnian Wang are with the School of Physics and Electronic Science, Hunan University of Science and Technology, Xiangtan 411199, China (e-mail: jnwang@hnust.edu.cn).

Digital Object Identifier 10.1109/TIFS.2025.3611788

TABLE I
SUMMARY OF EXISTING FEM-SCAS ON NRF52 SOC IMPLEMENTATIONS OF AES

Existing works	Target	# Traces to Extract the Key	Distance	# Encryption Repetitions	# Profiling Data	# Key Enumeration	Experimental Environment	Year
[10]	TinyAES	1.4K	10m	500	130K × 500	2^0	Anechoic chamber	2018
[11]	TinyAES	5K	15 m	1K	10K × 500	2^{23}	Office without interference	2020
[12]	TinyAES	10K	15 m	1	200K × 100	2^0	Office without interference	2020
[13]	TinyAES	341	15 m	1	200K × 100	2^0	Office without interference	2021
[14]	TinyAES	6K	0.7 m	1	-	2^0	Office without interference	2022
[15]	TinyAES	2K	1 m	1	-	2^0	Office without interference	2023
[20]	AES-128	15K	7 m	10	150K × 10	2^{32}	Office without interference	2023
[21]	RP-Masked AES	300K	0.2 m	1	100K × 100	2^0	Office without interference	2024
[22]	AES Accelerator	90K	1 m	10K	300K × 10K	2^0	Office without interference	2025
This work	TinyAES	141	15 m	1	40K × 100	2^0	Office without interference	2025
This work	TinyAES	2K	15 m	1	40K × 100	2^0	Office with interference	2025

each trace derived from the average of 100 measurements of the same encryption (100 repetitions), the Convolutional Neural Network (CNN) model can recover the key by using < 10 traces. In practical scenarios, adversaries can only capture traces without repeating each encryption more than once. When it comes to the case where the traces are without any repetitions, a successful attack in [13] requires > 340 traces due to the presence of environmental noise. This highlights that existing profiling strategies may still be far from the optimal approach for attacks in real-world scenarios. Furthermore, current research on this topic has predominantly focused on conducting attacks within controlled environments such as anechoic chambers or simple office corridors, without considering the potential presence of some interference sources, particularly running at the capturing center frequency. This is definitely a more complex and realistic condition. From 2021 to 2024, there are several works [16], [17], [18] focus on utilizing Denoising Autoencoder (DAE) models to reduce the noise effect in power and near field EM traces. However, power and near field EM traces are usually contain much more features of the key-dependent sensitive data, making the DAE model inherently easier to retain essential features for subsequent classification. In the context of FEM-SCAs, their DAE models tend to diminish key-dependent features to undetectable levels in their efforts to reduce noise. This is primarily due to the inherently minimal characteristics of features in far-field EM traces (as shown in the following experiments), a challenge that intensifies when these traces are captured over long distances or within complex environments.

To solve this problem, a potential solution is to build the DAE filter for mapping "noisy" traces to "clean" traces in a Domain-Adversarial Neural Network (DANN) model [19]. A DANN is a type of deep neural network designed to enhance the generalization capacity in unseen domains through the adversarial task between the feature extractor and the domain discriminator. This is particularly promising in FEM-SCAs where the "clean" training traces (source domain) might not perfectly represent the "noisy" victim traces (target domain).

Contributions. To reconstruct key-dependent features from noisy and distorted far-field EM traces, which are captured over long distances or within complex environments, we propose a new DANN scheme called Domain-Adversarial ReFeature Nueral Network (DAR-NN). By collaboratively utilizing two other deep-learning classifiers as regularization terms of the DAE model, it aims to rebuild key-dependent features for further attacks, thereby achieving a more efficient screaming channel attack.

We first show that in a simple office corridor environment, the proposed DAR-NN model achieves to increase the leakage level of traces captured at 15 m distance to an nRF52 Bluetooth device implementation of AES. With the help of the DAE filter, the CNN classifier is feasible to further recover the AES key by using only **141 traces** at 15 m, which is 58.7% more efficient than the same classifier in [13], while simultaneously requiring 80% less profiling data.

Afterwards, we test our model in a more complex experimental environment with an interference radio source activated at the capturing radio frequency. We show that the proposed DAR-NN model can still recover the subkey by using around 2K traces at 15 m distance, while the baseline models fail to conduct the attack by utilizing 5K traces without key enumerations.

Next, we evaluate the transferability of the proposed model using power traces captured from the STM32 and ATXmega MCU implementations of AES. The results demonstrate that the model effectively mitigates the impact of noise and interference across different side channels and platforms, confirming its adaptability.

II. BACKGROUND

This section begins with an overview of how EM emanations from the CPU coupled with radio frequency signals can be utilized as side channels. Afterwards, we provide an introduction of Deep-Learning Side-Channel Attacks (DLSCAs), including a review of the current state of research in this field. Subsequently, this section explores relevant background information on DANN and DAE.

A. Far-Field EM Side Channel

A mixed-signal circuit, also known as Radio Frequency Integrated Circuits (RFICs), combines both analog and digital components within a single integrated circuit. These circuits typically require fewer materials and occupy less space, widely employed in diverse electronic applications.

However, the co-integration of digital and analog circuits within the same silicon die presents a challenge in managing noise generated during digital instruction executions.

The Radio Frequency (RF) block within the analog part demonstrates considerable sensitivity to this noise. The CPU core within the digital circuit of the mixed-signal chip generates square wave noise due to the frequent switching of the clock signal. The side-channel leakage during the execution of software implementations of cryptographic algorithms may get modulation by this square wave. This modulated signal, carrying side-channel information, can mix with the baseband signal of the Voltage-Controlled Oscillator (VCO) in the analog circuit, a process facilitated by substrate coupling. The RF block and VCO then further modulate the received signal to a high frequency as defined by the wireless transmission protocol, transmitting it using the on-chip antenna. This unintended transmission of leakage along the wireless channel (with shifted bandwidth center frequency) could be detected by adversaries at a long distance. Subsequently, adversaries can extract the sensitive data of mixed-signal devices by analyzing traces captured from the wireless channel without having physical access to the victim. In addition to targeting implementations of unprotected AES as described above, FEM-SCAs have also been applied to compromise masked AES [21] and post-quantem cryptographic implementations [23], as well as to analyze re-keying protocol of Bluetooth [24].

B. Deep Learning Side-Channel Attack

Over the past two decades, SCA techniques have experienced a considerable evolution as illustrated in [25]. Initially emerging as straightforward techniques such as Correlation Power Analysis (CPA) [26] and Algebraic Side-Channel Analysis (ASCA) [27], they have progressed alongside technological advancements to encompass sophisticated methods such as Template Attacks (TAs) [28] and deep learning-based approaches. Deep Learning (DL) techniques have become exceedingly popular for their exceptional ability to identify complex patterns and make accurate classifications from extensive datasets. Generally, a well-trained deep-learning model can make the attack several orders magnitude more efficient than conventional signal processing methods [29], [30], [31]. Furthermore, various neural networks, including CNNs, are increasingly being utilized by adversaries to overcome countermeasures against SCAs [32], [33]. In [32], the CNN-based strategy markedly simplifies the attack process, removing the necessity for trace alignment and the identification of Points of Interest (PoIs). Afterwards, [34] investigates the impact of various neural network hyperparameters on DLSCAs, introducing the widely recognized benchmark, the ASCAD dataset. In the following phase, more and more advanced DL techniques are applied in DLSCAs for different attack scenarios. For example, [35] proposed a DLSCA framework based on transfer learning to adapt the DL classifiers from profiling devices to a new target device. Afterwards, neural networks are employed to augment template attacks, resulting in a model with fewer hyperparameters to tune, thereby simplifying its usability [36]. Deep neural networks have even been successfully employed to compromise the FPGA implementations of masked Kyber-512 as demonstrated in [37]. Excluding certain exceptions [38], deep-learning techniques are predominantly employed in profiled SCAs, which typically involve two stages: profiling and attack.

During the profiling stage, adversaries are often presumed to have full control over one or more profiling device(s), identical to the victim's and running the same cryptographic algorithm. This allows the attacker to capture numerous side-channel traces and associated information. A chosen deep-learning model is trained to establish a leakage profile linking the secret-dependent values to the traces.

During the attack stage, adversaries could obtain a small amount of side-channel traces and use the pre-trained model to deduce the secret. In FEM-SCAs, the requirement for physical access is waived as the attacker can get traces remotely.

C. DAE and DANN

The goal of an autoencoder is to minimize the difference between the input and the decoded output, learning to find features of the data. However, the uniqueness of a DAE model lies in training the model by intentionally introducing noise to the input data. In a DAE model, the encoder's role is to map the noise added data to the latent representation space, while the decoder aims to restore the original data without noise. By compelling the model to handle noisy data, the model may learn a special non-linear mapping profile from noise-added data to clean data. Consequently, the model tends to eliminate noise and redundant information, extracting features of the data. Since 2019, DAE models have been introduced to reduce the noise levels in power and near-field EM traces. Yang et al. [39] proposes a Convolutional Denoising Autoencoder (CDAE) that can learn a distinct non-linear mapping between traces during the profiling phase. At the attack stage, the methodology outlined by [39] involves first using the trained CDAE to obtain new traces, followed by analyzing these refined traces to extract the secret key by employing template attacks. Afterwards, [40] introduces another CDAE model which aims to bypass certain countermeasures of AES implementations by using the near field EM emanations as the side channel. In 2023, [18] proposes a conditional variational autoencoder to bridge DL and SCA paradigms based on theoretical results provided by stochastic attacks. By following the path of [17], [40] further designs a multi-loss DAE model which makes the power and near field EM based SCAs more efficient. In 2024, [41] proposes a Leakage Distillation-based Profiling Attack (LD-PA) approach which uses an encoder to extract multivariate leakage from raw traces and transform it into an effective representation.

