electronics

MDPI

Article

A Quantitative Analysis of Non-Profiled Side-Channel Attacks Based on Attention Mechanism

Kangran Pu, Hua Dang, Fancong Kong, Jingqi Zhang and Weijiang Wang *

Beijing Institute of Technology, Beijing 100081, China
* Correspondence: wangweijiangbit@163.com

Abstract: In recent years, the deep learning method has emerged as a mainstream approach to non-profiled side-channel attacks. However, most existing methods of deep learning-based non-profiled side-channel attack rely on traditional metrics such as loss and accuracy, which often suffer from unclear results in practical scenarios. Furthermore, most previous studies have not fully considered the properties of power traces as long time-series data. In this paper, a novel non-profiled side-channel attack architecture is proposed, which incorporates the attention mechanism and derives a corresponding attention metric. By attaching the attention mechanism after the network layers, the attention mechanism provides a quantitative prediction of correct key. Moreover, this architecture can effectively extract and analyze the features from long power traces. The success rate on different datasets is at least 86%, which demonstrates the superior reliability of this architecture compared to other works when facing various countermeasures and noise. Notably, even in scenarios where traditional loss and accuracy metrics fail to provide reliable results, the proposed attention metric remains capable of accurately distinguishing the correct key.

Keywords: non-profiled attacks; deep learning; attention mechanism; quantitative analysis

Citation: Pu, K.; Dang, H.; Kong, F.; Zhang, J.; Wang, W. A Quantitative Analysis of Non-Profiled Side-Channel Attacks Based on Attention Mechanism. *Electronics* **2023**, *12*, 3279. https://doi.org/10.3390/electronics12153279

Academic Editors: Jiaji He, Haoqi Shan and Wei Hu

Received: 4 July 2023
Revised: 28 July 2023
Accepted: 29 July 2023
Published: 30 July 2023

1. Introduction

Side-channel attacks (SCAs), first proposed by Paul Kocher in 1996 [1], represent a powerful category of cryptanalytic attacks. The SCAs leverage the leakage of physical information from the targeted device that executes the cryptographic algorithm to recover the encryption key. The instantaneous power consumed by a cryptographic device is influenced by the processed data and the executed operations. By analyzing the power characteristics of a cryptographic device, the encryption key becomes accessible with the knowledge of the operating cryptographic algorithm.

Side-channel attacks are generally categorized as profiled attacks and non-profiled attacks. Profiled attacks include techniques like template attacks [2,3] and stochastic attacks [4]. These attacks creates a profile on a device of the same type as the target device. Therefore, the attacker has full control over the profiling device. The power traces collected from the profiling device can be utilized to created a template to recover the keys of the target device.

Non-profiled attacks include simple power analysis (SPA) [5], differential power analysis (DPA) [6] and correlation power analysis (CPA) [7]. In these types of attacks, the attacker lacks the capability to manipulate the target device in advance to obtain relevant physical information related to the cryptographic operation. Instead, such information can only be captured during the attack. To retrieve the encryption key, statistical analysis tools are employed to calculate the correlation between the leakage points and the sensitive data.

With the development of deep learning techniques in recent years, the integration of deep learning techniques with side-channel attacks has improved the performance of SCAs. The pioneering work of Maghrebi in 2016 employs deep learning techniques for side-channel attacks [8]. Deep learning-based side-channel attack (DLSCA) offers advantages

over traditional template attacks by overcoming their limitations [9]. DLSCA has been proved to have the capability of breaking countermeasures in SCA, such as masking [10,11], shuffling [12], misalignment by random delay [13], and a more recent hiding technique called random dynamic frequency scaling (RDFS) countermeasure [14]. In addition to the works carried out on profiled attacks, non-profiled attacks have also been studied [15]. Considering the limited replicability of the target device in actual attack scenarios, this paper mainly focus on non-profiled attacks.