However, existing DAE-based SCAs are focusing on power or near field EM SCAs, in which traces are observed to encompass a broader array of features pertaining to key-dependent sensitive data than far-field EM traces. Within the scope of FEM-SCAs, DAE models often encounter difficulties, as they tend to reduce key features to undetectable levels while attempting to minimize noise. This challenge is further exacerbated when such traces are captured in complex environments. To address this issue, combining DAE with DANN models might be a good way to go. The adversarial process in a DANN [19] is a critical component that aims to produce domain-invariant features to enhance the model's performance on unseen domains. This process involves a dual-objective optimization strategy within the network architecture, incorporating both a primary task classifier and a domain discriminator. To enforce the learning of

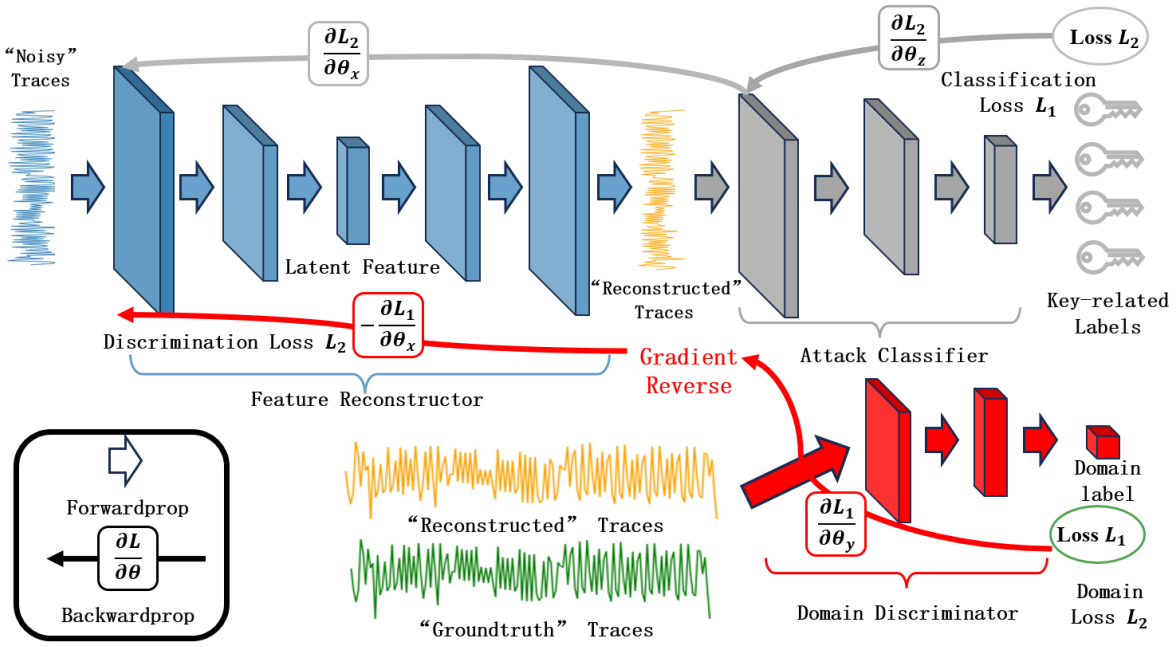


Fig. 1. The profiling stage of the DAR-NN model.

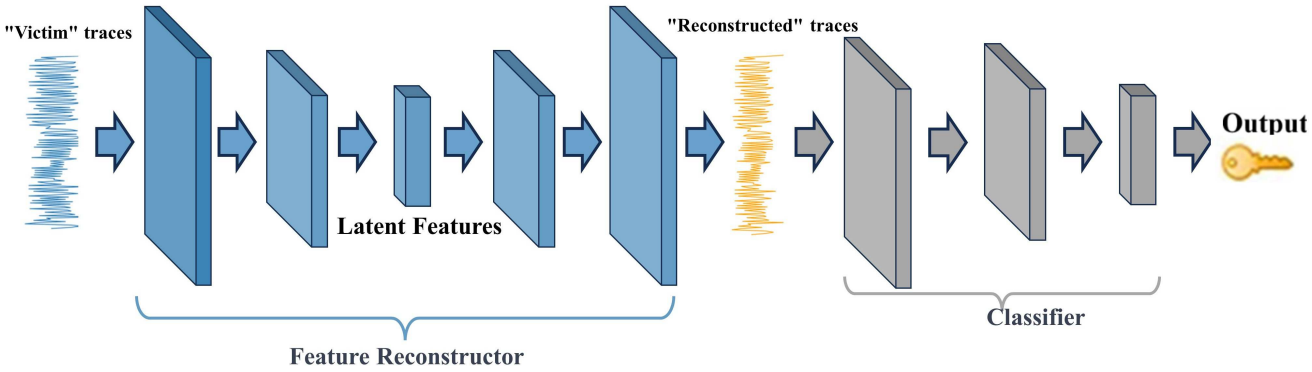


Fig. 2. The attack stage of the DAR-NN model.

domain-invariant features, a gradient reversal layer is integrated between the feature extractor and the domain discriminator. During the backpropagation phase, this layer inverts the gradient signals from the domain discriminator. Consequently, while the domain discriminator is trained to maximize its ability to distinguish between the domains, the feature extractor receives reversed gradient signals encouraging it to minimize this distinction. As a result, the feature extractor is optimized to deceive the domain discriminator by producing features that are indistinguishable between domains. This adversarial process thereby promotes the development of features that are both useful for the primary task and robust against domain variability.

In the context of side-channel attacks, several works show great potential of DANNs, particularly in remote attacks where environmental conditions are unpredictable. In 2023, [7] and [42] employ domain-adversarial learning strategies to eliminate the domain-specific information for keystroke and speech eavasdropping, respectively. These illustrate that DANN framework is highly promising for mitigating the noise and interference in FEM-SCAs scenarios.

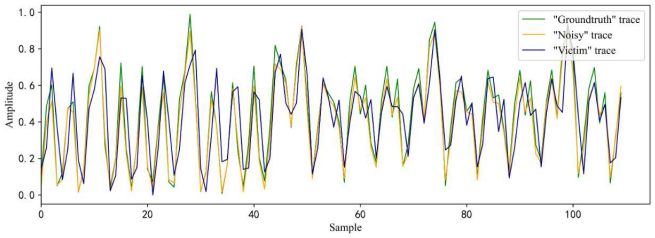


Fig. 3. The comparison of the plots of "groundtruth", "noisy" and "victim" traces (normalized).

III. Domain-Adversarial ReFeature Neural Network

In this section, we introduce how the proposed DAR-NN model works in FEM-SCAs to address the challenge described above and achieve a more efficient attack. Fig. 1 shows an overview of the training process of the proposed model and Fig. 2 illustrates how the trained model is used to attack.

From the current state of research [14], [15], it is evident that a significant challenge in FEM-SCAs is the presence of

substantial environmental noise and interference. These factors result in a relatively less efficient attack and profiling [13], as the captured noisy and affected traces exhibit a significant amount of feature distortion (as shown in the following experiments and in Fig. 3). The main idea behind the proposed DAR-NN model is to build the mapping from the "noisy" to "groundtruth" traces while using a CNN classifier pre-trained for the attack and a MLP discriminator as regularization terms. However, it is impractical to capture far-field EM traces truly "clean". Thus, we define the concept of "groundtruth", "noisy", "reconstructed" and "victim" traces as follows.

"Groundtruth" traces. We call traces captured by using a coaxial cable directly touched to the RF block of the profiling device with 100 repetitions as "groundtruth" traces. Each trace is the average of 100 measurements of the same encryption. Note that a coaxial cable can establish a direct connection between the transmitter and the radio to record signals without emitting them externally.

"Noisy" traces. We call traces captured from the profiling device by using a coaxial cable without any repetition as "noisy" traces. The study in [13] demonstrated that averaged trace derived from multiple measurements of the same encryption exhibits significantly higher leakage levels compared to traces without any repetitions.

"Reconstructed" traces. We call traces mapped from "noisy" traces by using the feature reconstructor of the DAR-NN model as "reconstructed" traces.

"Victim" traces. We call traces captured from the victim device at long distances as "victim" traces. Despite the impact caused by the noise and interference, the amplitude of the received signal is inherently proportional to the inverse of square of the distance. Fig. 3 shows comparison of the plots of normalized "groundtruth", "noisy" and "victim" traces.

The profiling stage of the DAR-NN model can be divided into three parts: reconstructor, discriminator and classifier.

A. Reconstructor

To build the mapping from captured traces to "groundtruth" traces, we first build a DAE model as the feature reconstructor by using the "noisy-groundtruth" trace pairs. For each pair, both traces are representing the same encryption (same key and plaintext). The left part of Fig. 1 illustrates the functioning of the reconstructor. It initially extracts key-dependent features from "noisy" traces and then rebuilds these features into "reconstructed" traces that are sufficiently close to "groundtruth" traces.

Let $\boldsymbol{\mathcal{T}} = \{\mathcal{T}_1, \ldots, \mathcal{T}_{|\boldsymbol{\mathcal{T}}|}\}$, where $\mathcal{T}_i \in \mathbb{R}^m$, for $i \in \{1, \ldots, |\boldsymbol{\mathcal{T}}|\}$, be a set of "noisy" traces representing the computation of the last round of AES which are captured from a profiling nRF52 SoC implementation of TinyAES by using the coaxial cable without any repetitions, with randomly generated plaintexts $\mathcal{P}_i \in \{0,1\}^{128}$ and a fixed key $K \in \{0,1\}^{128}$. For each trace, the reconstructor processes it through a series of neural layers and compresses the trace into the most important latent features. Afterwards, another set of layers, typically mirroring the encoder, is used to reconstruct the trace from the compressed form. We denote the "reconstructed" trace set generated by using the reconstructor as $\boldsymbol{\mathcal{T}'} = \mathcal{N}_x(\boldsymbol{\mathcal{T}}) = \{\mathcal{T'}_1, \ldots, \mathcal{T'}_{|\boldsymbol{\mathcal{T}}|}\}$, where $\mathcal{T'}_i \in \mathbb{R}^m$, for $i \in \{1, \ldots, |\boldsymbol{\mathcal{T}}|\}$.

Before the training of the entire DAR-NN model, we use Mean Square Error (MSE) to quantify the pre-loss of the reconstructor L_0 to represent the difference in shape between a "reconstructed" trace $\mathcal{T'}_i$ and the corresponding "groundtruth" trace $\mathcal{T}^*_i$.

$$L_0 = \frac{1}{n}\sum_{j=1}^{n}(\mathcal{T'}_{i,j} - \mathcal{T}^*_{i,j})^2, \tag{1}$$

where $\mathcal{T'}_{i,j}$ and $\mathcal{T}^*_{i,j}$ denote the jth data point in $\mathcal{T'}_i$ and $\mathcal{T}^*_i$, respectively, and n is the length of the traces. To minimize the loss, the gradient of L_0 is computed and back-propagated through the reconstructor before the training process of the entire DAR-NN model. Notice that this pre-process is not shown in Fig. 1.

B. Discriminator

As shown in the right bottom of Fig. 1, we further design a MLP model as the domain discriminator which aims to minimize the difference between "reconstructed" and "groundtruth" traces. For each "noisy" trace $\mathcal{T}_i$ in $\boldsymbol{\mathcal{T}}$, we first label it with the corresponding "groundtruth" trace $\mathcal{T}^*_i$. Each "groundtruth" trace $\mathcal{T}^*_i$ in $\boldsymbol{\mathcal{T}}^*$ is the average of 100 measurements captured by using the coaxial cable with the same encryption to degenerate the noise level. The coaxial cable efficiently transmits EM signals from the on-chip antenna, offering a low-loss and low-impedance path for high-frequency signals. This cable comprises a central conductor, an insulating layer, a braided shield, and an outer jacket. The braided shield not only minimizes Electromagnetic Interference (EMI), but also serves as a return path for the transmitted signal.