In 2019, Timon introduced a novel approach called differential deep learning analysis (DDLA), which applies convolutional neural network (CNN) and multi-layer perceptron (MLP) to recover the encryption key in a non-profiled scenario [16]. The training process of DDLA only requires a limited number of power traces obtained from the targeted device with an unknown and fixed key. DDLA has demonstrated superior performance compared to classical non-profiled attacks like CPA. Moreover, the efficiency of DDLA has been evaluated on both non-protected and protected devices with hiding (desynchronized power traces) or masking techniques.

Following the current trends of DLSCA, recent studies have explored the utilization of deep learning models as an alternative approach to traditional non-profiling SCA. Ma implied recurrent neural network (RNN) architecture to non-profiled DLSCA in [17]. In this work, the combination of RNN and SCA has been proven to be effective by experiments. In 2021, Lu combined CNN with an attention mechanism and introduced a new attention metric to carry out non-profiled attacks on the advanced encryption standard (AES) [18]. However, the selection of a correct key still relies on the distinction of attention probability curves, which shows common drawback with traditional metrics when the distinction between metric curves is not significant enough. Moreover, the attention-based CNN model primarily focuses on short-term intervals within the power traces. The authors in [19] introduced a precise data preparation technique for feature extraction. They also modified the labels into three categories during CNN training, reducing the number of power traces required for non-profiled SCA.

Essentially, power traces are long time sequences. In existing works, in order to effectively capture and retain a large amount of information, the computation complexity of a model will increase and thereby reduce the efficiency of attacks. In contrast, the incorporation of the attention mechanism improves the ability of neural networks to process information in long sequences and simplifies the model while accelerating computations [20,21]. Therefore, this article proposes a feed-forward network model integrated with an attention mechanism to handle the long-term input traces. To evaluate the performance of the proposed model, the attention mechanism is combined with both CNN and long short-term memory (LSTM) networks. Subsequently, a series of experiments is conducted on diverse datasets.

In addition, most existing works follow the metrics proposed by Timon for key recovery [22–25]. However, these metrics suffer from a drawback, as they lack quantitative analysis and accuracy. Because an obvious distinction in the network-training metrics is required to distinguish the keys, this method may be susceptible to the subjective judgment of attackers and sometimes inaccurate. To overcome the inaccuracies of traditional metrics, a novel metric based on the attention mechanism is proposed in this paper, which can directly distinguish the key by locating the maximum metric value.

1.1. Our Contributions

The contributions of this paper can be summarized as follows.

1. Introducing a novel neural network architecture integrated with the attention mechanism
   In this paper, a novel neural network architecture for non-profiled SCA is presented, named non-profiled attention-based side-channel attack (NASCA). The NASCA architecture comprises three main components. The first component is a feed-forward neural network, which is implemented by CNN and LSTM networks in this study. The second component is an attention network, which effectively captures features.

The last component consists of fully connected (FC) layers. With this innovative architecture, the information embedded in the long-term input traces is effectively captured, leading to successful side-channel attacks.

2. Introducing a novel attention metric of non-profiled DLSCA
   A novel metric based on the attention score vector is proposed, introducing quantitative analysis to non-profiled DLSCA, which offers more accurate results compared to traditional metrics. During the training process for each key, the attention mechanism calculates the attention score vector based on the feature vectors generated by the input network. For the correct key, the elements within the attention score vector exhibit a relatively large statistical dispersion, which can be quantified by the standard deviation. Therefore, the mean standard deviation of the attention score vector is adopted as the new metric. Through experiments on various datasets, correct keys can be consistently distinguished by this metric, even when traditional metrics fail to provide correct results.
3. Demonstrating the robustness of the proposed architecture and metric
   To evaluate the robustness of the novel architecture and indicators, experiments were conducted on various datasets with different SCA countermeasures, such as datasets with only masks (ASCAD), and desynchronized datasets (AES_RD), as well as datasets with masking and desynchronization countermeasures (ASCAD_desync50). In addition, the proposed architecture was applied to the power traces with additional Gaussian noise. Despite the presence of various countermeasures or noise, the novel architecture and metrics successfully completed the attacks, demonstrating the robustness in different SCA scenarios.

1.2. Paper Organization

The rest of this paper is organized as follows. Section 2 provides a comprehensive introduction to the fundamental theories of the target encryption algorithm and deep learning-based non-profiled SCA, as well as the necessary background of CNN and LSTM. In Section 3, the structure and working process of the novel NASCA method with the proposed attention metric is introduced in detail. Section 4 focuses on the experimental evaluation of the NASCA method. Finally, Section 5 includes the conclusion and the discussion of future work.

2. Preliminaries

2.1. Aes-128 Encryption

AES [26] is a block cipher algorithm based on field operations which is capable of encrypting a 128-bit packet using a key of 128, 192, or 256 bits in length. The specific encryption variants are referred to as AES-128, AES-192, and AES-256. Among these, AES-128 is the most commonly used block cipher algorithm in a wide range of applications. Consequently, the side-channel attack discussed in this paper is based on the implementation of AES-128.

The AES-128 encryption process requires ten iterations. Each round of the AES transformation consists of four fundamental steps: SubBytes, ShiftRows, MixColumns, and AddRoundKey. Notably, the last round excludes the MixColumns step.

The SubBytes operation is the only non-linear transformation within the AES algorithm which is closely related to the encryption key. Therefore, the key can be recovered from the intermediate value which corresponds to the output of the SubBytes operation. Moreover, SubBytes is executed frequently on the encryption device, resulting in obvious power consumption compared to other operations. Consequently, it becomes easier to identify the location of the SubBytes operation within the power trace, thereby enhancing the efficiency of side-channel analysis. As a result, the SubBytes operation of the AES algorithm is commonly selected as the point of interest (POI) [27] for side-channel attacks.

Let P denote a byte of the plaintext, K denote the key byte, $\oplus$ represent the XOR operation, and Y denote the output obtained after the SubBytes operation. The SubBytes operation can be represented as

$$Y = Sbox(P \oplus K) \tag{1}$$

2.2. Deep Learning-Based Non-Profiled Side-Channel Attack

Non-profiled side-channel attack means the attacker does not have access to a template device, which poses a challenge for deep learning methods due to the lack of labels from side-channel measurements. However, deep learning-based side-channel attack (DDLA) addresses this issue by generating training labels using key hypotheses, thereby solving the label problem.

In DDLA, the training labels are generated from key hypotheses, similar to how conventional correlation power analysis operates. The hypothetical intermediate values are generated form the key hypotheses K and accessible plaintexts P by (1), which exhibit a strong correlation with the POI. By using the least significant bit (LSB) of Y as labels, DDLA enables the training of deep learning networks, even in non-profiled scenarios.

In deep learning, networks trained with correct labels generally outperform networks trained with mislabeled data [28]. To distinguish the correct key, metrics for deep learning performance description such as loss and accuracy are utilized in DDLA. These metrics reflect the performance of the training process of different key hypotheses. For example, the correct key hypothesis corresponds to a clear distinction in these metrics, allowing for the identification of the correct key. Through the distinction of the metrics, DDLA effectively determines the correct key in a non-profiled SCA. The specific algorithm of DDLA is illustrated in Algorithm 1.

Algorithm 1 Differential Deep Learning Analysis (DDLA).

Input: N power traces $\{T_i\}_{1\leq i\leq N}$ with corresponding plaintexts $\{P_i\}_{1\leq i\leq N}$, an neural network Net, number of epochs Ne, substitution box $Sbox$
Output :The metrics of training m_k

for $k \in (0, 255)$ **do**
  Initialize the training parameters of Net_k
  Compute the hypothetical intermediate values $(H_{i,k})_{1\leq i\leq N} = Sbox(P_i \oplus k)$
  Compute the training labels $L_{i,k} = LSB(H_{i,k})$
  for $e < Ne$ **do**
    Perform Deep Learning Training Process: $DL_k = Net\{T_i, L_{i,k}\}$
  end for
  Compute and save metrics m_k of DL_k
end for
return Metrics $\{m_k\}_{0\leq k<256}$

2.3. CNN

CNN is widely recognized for its remarkable success in the field of computer vision [29]. It consists of two main types of layers: convolutional layer and pooling layer. The basic structure of CNN is shown in Figure 1.