Afterwards, we label every "groundtruth" trace in $\boldsymbol{\mathcal{T}}'$ with the domain tag 0 and every "reconstructed" trace in $\boldsymbol{\mathcal{T}}^*$ with another domain tag 1. We denote the ith domain label as d_i for each trace i. The resulting labeled sets is used to train a MLP domain discriminator $\mathcal{N}_y : \mathbb{R}^m \rightarrow \mathbb{I}^2$ which maps each trace into a domain *score* vector $V_i \in \mathbb{I}^2$, whose elements v_i represent the probability of the domain of the ith trace. To quantify the discrimination error L_1, we use categorical cross-entropy loss as shown in the following formula:

$$L_1 = -\sum_i d_i \log(V_i) \tag{2}$$

To minimize the loss, the gradient of L_1 is computed and back-propagated through the discriminator to tune its internal parameters. This makes the discriminator enhance its capability to discriminate between "groundtruth" and "reconstructed" traces. Afterwards, we further reverse the gradient of L_1 by multiplying it by a negative scalar during the backpropagation through the DAE reconstructor. This procedure is adopted with the intention of refining the DAE model to such an extent that it can effectively deceive the discriminator. By doing this, we aim to maximize the reconstruction of key-dependentfeatures in "noisy" traces. In this adversarial process, the discriminator and the reconstructor are engaged in a form of game where the goal of the reconstructor is to generate traces that are indistinguishable from the actual "noisy" traces. At the same time, the discriminator tries to differentiate between the "groundtruth" traces and the "reconstructed" ones. In this setup, the DAE reconstructor gets better at generating features that look like the ones in "groundtruth" traces.

However, according to our experimental results, a DAE model lacking additional regularization term might encounter an average trap. In this scenario, the DAE model maps all "noisy" traces to an averaged "reconstructed" trace. Although this averaged "reconstructed" trace exhibits a relatively small categorical error and MSE distance to all "groundtruth" traces, it retains a small amount of key-dependent features. To address this, we further introduce a CNN classifier as a regularization term for the DAR-NN.

C. Classifier

Instead of distinguishing between real and generated data, the classifier in our DAR-NN model is designed to identify to which extent the "reconstructed" traces can be correctly classified to the corresponding key-sensitive intermediate value of AES.

Let $\mathcal{C}_i \in \{0,1\}^{128}$ be the ciphertext which is generated when the trace $\mathcal{T}_i$ is captured. We use $\mathcal{C}_{i,k}$ to denote the kth byte of the ciphertext $\mathcal{C}_i$, $k \in \{0,1,\ldots,15\}$. Let $\mathbb{I} = \{x \in \mathbb{R} \mid 0 \le x \le 1\}$. To train the CNN classifier for a subkey K_k, each "reconstructed" trace $\mathcal{T}'_i \in \mathcal{T}'$, $i \in \{1,\ldots,|\mathcal{T}|\}$, is assigned to another label, $y_k(\mathcal{T}_i)$, computed as

$$y_k(\mathcal{T}'_i) = \mathcal{C}_{i,k} \oplus RK10_k, \tag{3}$$

where $RK10_k$ is the kth byte of the 10th-round key $RK10$, which is derived from K.

The resulting labeled set of "reconstructed" traces is used to train the CNN classifier $\mathcal{N}_z : \mathbb{R}^m \to \mathbb{I}^{256}$ which maps each "reconstructed" trace $\mathcal{T}'_i \in \mathbb{R}^m$ into a *score* vector $S_{i,k} = \mathcal{N}_z(\mathcal{T}'_i) \in \mathbb{I}^{256}$, whose elements $s_{i,k,h}$ represent the probability that the *SBox* output in the last round is equal to $h \in \{0,1,\ldots,255\}$ when the kth byte of $\mathcal{C}_i$ is computed:

$$s_{i,k,h} = \Pr(\mathcal{C}_{i,k} \oplus RK10_k = h). \tag{4}$$

Each label $y_k(\mathcal{T}'_i)$ for the ith trace with kth byte of the subkey can be represented as a one-hot encoded *ground truth* vector $t_{i,k,h} \in \{0,1\}^{256}$ defined by:

$$t_{i,k,h} = \begin{cases} 1 \text{ if } h = y_k(\mathcal{T}'_i) \\ 0 \text{ otherwise,} \end{cases} \tag{5}$$

where $h \in [0,255]$. To quantify the classification error L_2 of the classifier, we use categorical cross-entropy loss as shown in the following formula:

$$L_2 = -\sum_{h=0}^{255} t_{i,k,h} \log \left(\frac{e^{s_{i,k,y_k(\mathcal{T}'_i)}}}{\sum_{h=0}^{255} e^{s_{i,k,h}}} \right) \tag{6}$$

As we have described above, by minimizing L_2, the classifier aims to ensure that the "reconstructed" traces retain more features related to sensitive information. To comprehensively consider the contributions of the different parts, the total loss L of the DAE reconstructor is a linear combination of the two:

$$L = (1-\beta)\, L_2 - \beta L_1, \tag{7}$$

where β is the **loss factor** for adjusting the contribution of elements in L. To minimize the loss, the gradient of L with respect to the score is computed and backpropagated through the reconstructor network to tune its internal parameters utilizing the Adam optimizer as shown in Fig. 1. For optimizing the domain discriminator and the classifier, L_1 and L_2 are used separately by the same Adam optimizer.

At the attack stage, the trained reconstructor in DAR-NN is used independently as the filter to map the captured "victim" traces to "reconstructed" traces. Next, the newly generated "reconstructed" traces are used to derive the secret key of the victim device by using the adapted classifier in DAR-NN (see Fig. 2).

IV. Experimental Setup

Fig. 4 shows the core experimental setup components used to conduct the attack. The target device is a Nordic Semiconductor nRF52832 development kit (version 3.0), which supports Bluetooth 5. To capture far-field EM side-channel signals at the radio frequency, we utilize an Ettus Research N210 USRP Software-Defined Radio (SDR) paired with an Ettus XBS-400 bandwidth transceiver and a TL-ANT2424B grid parabolic antenna with a 24 *dBi* gain. Fig. 5 shows how to setup the FEM-SCA remotely by using the components described.

A. Victim Device

At the transmitter side, the victim nRF52 SoC runs TinyAES encryption periodically, broadcasting the ciphertexts via its on-chip antenna at a 2 *Mbps* data transmission rate. The device is powered by an ARM Cortex M4 core operating at 64 *MHz* and incorporates a 2.4 *GHz* multi-protocol radio. TinyAES is a straightforward and lightweight version of the AES-128 algorithm. Suppose the clock signal $s_1(t)$ from the target device's CPU core is presented by formula 8

$$s_1(t) = \sum_{n=-\infty}^{+\infty} A_n e^{i2n\pi f_s t}, \tag{8}$$

in which the clock frequency of the square wave is denoted as f_1. Formula 9 denotes the corresponding Fourier transform $S_1(f)$ of $s_1(t)$, in which A_n is used to present the Fourier series coefficients and δ is for the impulse function. The even terms of the Fourier series are not exactly equal to zero since the square wave noise is not an ideal square wave.

$$S_1(f) = \sum_{n=-\infty}^{+\infty} A_n \delta(f - nf_s) \tag{9}$$

Formula 10 defines A_n, in which τ is the duty cycle of the square wave.

$$A_n = \frac{sin\, n\pi\tau}{n\tau} \tag{10}$$

We use $s_2(t)$ to denote the side-channel signal generated from the Crypto block during the execution of the cryptographic algorithm. In the digital part, the side-channel signal $s_2(t)$ is first modulated by the square wave of the clock signal $s_1(t)$. We use $s_3(t) = s_1(t) \cdot s_2(t)$ to denote the resulting signal. Formula 11 shows the resulting signal in frequency domain.

$$S_3(f) = S_2(f) * S_1(f) = \sum_{n=-\infty}^{+\infty} A_n S_2(f - nf_1) \tag{11}$$

Fig. 4. Core experimental setup components: target device (nRF52 SoC) and the far-field EM signal capture equipment (SDR, antenna, and bandwidth transceiver).

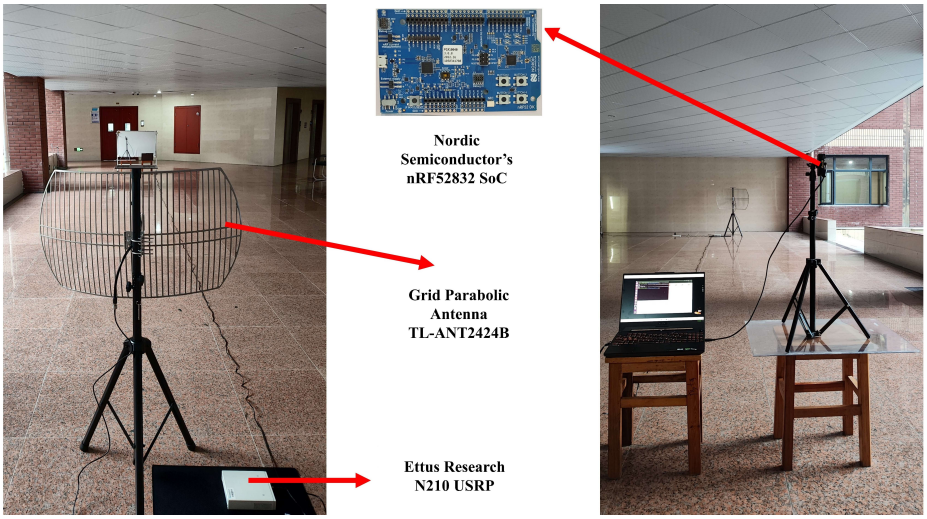


Fig. 5. Experimental setup.

Afterwards, in the analog part, the RF block modulates the received signal $s_3(t)$ to a radio frequency. Formula 12 presents the modulated signal in the time domain and frequency domain. We use $e^{i2\pi f_c t}$ to denote the radio carrier, in which f_c is the carrier frequency. After the modulation, the signal is first amplified and then transmitted through the antenna on chip.

$$s_4(t) = \sum_{n=-\infty}^{+\infty} A_n s_2(t) e^{i2\pi(nf_s+f_c)t}$$
$$S_4(f) = \sum_{n=-\infty}^{+\infty} A_n S_2(f - nf_s - f_c). \quad (12)$$

B. Capturing Device

At the receiver side as shown in the left part of Fig. 5, we utilize an Ettus Research N210 USRP SDR paired with a TL-ANT2424B grid parabolic antenna boasting a 24 *dBi* gain for capturing signals. When we set the center receiving frequency to $Nf_s \pm f_c$, the received signal in time and frequency domain ($(r(t))$ and $R(f)$) can be expressed by formula 13.

$$r(t) = \sum_{n\neq N} A_n s_2(t) e^{i2\pi(n-N)f_s t} + A_N s_2(t)$$
$$R(f) = \sum_{n\neq N} A_n S_2(f - (n-N)f_s) + A_N S_2(f). \quad (13)$$

By using a low pass filter with a threshold of 1.8 MHz, we can extract the side-channel signal $s_2(t)$ from the received signal $r(t)$. **We set the central receiving frequency to 2.272 *GHz*.** This frequency corresponds to the Bluetooth channel's center frequency $f_{\text{Bluetooth}} = 2.4$ GHz shifted by twice the target CPU's clock frequency $f_{clock} = 64MHz$ ($2.4\ GHz - 2 \times 0.068\ GHz = 2.272\ GHz$). In our experiments, we set the sampling frequency to 5 MHz, a value validated as sufficient for conducting the attack, as proved in [10], [11], [12], and [13].

Afterwards, we establish a Hack RF radio connected to a 3 *dBi* gain vertical antenna VERT2450, operating at 2.272 GHz, in the same experimental setup to serve as an interference source. We configure three different attack scenarios using the Hack RF radio as shown in Fig. 6 and 7. **Scenario 1 (no manual interference)**.