For the convolutional layer, it performs convolution operations to the input by the filters as a group of sliding windows. The weights of these filters are shared across different spatial locations, allowing similar patterns at various positions to be detected. Furthermore, the pooling layer slides over the input and pools local data into a single value using functions such as maximum or average. This pooling process helps reduce the dimension of the data and captures the most important features.The outputs of the stacked convolutional layers are often referred to as feature maps, which distinguish them from the raw input data. Moreover, these features exhibit characteristics such as translation invariance [30], making CNNs well suited for handling desynchronization issues within SCAs [31,32]. In order to effectively extract the features, some extra methods such as the

attention mechanism are combined with CNN [33], which provides an innovative approach for SCAs.

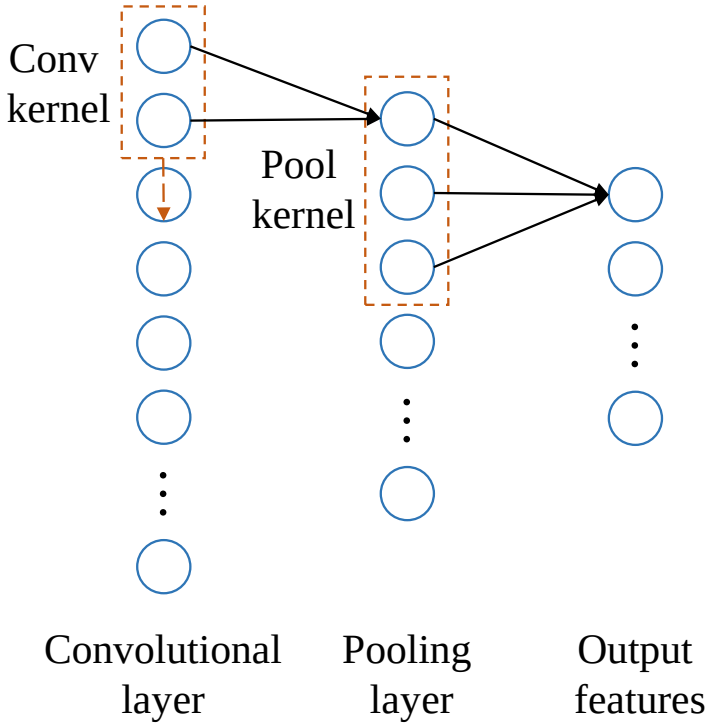

Figure 1. The structure of CNN.

2.4. LSTM

The LSTM network [34] contains multiple sub-networks referred to as memory cells. The LSTM cell operates with sequence data as input and produces feature maps as output. Each memory cell includes the input at the current time step (x_t), the cell state (c_t), the hidden state (h_t), and the output of the cell (y_t). Figure 2 illustrates the fundamental unit of an LSTM.