1) Aim: Test fundamental FEM-SCA efficiency limits under normal conditions.
2) Definition: Traces are directly captured at 15 m distance from the victim device, without the addition of any manual interference. Environmental noise is still present.

Scenario 2 (passive interference).

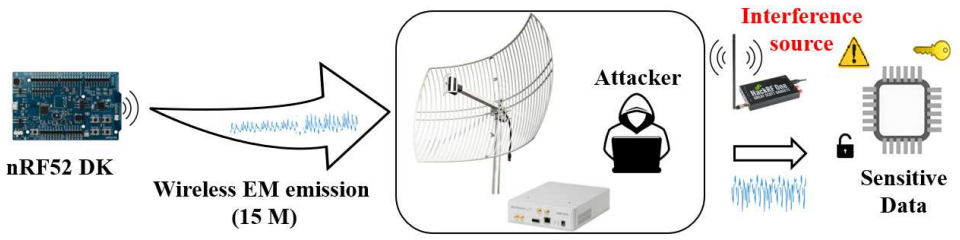


Fig. 6. Scenario: passive interference.

Fig. 7. Scenario: active interference.

1) Aim: Test the robustness of the DAR-NN based FEM-SCA against unintentional interference.
2) Definition: Interference originates from the backside of the receiver's parabolic antenna, simulating a situation where there is a passive interference source which is not intended for the protection (Fig. 6).

Scenario 3 (active interference).

1) Aim: Test the robustness of the DAR-NN based FEM-SCA against a potential countermeasure.
2) Definition: Interference originates from the same position as the victim device, simulating an active interference source as the countermeasure (Fig. 7).

C. Trace Synchronization

We apply the template-correlation method for synchronizing and segmenting traces from long captured FEM side-channel signal $s_2(t)$. Here is a step-by-step description of the template-correlation method, as illustrated in Fig. 8.

1) Signal Acquisition. The adversary first captures the raw FEM side-channel signals emitted by the target device by using the SDR.
2) Template Creation. Afterwards, we manually extract a representative segment from the captured signal to create a reference template, which embodies the expected side-channel features (e.g., 10th round *SBox* operations).
3) Pearson Correlation. The template is cross-correlated with the entire captured signal using Pearson correlation. This generates a correlation curve in which peaks indicate positions where the signal closely matches the template.
4) Automatic Cutting. Based on the correlation peaks, the raw signal is automatically segmented into individual, synchronized traces. Each trace corresponds to an instance of the targeted operations. The output is a time-aligned trace set, which is ready for further analysis.

Using the template-correlation method described above, we can successfully synchronize trace segments captured at different distances. Fig. 9 shows two segmented traces, one captured using the coaxial cable and the other at a distance of 15 m. From Fig. 9, we can find that even the traces share similar features for the same execution, the signal strength varies a lot. Therefore, in the following experiments, all segmented traces are pre-processed using the max-min normalization method, which is used to map the trace amplitude values into [0,1].

D. Dataset

In the following experiments, we use three datasets to represent different attack scenarios.

The first one is a publicly available dataset [13] captured at 15 m distance in an office corridor environment from the same nRF52 DK by using the same measuring equipment as shown in Fig. 5. All other settings remain consistent with those described above. Utilizing the public dataset facilitates a direct comparison between the proposed DAR-NN method and other existing approaches. The dataset includes a testing set containing 25K traces captured from 5 nRF52 DK boards. Each board contributes 5K traces to the dataset. We denote these 5 devices from D1 - D5. Each trace contains 400 data points, representing MixColumns and AddRoundKey operations of the 9th round and the complete last round of TinyAES. We use this dataset to simulate the scenario 1 (no interference) as defined. The second dataset simulates a scenario with passive interference. Initially, we capture interference traces originating from the radio source behind the receiver's parabolic antenna. Subsequently, these interference traces are added the first dataset, which are captured at 15 m distance. The third scenario simulates active interference, where the interference signals originate from the same position as the victim device. Similarly, we incorporate these interference traces into the first dataset to construct the new scenario.

Afterwards, we capture a training set from the profiling device D0 via a coaxial cable. The set contains 40K traces in total, each associated with its corresponding ciphertext and key. In addition, to generate the "groundtruth" traces for the training set, we further capture additional traces by repeating the same encryption 100 times and averaging out the resulting

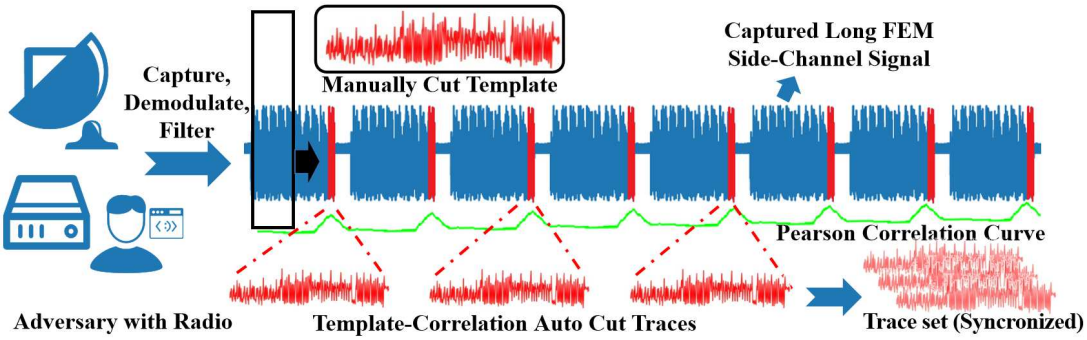


Fig. 8. Illustration of the template-correlation synchronization method.

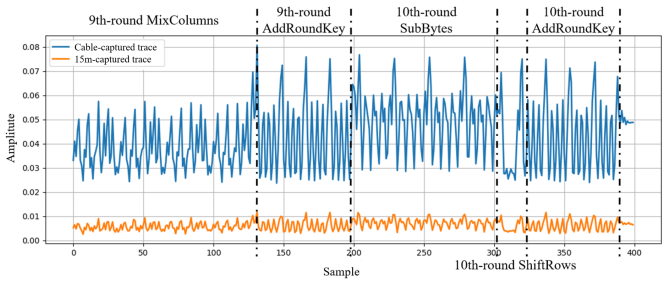


Fig. 9. The comparison of two segmented traces, one captured using the coaxial cable and the other at a distance of 15 m.

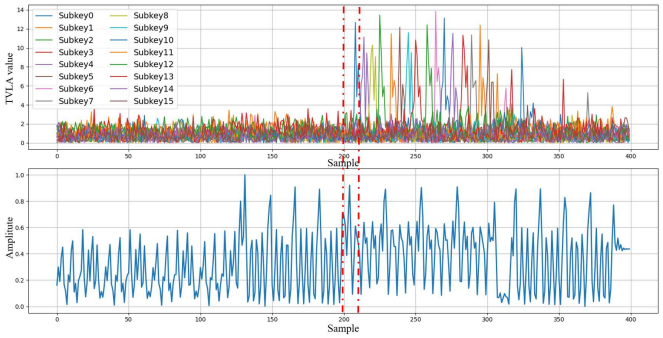


Fig. 10. T-test results of all 10th round subkeys on 5K "groundtruth" traces and an example trace (normalized).

traces. Consequently, each "groundtruth" trace represents the mean of 100 identical encryptions (the same plaintext and key). Fig. 9 plots a "groundtruth" trace captured by using the coaxial cable and a "victim" trace captured at 15 m distance to the victim device. All traces are denoted by using the dashed black line to illustrate the corresponding execution.

Attack Point. According to the architecture of the nRF52 DK, which incorporates a 32-bit ARM Cortex-M4 core, the chosen attack point is the **last-round *SBox* output** for two reasons. First, in software implementations of AES, the 8-bit output of the *SubBytes* operation is typically loaded from memory to a data bus, resulting in a more proportional and distinguishable leakage in side-channel traces compared to other intermediate operations. Second, in the FEM-SCA threat scenario, the adversary waives the requirement of physical access to the victim device, in which the ciphertext instead of the plaintext becomes a better choice as the known information for the attack, which makes the last round becomes the most viable target. This decision aligns with the methodology employed in [13] and [43].

E. Evaluation Metrics

We use the Test Vector Leakage Assessment (TVLA) method to detect the leakage and utilize the Partial Guessing Entropy (PGE) to assess the attack efficiency.

1) TVLA: In a TVLA (based on Welch's t-test), traces are divided into two groupsaccording to the selected attack point. The first group $\mathcal{T}_0$ contains traces ith $HW(label) > \beta$, while another $\mathcal{T}_1$ includes those with $HW(label) < \beta$. Here $HW(label)$ signifies the Hamming weight of the value of the attack point *label*. In TVLA, the Second-Order Statistical Test (SOST) is employed to evaluate the difference between the two trace groups, which is defined by formula 14.

$$SOST = \left(\frac{\mu_0 - \mu_1}{\sqrt{\frac{\sigma_0^2}{n_0} + \frac{\sigma_1^2}{n_1}}} \right)^2 \tag{14}$$

where μ_i and σ_i represent the mean and standard deviation of the set $\mathcal{T}_i$, respectively, while N_i denotes the number of data within the set.

2) PGE: The PGE [44] is for evaluating the effectiveness of subkey recovery in SCAs. It quantifies the degree to which a subkey can be deduced from a set of traces. It is commonly used as an indicator of the attack complexity, where lower entropy values are associated with an increased likelihood of the attacker's success. The *rank* of a subkey $\mathcal{K}_j$, denoted as $Rank(\mathcal{K}_j)$, signifies the count of candidates having a higher probability than the actual value of $\mathcal{K}_j$, within a trace set $\mathcal{T}$ along with the corresponding known information $\mathcal{X}$.

$$\begin{aligned} Rank(\mathcal{K}_j, \mathcal{T}) = |\{\mathcal{K}'_j \in \mathcal{K} : Pr[\mathcal{K}_j|\mathcal{X}, \mathcal{T}] \\ < Pr[\mathcal{K}'_j|\mathcal{X}, \mathcal{T}]\}|. \end{aligned} \tag{15}$$

Generally, we utilize the trained model $\mathcal{M}_j$ to categorize d sets of traces $\mathcal{T}_1, \ldots, \mathcal{T}_d$. The estimated PGE for the correct subkey is then computed as the average rank outcome across these sets.

$$PGE(\mathcal{K}_j) = \frac{\sum_{n=1}^{d} Rank(\mathcal{K}_j, \mathcal{T}_n)}{d} \tag{16}$$

In formula 16, for any trace set $\mathcal{T}_n$, the rank for $\mathcal{K}_j$ is denoted as $Rank(\mathcal{K}_j, \mathcal{T}_n)$.

TABLE II
THE ARCHITECTURE OF THE DAR-NN110 MODEL

(a) Reconstructor.

Layer (Type)	Output
Input (Dense)	(110, 1)
Conv 1 (Conv1D)	(110, 64)
AvgPooling 1	(55, 64)
Conv 2 (Conv1D)	(55, 32)
AvgPooling 2	(28, 32)
Input (Dense)	(28, 32)
Conv 3 (Conv1D)	(28, 32)
UpSampling1D 1	(56, 32)
Conv 4 (Conv1D)	(56, 64)
UpSampling1D 2	(112, 64)
Conv 5 (Conv1D)	(112, 1)
Flatten 1 (Flatten)	(112)
Output (Dense)	(110)
Total Parameters: 28,367	

(b) Discrimitor.