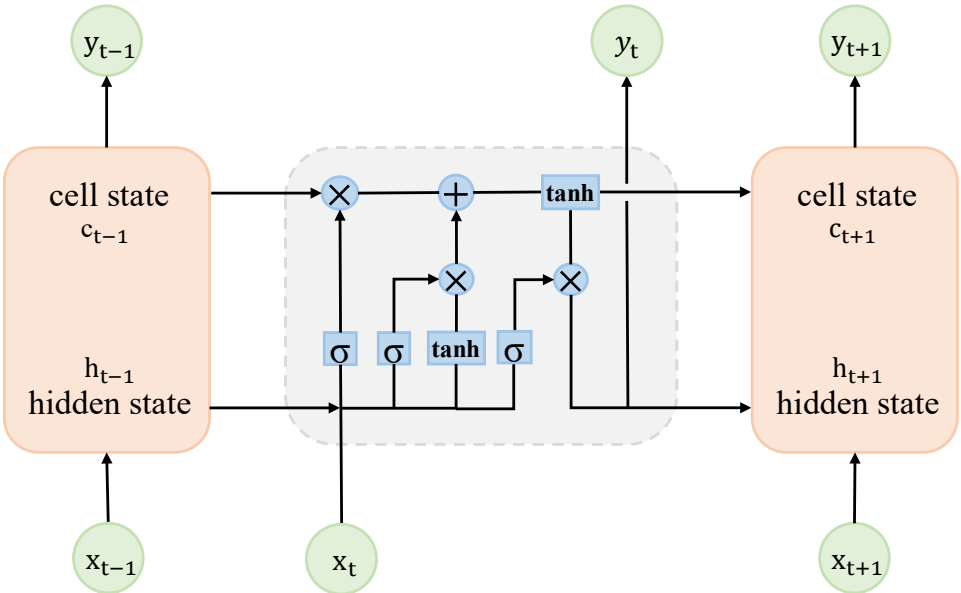

Figure 2. The structure of LSTM cell.

The outputs of an LSTM network differ from those of a traditional RNN [35]. The LSTM introduces a pathway known as the cell state in addition to the original output. The cell state pathway only involves multiplication and addition operations, without any non-linear units.The key advantage of LSTM lies in the additional gates, namely the input gate, forget gate, and output gate. The forget gate employs a non-linear activation function and multiplication operation to selectively discard unnecessary information from the cell state. The input gate combines the cell candidate state with new information, adding them to the cell state. The output gate determines the output information based on the cell state. By manipulating these gates, the LSTM can dynamically adjust the information flow, deciding how much information is retained from previous cells and what new information is added into the current cell state and then passed on to the next cell. This gating mechanism significantly enhances the ability to capture long-term dependencies and selectively retain relevant information.

3. Proposed Attention-Based Architecture and Metrics for Non-Profiled Attacks

In this section, a novel architecture for non-profiled side-channel attacks is proposed, which utilizes neural networks with the attention mechanism, and a quantitative attention metric is derived. This method is referred to as a non-profiled attention-based side-channel attack (NASCA) hereafter. In Section 3.1, we offer a detailed introduction to the specific non-profiled attack process with the new metric based on the proposed architecture. In Section 3.2, the structure of the CNN network integrated with the attention mechanism, denoted as NASCA-CNN, is provided. In Section 3.3, we describe the structure of the LSTM network integrated with the attention mechanism, which is denoted as NASCA-LSTM. In addition, how the attention mechanism is incorporated into the two networks and provides insights into their working mechanism is explained respectively.

3.1. Proposed Non-Profiled Attack Process with Attention Metrics

The essence of the attention mechanism draws inspiration from the human visual attention system. In simple terms, it functions as a mechanism to assign weight parameters, with the goal of aiding the model in capturing crucial information, aligning with the purpose of side-channel attacks in feature extraction from power traces. In DLSCA, the input of the network typically consists of lengthy temporal sequences, and the incorporation of attention mechanisms enables more effective connections between the whole sequence and leakage points. Additionally, the attention mechanism facilitates parallel computation with fewer parameters, leading to lower model complexity. When combined with other feature extraction networks, it does not cause a significant increase in the overall complexity.