Layer (Type)	Output
Input (Batch_Norm)	(110)
Dense1 (Dense)	(64)
Dense2 (Dense)	(32)
Dense3 (Dense)	(16)
Dense4 (Dense)	(8)
Dense5 (Dense)	(4)
Output (Dense)	(2)
Total Parameters: 10,334	

(c) Classifier.

Layer (Type)	Output
Input (Dense)	(110, 1)
Conv 1 (Conv1D)	(110, 4)
AvgPooling 1	(109, 4)
Conv 2 (Conv1D)	(109, 8)
AvgPooling 2	(108, 8)
Conv 3 (Conv1D)	(108, 16)
AvgPooling 3	(107, 16)
Conv 4 (Conv1D)	(107, 32)
AvgPooling 4	(106, 32)
Flatten 1 (Flatten)	(3392)
Dense1 (Dense)	(200)
Dense2 (Dense)	(200)
Output (Dense)	(256)
Total Parameters: 772,344	

F. Model Structure

Layer structures of the DAR-NN110 model is shown in Table IIa and Table IIc, in which is dedicated for the trace segment with 110 data points. Table IIa shows the architecture of the reconstructor of the DAR-NN110 model, while Table IIb and Table IIc are for the discriminator and classifier, respectively. The DAR-NN110 model focuses on trace interval which representing the 16 *SBox* operations, which is as the same trace interval used in [13] for comparison. To minimize the loss outlined in Section III, the network's parameters are adjusted by computing the loss gradient and applying backpropagation. This adjustment is executed using the Adam optimizer, set at a learning rate of 0.0005 without any decay. We train the model on 40K traces from the training set over 200 epochs with batch size configured at 128. The architecture of the classifier in the DAR-NN110 model is set identically to that in [13] for further comparison. The layer structures of the DAR-NN20 model are illustrated in Table IIa, Table IIb and Table IIc. The reconstructor uses Conv1D layers with kernel size 1, stride 1, and "same" padding, while the classifier is with kernel size 3, stride 2, and "valid" padding. The DAR-NN20 model focuses on trace interval which representing the first *SBox* operation. The 20-point leakage interval is derived from the TVLA result conducted on 25K "groundtruth" traces. For analytical purposes, the traces are divided into two categories based on the Hamming weight of the attack point. The top plot in Fig. 10 shows the TVLA result, and the bottom plot is a "groundtruth" trace. The leakage interval for training the DAR-NN20 model is marked by the dashed red line. All experiments are conducted on a desktop system equipped with an Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz processor and an NVIDIA GeForce GTX 1650 GPU.

G. Baselines

In our experiments, we compare the proposed DAR-NN model with the following five recent DLSCA advances.

1. **Switch-T Transformer** [45]. It is a multi-task Transformer which employs the Elastic Weight Consolidation (EWC) mechanism to reduce catastrophic forgetting. In the following experiments, the model is built on 40K "groundtruth" traces.

2. **ML-DAE** [17]. The Multi-Loss Denoising AutoEncoder (ML-DAE) model is designed by combining three loss terms for better generalization. Together, we apply the same CNN classifier in [17] for further subkey recovery. In the following experiments, the ML-DAE model is trained on 40K "noisy-groundtruth" trace pairs.

3. **LC-CNN** [46]. A CNN model that optimizes Label Correlation (LC) for faster convergence. In the following, the LC-CNN model is built on 40K "groundtruth" traces.

4. **Meta-Transfer CNN** [35]. We select the meta-transfer learning strategy for comparison as it seems to be promising in the FEM-SCA case. The model built on this method can first learn features from "groundtruth" traces, then adapts to "noisy" traces. Therefore, the Meta-Transfer CNN model in the following experiments is trained on 40K "groundtruth" traces and then transferred on 40K "noisy" traces.

5. **SOTA CNN** [13]. This CNN model achieves the state-of-the-art (SOTA) FEM-SCA result at 15 m distance to the victim nRF52 implementation of AES, which is to use 341 traces to recover a 10th-round subkey. In the following experiments, the SOTA CNN is built on 40K "groundtruth" traces.

V. EXPERIMENTAL RESULTS

In this section, we evaluate the proposed DAR-NN model on the datasets representing three different attack scenarios.[1] For most of the experiments, we focus on recovering the subkey $RK10_0$ and the process for other subkeys is the same.

A. Scenario: No Manual Interference

In this experiment, we test the model on traces captured at 15 m distance to the victim device in an office corridor environment without the manual interference. We first evaluate how different loss factors (see Section III) in the reconstructor's loss L can affect the attack efficiency. Table IV shows the average number of traces necessary to recover the subkey (PGE=0) across models incorporating different loss factors. All results are derived from the mean of 100 individual tests. From Table IV, we find that when the loss factors β set around 0.5, the DAR-NN can achieve the best attack efficiency. In this case, the DAR-NN110 can recover the subkey at 15 distance with 220 traces, which is 35.5% more efficient than the SOTA record in [13]. Fig. 11 (a) shows the PGE result

[1]Our code and traces are publicly available at https://github.com/SCA-HNUST/DAR-NN-based-FEM-SCA

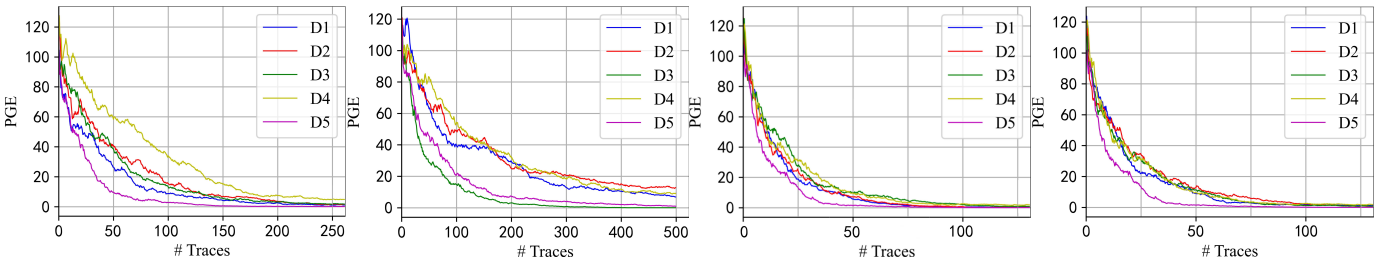


Fig. 11. Average PGE of different models tested on traces captured at 15 m distance to the victim without manual inteference. (a) DAR-NN110 model. (b) SOTA CNN model. (c) DAR-NN20 model. (d) DAR-NN20_200K model.

TABLE III
THE ARCHITECTURE OF THE DAR-NN20 MODEL

(a) Reconstructor.

Layer (Type)	Output
Input (Dense)	(20, 1)
Conv 1 (Conv1D)	(20, 32)
AvgPooling 1	(10, 32)
Conv 2 (Conv1D)	(10, 16)
AvgPooling 2	(5, 16)
Input (Dense)	(5, 16)
Conv 3 (Conv1D)	(5, 16)
UpSampling1D 1	(10, 16)
Conv 4 (Conv1D)	(10, 32)
UpSampling1D 2	(20, 32)
Conv 5 (Conv1D)	(20, 1)
Total Parameters: 4,129	

(b) Discrimitor.

Layer (Type)	Output
Input (Batch_Norm)	(20)
Dense1 (Dense)	(16)
Dense2 (Dense)	(8)
Dense3 (Dense)	(4)
Dense4 (Dense)	(2)
Output (Dense)	(2)
Total Parameters: 604	

(c) Classifier.

Layer (Type)	Output
Input (Dense)	(20, 1)
Conv 1 (Conv1D)	(18, 32)
AvgPooling 1	(9, 32)
Conv 2 (Conv1D)	(7, 64)
AvgPooling 2	(3, 64)
Flatten 1 (Flatten)	(192)
Dense1 (Dense)	(128)
Output (Dense)	(256)
Total Parameters: 64,064	

TABLE IV
ABLATION TESTS: AVERAGE NUMBER OF TRACES REQUIRED TO RECOVER THE TARGET SUBKEY AT 15 M BY USING THE DAR-NN110 MODEL ACROSS VICTIM DEVICES WITH DIFFERENT LOSS FACTORS (RESULTS FOR 100 TESTS)

β \ Device	D1	D2	D3	D4	D5	AVG
0.2	578	852	511	548	318	562
0.3	536	814	471	509	277	522
0.4	291	438	261	323	169	296
0.5	233	288	237	235	107	220
0.6	1231	1672	1476	2014	512	1381
0.7	2737	3801	3022	>4999	1998	/
0.8	>4999	>4999	>4999	>4999	>4999	/

TABLE V
AVERAGE NUMBER OF TRACES REQUIRED TO RECOVER THE TARGET SUBKEY AT 15 M BY USING THE DAR-NN110 MODEL WITH DIFFERENT INTERFERENCE RATIO IN PASSIVE AND ACTIVE SCENARIOS (RESULTS FOR 100 TESTS)

Interference ratio	2.5%	5.0%	12.5%	20.0%	25.0%
Passive interference	447	496	698	916	1305
Active interference	562	492	699	1321	1995

TABLE VI
AVERAGE NUMBER OF TRACES REQUIRED BY MODELS TO RECOVER THE TARGET SUBKEY (PGE=0) AT 15 M DISTANCE WITH NO MANUAL INTERFERENCE (RESULTS FOR 100 TESTS)

Victim device	D1	D2	D3	D4	D5	AVG
DAR-NN20	123	154	131	169	83	**141**
Switch-T Transformer [45]	2409	3063	3918	3358	2134	2976
ML-DAE [17]	785	1043	487	703	316	667
LC-CNN [46]	1667	1358	3349	>4999	1138	/
Meta-Transfer CNN [35]	>4999	3518	1365	>4999	1485	/
SOTA CNN [13]	292	467	256	384	306	341

on 15 m traces of the DAR-NN110 model with β set to 0.5. Therefore, the loss factors β is set to 0.5 in our following experiments.

Afterwards, we further combine the reconstructor of the DAR-NN110 model and the pre-trained CNN model in [13], to investigate if it necessary to train the classifier with the reconstructor. Our aim is to acquire a more comprehensive understanding of the impact that design decisions for the classifier have on the model's overall performance. Fig. 11 (b) shows the PGE result on 15 m traces of the SOTA CNN model. We can find that it takes 678 traces on average to recover a subkey. Therefore, in the following experiments, the CNN classifier of the DAR-NN model is concurrently optimized with updates to the DAR-NN model. This approach enables the classifier to adjust adaptively to modifications in the overall model structure, thereby effectively accommodating dynamic fluctuations in the training data. Nonetheless, this strategy might require an increased number of training iterations to reach peak performance.

Next, we test to which extent the DAR-NN20 model can recover the key from the "victim" traces and Fig. 11 (c) shows the PGE result. We find that by focusing on the specific leakage interval of the target *SBox* operation, the DAR-NN20 model is able to recover the subkey by using only **141** traces on average, as illustrated in Table. VI. This

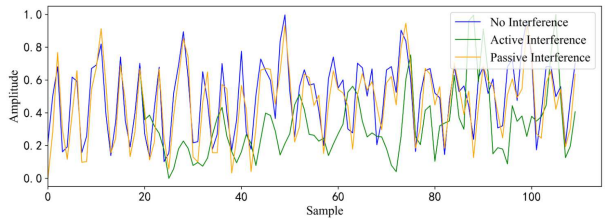


Fig. 12. The comparison of traces with no interference, 25.0% passive interference, and 25.0% active interference.

is a 58.7% improvement over the previous attack in [13]. By considering the CNN model in [13] is trained on 200K traces obtained from 5 different profiling devices, the DAR-NN20 model even requires 80% less profiling data, since it is built on only 40K traces captured from a single profiling device. To explore the strength of the DAR-NN20 model, we also tried to train another DAR-NN20_200K model on 200K "noisy" traces with the same amount of "groundtruth" traces captured from 5 profiling devices as the same training condition in [13]. Notice that the DAR-NN20_200K shows strong deployment practicality with efficient training (800s total), modest memory needs (3.7 - 3.9GB), and fast attacks (60s for 100 tests with each uses 5K traces). This aims to provide insights into how the model generalizes with a diverse training set of devices. Fig. 11 (d) shows that the model requires 154 traces on average to recover a subkey at 15 m distance, which is similar to the DAR-NN20 model's result. This indicates that 40K traces have already enough for the DAR-NN model to reach the optimal.

B. Scenario: Passive Interference

The second experiment aims to explore to which extent the proposed DAR-NN model can pose a threat in an attack scenario with passive interference originating from behind the attacker's parabolic antenna. In this part, we add the interference traces to "victim" traces with 5 proportions: 2.5%, 5.0%, 12.5%, 20.0%, 25.0%. For example, a 25% interference implies that 25% sample point in "victim" traces are subject to interference. Fig. 12 shows the comparison of the plots of traces with no interference, 25.0% passive interference and 25.0% active interference. These plot represent the 110 leakage interval of 16 *SBox* computations in the last round of AES. All traces are scaled to the interval [0,1]. From Fig. 12, we can clearly see significant differences in the shape of traces. The introduction of passive interference, represented by the orange trace, results in moderate distortion of the signal, while active interference, shown by the green trace, causes substantial signal distortion. This suggests that active interference has a more pronounced effect on signal degradation compared to passive interference.

Afterwards, we test the DAR-NN110 and the state-of-the-art CNN model on the resulting data affected by the passive interference. The second row of Table V shows the average number of traces required by DAR-NN110 model to recover the key with different passive interference ratios. All results are derived from the mean of 100 individual tests and, for each test, we get the number of traces by evaluating the model on 5 devices. From Table V, we can find that as the proportion of interference increases, the efficiency of the DAR-NN110 model diminishes, which fits our expectation. When the interference ratio is set to 2.5%, the DAR-NN110 model needs around 450 traces to recover the key. When it comes to the ratio of 25.0%, this result becomes around 1.3K.

TABLE VII

AVERAGE NUMBER OF TRACES REQUIRED BY MODELS TO RECOVER THE TARGET SUBKEY (PGE=0) AT 15 M DISTANCE WITH 2.5% PASSIVE INTERFERENCE RATIO (RESULTS FOR 100 TESTS)

Victim device	D1	D2	D3	D4	D5
DAR-NN	838	430	289	533	149
Switch-T Transformer [45]	>4999	>4999	>4999	>4999	>4999
ML-DAE [17]	1436	1333	1716	1680	947
LC-CNN [46]	>4999	3530	>4999	>4999	2177
Meta-Transfer CNN [35]	>4999	>4999	2875	>4999	2792
SOTA CNN [13]	>4999	>4999	1305	1284	730

TABLE VIII

AVERAGE NUMBER OF TRACES REQUIRED BY MODELS TO RECOVER THE TARGET SUBKEY (PGE=0) AT 15 M DISTANCE WITH 25.0% PASSIVE INTERFERENCE RATIO (RESULTS FOR 100 TESTS)

Victim device	D1	D2	D3	D4	D5
DAR-NN	1432	1563	876	1802	852
Switch-T Transformer [45]	>4999	>4999	>4999	>4999	>4999
ML-DAE [17]	3950	4216	3222	>4999	3030
LC-CNN [46]	>4999	>4999	>4999	>4999	>4999
Meta-Transfer CNN [35]	>4999	>4999	>4999	>4999	>4999
SOTA CNN [13]	>4999	>4999	3079	>4999	1969

For more details, the second row of Table. VII and VIII shows the average number of traces required by the DAR-NN110 model on different victim devices with the interference ratio set to 2.5% and 25.0%, respectively. We can find that D5 is an easier target for the DAR-NN110 model in the environment with passive interference. D4 exhibits the highest far-field EM side-channel security resistance in this case. This indicates that the board diversity is still a big issue to affect the attack efficiency of the DAR-NN model.

For the baseline model, it is unable to recover the subkey for certain devices within 5K traces, even with a minimal interference ratio of only 2.5%. For example, when the interference ratio set to 2.5%, the SOTA CNN model fails to recover the key from D1 and D2 within 5K traces. When it comes to the interference ratio of 25.0%, there is one more device, D4, which cannot be compromised by the SOTA CNN model. The last row of Table VII and VIII shows the average number of traces required by the SOTA CNN model on different victim devices with the interference ratio set to 2.5% and 25.0%, respectively. By comparing the results of the DAR-NN110 model and selected baseline models in Table VII and VIII, we can find that the proposed DAR-NN model shows a great capacity for dealing with the passive interference.

C. Scenario: Active Interference

Next, we check to which extent the active interference from the victim device side can protect the sensitive data from FEM-SCAs. As the same as above, the interference ratio is set from 2.5% to 25%. Fig. 12 demonstrates the extent of distortion in the trace with 25% active interference, highlighting its significance. We test the DAR-NN110 and the

TABLE IX

AVERAGE NUMBER OF TRACES REQUIRED BY MODELS TO RECOVER THE TARGET SUBKEY (PGE=0) AT 15 M DISTANCE WITH 2.5% ACTIVE INTERFERENCE RATIO (RESULTS FOR 100 TESTS)

Victim device	D1	D2	D3	D4	D5
DAR-NN	1123	418	327	684	258
Switch-T Transformer [45]	>4999	>4999	>4999	>4999	>4999
ML-DAE [17]	1525	1470	1686	2142	1110
LC-CNN [46]	>4999	3938	>4999	>4999	1938
Meta-Transfer CNN [35]	>4999	>4999	3399	>4999	2542
SOTA CNN [13]	>4999	2386	1140	2686	1080

TABLE X

AVERAGE NUMBER OF TRACES REQUIRED BY MODELS TO RECOVER THE TARGET SUBKEY (PGE=0) AT 15 M DISTANCE WITH 25.0% ACTIVE INTERFERENCE RATIO (RESULTS FOR 100 TESTS)

Victim device	D1	D2	D3	D4	D5
DAR-NN	4789	2718	692	1008	771
Switch-T Transformer [45]	>4999	>4999	>4999	>4999	>4999
ML-DAE [17]	>4999	>4999	4126	>4999	3671
LC-CNN [46]	>4999	>4999	>4999	>4999	>4999
Meta-Transfer CNN [35]	>4999	>4999	>4999	>4999	>4999
SOTA CNN [13]	>4999	>4999	4084	>4999	3883

selected baseline models on the resulting data affected by the active interference.

The second row of Table V shows the average number of traces required by our DAR-NN110 model to recover the key with different active interference ratios. We can find that it becomes much more difficult for the model to conduct a successful attack when traces are with higher ratio of interference, which fits our expectation. From Table V, we can find that the model's attack efficiency decreases as the ratio of interference increases in most cases. However, there is still an exception. When the active interference ratio is set to 5.0%, the number of traces required by the DAR-NN110 model decreases to 492, which is more efficient than the case with 2.5% interference. Our hypothesis is that traces with 5.0% interference may accidentally contain more features as the interference is randomly added to the "victim" traces.

For more details, the second row of Table IX and X shows the average number of traces required by the DAR-NN110 model on different victim devices with 2.5% and 25.0% active interference added, respectively. In this case, D1 shows a higher side-channel resistance to FEM-SCAs in the active interference environment. Considering that D4 is the top secure one in the passive interference case, this indicates that the side-channel resistance capabilities of different devices can vary depending on the environmental context.

In Table IX and X, we also compare the attack efficiency of our DAR-NN110 model with the baseline models. We can find that as the same as in the passive interference case, all baseline models cannot recover the subkey within 5K traces for D1, D2 and D4 without key enumerations. By analyzing the results in Table VII and VIII, the disparity in trace requirements indicates a higher efficiency of the DAR-NN model in key recovery under conditions of active interference. Additionally, there is a notable increase in the number of traces required for all models when the active interference ratio rises from 2.5% to 25%, demonstrating that higher levels of interference complicate the attack process. By combining the results on both passive and active cases, we can find that the DAR-NN model consistently requires fewer traces than the SOTA CNN model, which suggests that it is a more effective tool for security analysis in these scenarios.

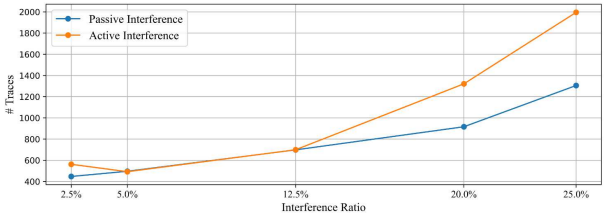


Fig. 13. The trend of average number of traces required to recover the target subkey (PGE=0) at 15 m distance by using the DAR-NN110 model with different interference ratio in passive and active scenarios.

TABLE XI

AVERAGE NUMBER OF TRACES REQUIRED BY THE DAR-NN110 MODEL TO RECOVER THE TARGET SUBKEY AT DIFFERENT ATTACK DISTANCES (RESULTS FOR 100 TESTS)

Distance \ Device	D1	D2	D3	D4	D5	AVG
0 m (Coaxial Cable)	41	51	42	57	29	44
5 m (Antenna)	70	87	76	116	40	78
10 m (Antenna)	198	295	216	301	119	226
15 m (Antenna)	233	288	237	235	105	220

We further plot the result of Table V in Fig. 13 to illustrate how interference can make the attack more difficult. It is obvious that the trend suggests a positive correlation between the interference ratio and the difficulty of attack execution. In particular, active interference appears to have a more pronounced effect on the increase in the number of traces required compared to passive interference, particularly evident beyond the 12.5% interference ratio threshold. This observation shows that active interference could potentially serve as a countermeasure against FEM-SCAs, as indicated by its greater impact on the required number of traces for such attacks, especially at higher interference ratios.

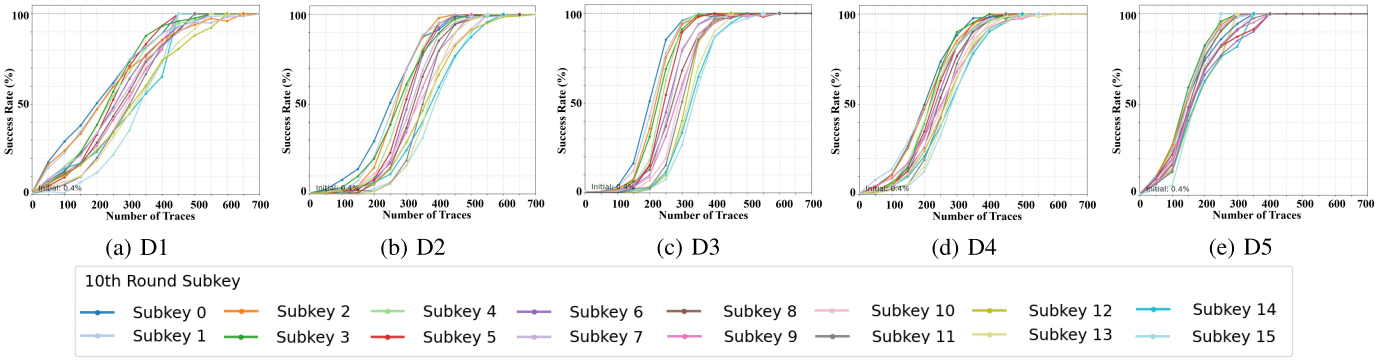
D. Distance Analysis

Next, we evaluate the effectiveness of the proposed DAR-NN110 model on FEM side-channel trace captured at other distances, validated through systematic tests at 0 m (coaxial cable), 5 m, 10 m, and 15 m. For each victim device, we further capture traces at 5 m and 10 m with the same fixed AES key as in the case of 15 m distance. For each position, we test the model under both passive and active interference. Table. XI shows the average number of traces required to recover the first subkey of the 10th round of the victim device at different capture positions. For each result, we run 100 independent tests and calculate the average number of traces required to achieve PGE=0. The results reveal a clear trend where the effectiveness of the attack decreases as the distance between the attacker and the target increases. At 0 m using a coaxial cable, the attack is most efficient, requiring an average of only 44 traces due to the stable and high-quality

TABLE XII

SELECTED 20-POINT TRACE SEGMENTS ALIGNED WITH 10TH-ROUND SBOX LOOKUP OPERATIONS FOR TRAINING 16 DAR-NN20 MODELS (M0 – M15) AND CORRESPONDING 10TH-ROUND SUBKEY VALUES FROM VICTIM DEVICES D1–D5

Model	M0	M1	M2	M3	M4	M5	M6	M7
Trace Interval	[200:220]	[225:245]	[250:270]	[275:295]	[206:226]	[231:251]	[256:276]	[281:301]
Target	Subkey 0	Subkey 1	Subkey 2	Subkey 3	Subkey 4	Subkey 5	Subkey 6	Subkey 7
Subkey Value (D1)	0xC7	0xB0	0x8C	0x2B	0x75	0xF9	0x5E	0xFD
Subkey Value (D2)	0xBA	0x17	0x3E	0x50	0x8F	0x06	0x92	0xE6
Subkey Value (D3)	0xA7	0xF8	0x03	0x9D	0xC6	0x70	0xDD	0x2A
Subkey Value (D4)	0x09	0x73	0xC4	0x65	0xBC	0xF4	0x3F	0xE0
Subkey Value (D5)	0x58	0x75	0x60	0x21	0x8F	0xC0	0xAE	0x81
Model	M8	M9	M10	M11	M12	M13	M14	M15
Trace Interval	[212:232]	[237:257]	[262:282]	[287:307]	[217:237]	[242:262]	[268:288]	[290:310]
Target	Subkey 8	Subkey 9	Subkey 10	Subkey 11	Subkey 12	Subkey 13	Subkey 14	Subkey 15
Subkey Value (D1)	0xAA	0x36	0x1D	0x6D	0x70	0xF8	0x27	0x2D
Subkey Value (D2)	0x2B	0x29	0x3D	0x61	0xA8	0xA1	0x55	0x4C
Subkey Value (D3)	0xB0	0xFE	0x6D	0xCB	0xD2	0x46	0x77	0x6B
Subkey Value (D4)	0x96	0xED	0xEE	0xB3	0xD5	0xB7	0x2C	0x46
Subkey Value (D5)	0x4A	0x44	0x4B	0x18	0x9E	0x3C	0x34	0xB6



(a) D1 (b) D2 (c) D3 (d) D4 (e) D5

(f) Legend

Fig. 14. Average attack success rates for the full AES key recovery of the DAR-NN model at 15 m distance across devices D1 – D5.

connection. Interestingly, the attack requires a similar number of traces at 10 m and 15 m, which is beyond our expectation. Our hypothesis is that environmental factors such as ambient interference or multi-path effects could dominate over pure distance-based attenuation on FEM side-channel leakage when the attack distance is beyond a certain range.

E. Full Key Recovery

To demonstrate the practical threat posed by adversaries, we conduct a comprehensive key recovery analysis by training 16 DAR-NN20 models (M0 - M15), each targeting a specific subkey byte. The training parameters and settings are identical to the best DAR-NN20 model in Subsection V-A. Based on the leakage detection results shown in Fig. 10, we carefully select 20-point trace segments corresponding to each 10th-round *SBox* lookup operation, from the original 400-point traces, to train each model separately. Once the full last-round key is recovered, the adversary can straightforwardly derive the original AES key by reversing the key expansion process. The precise intervals used for training models and the AES key value of each device are detailed in Table XII. Notice that the order of the 16 segments intervals aligns with the byte rearrangement result of the *ShiftRows* procedure.

After M0 - M15 are trained by using the "noisy" - "groundtruth" trace pairs, we further test all these models on traces captured at 15 m distance to the victim device in an office corridor environment without the manual interference. Fig. 14 shows the key recovery success rate of M0-M15 for the 10th round subkey (subkey 0 – 15) across five victim devices (D1 - D5) at 15 m distance. Each plot represents a different device, illustrating how many traces are empirically required to achieve a certain success rate in the subkey recovery task. The success rate curves for each device and subkey are generated through a rigorous testing process. To construct each result (curve), we conduct 20 separate test points, with the number of traces incrementing from 50 to 1000 in steps of 50 per point. For every individual test point, we run 100 independent attack attempts to get the average. This means that for each complete curve, we execute a total of 2K attack instances (20 test points $\times$ 100 attempts) to obtain a robust characterization of the attack's effectiveness. The result shows that DAR-NN model is feasible to recover the full key of the victim AES implementation using $\leq$700 traces.

Next, we test the models on traces captured from D1 - D5 at 15 m distance with 25% passive and active interference. Similarly, to derive each result, we conduct 20 separate test points, with the number of traces incrementing from 250 to 5000 in steps of 250 per point. Each individual test point is built on the average of 100 independent attack attempts. Table. XIII shows the number of traces required by the 16

TABLE XIII

NUMBER OF TRACES REQUIRED TO RECOVER THE FULL 10TH ROUNDKEY FOR MODELS M0-M15 WITH A 100% SUCCESS RATE IN DIFFERENT SCENARIOS (RESULTS FOR 100 TESTS)

Scenario \ Device	D1	D2	D3	D4	D5
Without Interference	700	700	600	600	400
Passive Interference	4250	4500	3750	>4999	3250
Active Interference	>4999	>4999	4250	>4999	3750

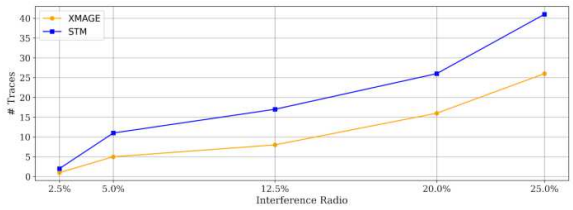


Fig. 15. The trend of average number of traces required to recover the subkey by using the DAR-NN20 model with different interference ratio to compromise XMEGA and STM boards by using power consumption as the side channel.

DAR-NN20 models to recover the full 10th roundkey of D1 - D5 with a 100% success rate.

F. Transferability

In this subsection, we conduct further evaluations using **power traces** captured from an 8 bit Atmel ATxmega128D4 and a 32 bit STM32F3 implementation of AES-128, to experimentally explore the transferability of the proposed model to other target devices, as well as alternative side channels. We refer to these two devices as XMEGA and STM boards. For each type of board, we prepare two devices: one designated for profiling and the other configured as the victim device. To capture power traces from these two boards, we use the NewAE ChipWhisperer-Lite tool kit as the power measuring equipment. In both cases, we define the "groundtruth", "noisy", "reconstructed", and "victim" traces as follows.

We call traces captured by using an SMA cable directly touched to the power supply of the profiling device as ***"groundtruth" traces***. In addition, we call traces captured from the profiling device with manually added white noise in a fixed interference ratio (2.5%) as ***"noisy" traces***. Afterwards, we call traces captured from the victim device with manually added white noise in different interference ratios (2.5% - 25%) as ***"victim" traces***.

The DAR-NN20 model with a loss factor of $\beta = 0.5$ is chosen for the subsequent experiments. We focus on the first 8-bit subkey. In both cases, we capture 100K traces from the profiling device as the training set, with 20% of traces randomly set aside for validation. Afterwards, we evaluate the trained DAR-NN20 model on 10K traces captured from the victim devices with different interference ratios. Fig. 15 shows the average number of traces required to recover the subkey with different interference ratio. The model can consistently recover the target subkey using < 50 traces. This highlights the transferability of the proposed DAR-NN model under high interference conditions.

VI. CONCLUSION AND FUTURE WORKS

In this paper, we propose a DAR-NN model to rebuild features from remotely captured traces in FEM-SCA scenarios. We first experimentally show that the DAR-NN model achieves a 58.7% improvement in attack efficiency over exiting approaches while necessitating 80% less profiling data. Afterwards, we establish two complex experimental settings by incorporating a radio to act as the source of interference. Our results demonstrate that the DAR-NN model significantly surpasses the efficiency of the baseline models by a multiple-fold margin. In addition, the model also demonstrates a strong capacity for robustness in complex environments characterized by varying interference ratios. Besides, we conduct further testing of the proposed methods on other side channels and victim devices to demonstrate the transferability of the approach. Future works include conducting similar attacks on hardware implementations of AES at long distances by using far-field EM emanations as the side channel. Certainly, the most important future work is the development of countermeasures against FEM-SCAs.

REFERENCES

[1] S. Qu, Y. Wang, J. Yu, C. Zhang, and D. Gu, "Trace copilot: Automatically locating cryptographic operations in side-channel traces by firmware binary instrumenting," *IACR Trans. Cryptograph. Hardw. Embedded Syst.*, vol. 2025, no. 1, pp. 128–159, Dec. 2024.

[2] B. Nassi et al., "Optical cryptanalysis: Recovering cryptographic keys from power LED light fluctuations," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur. (CCS)*, 2023, pp. 268–280.

[3] Y. Gao et al., "DeepTheft: Stealing DNN model architectures through power side channel," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2024, pp. 3311–3326.

[4] L. Batina, S. Bhasin, D. Jap, and S. Picek, "*CSI NN*: Reverse engineering of neural network architectures through electromagnetic side channel," in *Proc. USENIX Secur. Symp.*, 2019, pp. 515–532.

[5] M. Oberhuber, M. Unterguggenberger, L. Maar, A. Kogler, S. Mangard, and A. Kogler, "Power-related side-channel attacks using the Android sensor framework," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, 2025, pp. 1–18.

[6] T. Ni et al., "Uncovering user interactions on smartphones via contactless wireless charging side channels," in *Proc. IEEE Symp. Secur. Privacy (S&P)*, May 2023, pp. 3399–3415.

[7] J. Hu et al., "Password-stealing without hacking: Wi-Fi enabled practical keystroke eavesdropping," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur. (CCS)*, 2023, pp. 239–252.

[8] M. Cardaioli, M. Conti, K. S. Balagani, and P. Gasti, "Your PIN sounds good! Augmentation of PIN guessing strategies via audio leakage," in *Proc. Eur. Symp. Res. Comput. Secur. (ESORICS)*, 2020, pp. 720–735.

[9] D. He, H. Wang, T. Deng, J. Liu, and J. Wang, "Improving IIoT security: Unveiling threats through advanced side-channel analysis," *Comput. Secur.*, vol. 148, Jan. 2025, Art. no. 104135.

[10] G. Camurati, S. Poeplau, M. Muench, T. Hayes, and A. Francillon, "Screaming channels: When electromagnetic side channels meet radio transceivers," in *Proc. ACM SIGSAC Conf. Comput. Commun. Security*, 2018, pp. 163–177.

[11] G. Camurati, A. Francillon, and F.-X. Standaert, "Understanding screaming channels: From a detailed analysis to improved attacks," *IACR Trans. Cryptograph. Hardw. Embedded Syst.*, vol. 2020, no. 3, pp. 358–401, Jun. 2020.

[12] R. Wang, H. Wang, and E. Dubrova, "Far field EM side-channel attack on AES using deep learning," in *Proc. 4th ACM Workshop Attacks Solutions Hardw. Secur.*, Nov. 2020, pp. 35–44.

[13] R. Wang, H. Wang, E. Dubrova, and M. Brisfors, "Advanced far field EM side-channel attack on AES," in *Proc. ACM Cyber-Phys. Syst. Secur. Workshop (CPSS)*, 2021, pp. 29–39.

[14] Z. Hong-Yi, G. Da-Wu, Z. Chi, L. Yan, and Y. Yi-Dong, "Wireless side-channel analysis method based on repeater," *J. Cryptologic Res.*, vol. 9, no. 1, pp. 175–188, 2022.

[15] Z. Hong-Yi, G. Da-Wu, C. Pei, Q. Shi-Pei, and L. xiao Wei, "Wireless side-channel analysis method based on spectral addition," *J. Cryptologic Res.*, vol. 10, no. 4, pp. 862–878, 2023.

[16] S. Paguada, L. Batina, and I. Armendariz, "Toward practical autoencoder-based side-channel analysis evaluations," *Comput. Netw.*, vol. 196, Sep. 2021, Art. no. 108230.
[17] F. Hu, J. Shen, and P. Vijayakumar, "Side-channel attacks based on multi-loss regularized denoising AutoEncoder," *IEEE Trans. Inf. Forensics Security*, vol. 19, pp. 2051–2065, 2024.
[18] G. Zaid, L. Bossuet, M. Carbone, A. Habrard, and A. Venelli, "Conditional variational AutoEncoder based on stochastic attacks," *IACR Trans. Cryptograph. Hardw. Embedded Syst.*, vol. 2023, no. 2, pp. 310–357, Mar. 2023.
[19] Y. Ganin et al., "Domain-adversarial training of neural networks," *J. Mach. Learn. Res.*, vol. 17, no. 59, pp. 1–35, 2016.
[20] J. Guillaume, M. Pelcat, A. Nafkha, and R. Salvador, "Attacking at non-harmonic frequencies in screaming-channel attacks," in *Proc. Int. Conf. Smart Card Res. Adv. Appl.*, 2023, pp. 87–106.
[21] H. Wang, "Amplitude-modulated EM side-channel attack on provably secure masked AES," *J. Cryptograph. Eng.*, vol. 14, no. 3, pp. 537–549, Sep. 2024.
[22] Y. Ji, E. Dubrova, and R. Wang, "Is your chip leaking secrets via RF signals?," in *Proc. IEEE 55th Int. Symp. Multiple-Valued Log. (ISMVL)*, Jun. 2025, pp. 141–146.
[23] R. Wang, K. Ngo, and E. Dubrova, "Side-channel analysis of saber KEM using amplitude-modulated EM emanations," in *Proc. 25th Euromicro Conf. Digit. Syst. Design (DSD)*, Aug. 2022, pp. 488–495.
[24] P. Cao, C. Zhang, X.-J. Lu, H.-N. Lu, and D.-W. Gu, "Side-channel analysis for the re-keying protocol of Bluetooth low energy," *J. Comput. Sci. Technol.*, vol. 38, no. 5, pp. 1132–1148, Sep. 2023.
[25] S. Picek, G. Perin, L. Mariot, L. Wu, and L. Batina, "SoK: Deep learning-based physical side-channel analysis," *ACM Comput. Surv.*, vol. 55, no. 11, pp. 1–35, Nov. 2023.
[26] E. Brier, C. Clavier, and F. Olivier, "Correlation power analysis with a leakage model," in *Proc. Int. Workshop Cryptograph. Hardw. Embedded Syst.* Cham, Switzerland: Springer, 2004, pp. 16–29.
[27] S. Guo et al., "Exploiting the incomplete diffusion feature: A specialized analytical side-channel attack against the AES and its application to microcontroller implementations," *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 6, pp. 999–1014, Jun. 2014.
[28] S. T. Chari, J. R. Rao, and P. Rohatgi, "Template attacks," in *Proc. Inte. Workshop Crypto. Hardw. Embedded Syst. (CHES)*, 2003, pp. 13–28.
[29] D. Das, A. Golder, J. Danial, S. Ghosh, A. Raychowdhury, and S. Sen, "X-DeepSCA: Cross-device deep learning side channel attack," in *Proc. ACM/IEEE Design Autom. Conf. (DAC)*, Jun. 2019, p. 818.
[30] S. Hajra, M. Alam, S. Saha, S. Picek, and D. Mukhopadhyay, "On the instability of softmax attention-based deep learning models in SCA," *IEEE Trans. Inf. Forensics Security*, vol. 19, pp. 514–528, 2023.
[31] M. Staib and A. Moradi, "Deep learning side-channel collision attack," *IACR Trans. Cryptograph. Hardw. Embedded Syst.*, vol. 2023, no. 3, pp. 422–444, Jun. 2023.
[32] E. Cagli, C. Dumas, and E. Prouff, "Convolutional neural networks with data augmentation against jitter-based countermeasures: Profiling attacks without pre-processing," in *Proc. Int. Workshop Cryptograph. Hardw. Embedded Syst. (CHES)*, 2017, pp. 45–68.
[33] J. Kim, S. Picek, A. Heuser, S. Bhasin, and A. Hanjalic, "Make some Noise. Unleashing the power of convolutional neural networks for profiled side-channel analysis," *IACR Trans. Cryptograph. Hardw. Embedded Syst.*, vol. 2019, no. 3, pp. 148–179, May 2019.
[34] R. Benadjila, E. Prouff, R. Strullu, E. Cagli, and C. Dumas, "Deep learning for side-channel analysis and introduction to ASCAD database," *J. Cryptograph. Eng.*, vol. 10, no. 2, pp. 163–188, Jun. 2020.
[35] H. Yu, H. Shan, M. Panoff, and Y. Jin, "Cross-device profiled side-channel attacks using meta-transfer learning," in *Proc. 58th ACM/IEEE Design Autom. Conf. (DAC)*, Dec. 2021, pp. 703–708.
[36] L. Wu, G. Perin, and S. Picek, "The best of two worlds: Deep learning-assisted template attack," *IACR Trans. Cryptograph. Hardw. Embedded Syst.*, vol. 2022, no. 3, pp. 413–437, Jun. 2022.
[37] Y. Ji and E. Dubrova, "A side-channel attack on a masked hardware implementation of CRYSTALS-Kyber," in *Proc. Workshop Attacks Solutions Hardw. Secur. (ASHES)*, vol. 15, 2025, pp. 27–37.
[38] B. Timon, "Non-profiled deep learning-based side-channel attacks with sensitivity analysis," in *Proc. IACR Trans. Cryptograph. Hardw. Embedded Syst. (TCHES)*, 2019, pp. 107–131.
[39] G. Yang, H. Li, J. Ming, and Y. Zhou, "CDAE: Towards empowering denoising in side-channel analysis," in *Proc. Int. Conf. Inf. Commun. Secur. (ICICS)*, 2020, pp. 269–286.
[40] L. Wu and S. Picek, "Remove some noise: On pre-processing of side-channel measurements with autoencoders," *IACR Trans. Cryptograph. Hardw. Embedded Syst.*, vol. 2020, no. 4, pp. 389–415, Aug. 2020.
[41] C. Xiao, M. Tang, S. Karayalcin, and W. Cheng, "LD-PA: Distilling univariate leakage for deep learning-based profiling attacks," *IEEE Trans. Inf. Forensics Security*, vol. 20, pp. 17–30, 2025.
[42] C. Shi et al., "Privacy leakage via speech-induced vibrations on room objects through remote sensing based on phased-MIMO," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Nov. 2023, pp. 75–89.
[43] H. Wang, "Deep learning side-channel attacks on advanced encryption standard," Ph.D. dissertation, School EECS, KTH Royal Inst. Technol., Stockholm, Sweden, 2023.
[44] H. Pahlevanzadeh, J. Dofe, and Q. Yu, "Assessing CPA resistance of AES with different fault tolerance mechanisms," in *Proc. Asia South Pacific Design Autom. Conf. (ASP-DAC)*, 2016, pp. 661–666.
[45] J. Y. Liao, H. Wang, J. Wang, and Y. Tang, "Switch-T: A novel multi-task deep-learning network for cross-device side-channel attack," *J. Inf. Secur. Appl. (JISA)*, vol. 93, Sep. 2025, Art. no. 104146.
[46] L. Wu et al., "Label correlation in deep learning-based side-channel analysis," *IEEE Trans. Inf. Forensics Security*, vol. 18, pp. 3849–3861, 2023.

Huanyu Wang received the B.S. degree in electronic engineering from Dalian University of Technology, Dalian, China, and the M.S. and Ph.D. degrees in electronic engineering from the KTH Royal Institute of Technology, Stockholm, Sweden. He is currently an Associate Professor with Hunan University of Science and Technology, Xiangtan, China. His current research interests include hardware security, side-channel analysis, and deep learning-based applications.

Dalin He was born in Changsha, China, in 2000. He received the B.S. degree in optoelectronics engineering from Hunan University of Science and Technology, Xiangtan, China, where he is currently pursuing the master's degree in integrated circuit engineering. His current research interests include hardware security, side-channel analysis, and control flow monitoring.

Deng Tuo received the B.S. degree in optoelectronics engineering from Hunan University of Science and Technology, Xiangtan, China, where he is currently pursuing the master's degree in integrated circuit engineering. His current research interests include hardware security and side-channel analysis.

Junnian Wang received the B.S. and M.S. degrees in radio physics from Lanzhou University, China, and the Ph.D. degree in control theory and engineering from Central South University, Changsha, China. He is currently a Professor with Hunan University of Science and Technology, Hunan, China. He has authored over 50 scientific publications, more than 20 of which are indexed in SCI/EI. His research interests include deep learning, information processing, and fault diagnosis.