For the AES algorithm, the output of SubBytes is commonly chosen as the intermediate value to calculate labels. In the non-profiled attack process, each key hypothesis and plaintext are utilized to conduct an XOR operation, followed by the SubBytes operation. The LSB value of the S-box output is determined as the training label in this paper. Subsequently, the network can be trained using power traces as input data. The mean standard deviation of the attention score vector described in (3) is proposed as a new metric for non-profiled attacks. In (2) and (3), α denotes the attention score vector mentioned in (6) and (9), n is the length of α, and *epochs* denotes the total number of training epochs:

$$\sigma_i = \sqrt{\frac{\sum_{i=1}^{n} (\alpha_i - \bar{\alpha})^2}{n}} \tag{2}$$

$$m_k = \frac{\sum_{i=1}^{epochs} \sigma_i}{epochs} \tag{3}$$

For the correct key hypothesis, the actual traces and calculated labels are consistent, leading to a converged network and promising training results characterized by high

accuracy and low loss. The elements in the feature vector that are most relevant to the final output label will receive higher attention scores assigned by the attention network. As a result, there will be noticeable variations in the attention scores between these relevant elements and other unrelated elements. This leads to significant statistical dispersion in the attention score vector as a whole. To quantify the severity of the statistical dispersion, the standard deviation can be employed as an indicator. By calculating the mean standard deviation of the attention score vector generated for each key hypothesis during the training process, we can utilize it as a metric for distinguishing different keys. The higher standard deviation of the attention score vector indicates greater statistical dispersion and provides valuable insight into the distinction of different key hypotheses. Figure 3 gives an illustration of the attention scores of different keys.

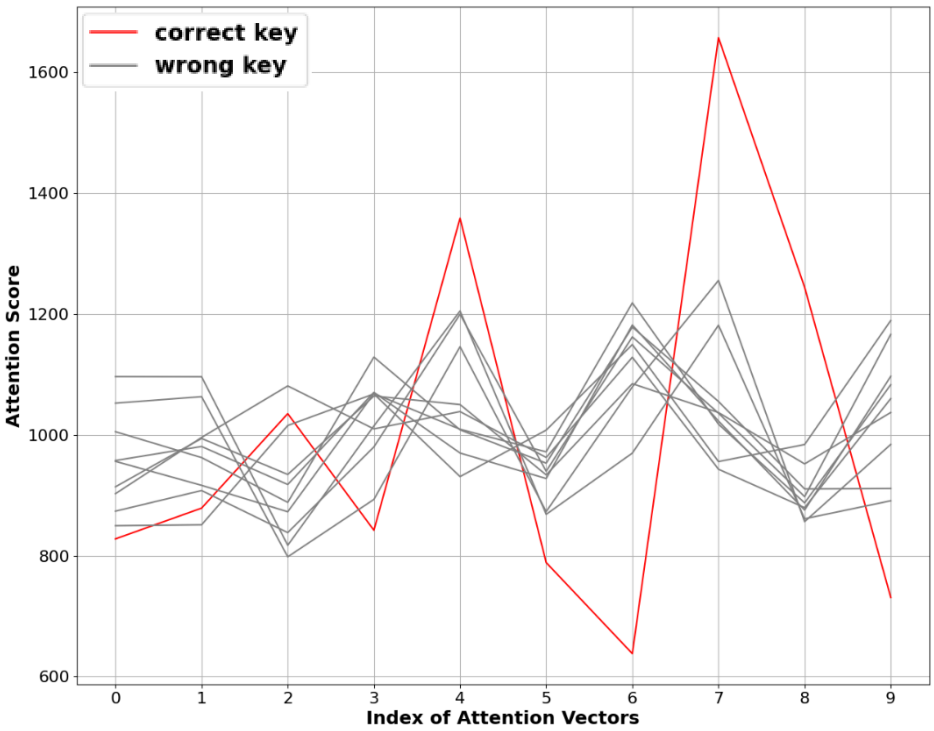

Figure 3. An illustration of attention score vector.

On the contrary, each wrong key hypothesis leads to mislabeled traces. As a consequence, the network fails to converge during the training process, which yields lower accuracy and higher loss compared to converged networks. Regarding the proposed attention metric, the attention mechanism fails to assign higher attention scores to the relevant elements in the feature vectors based on the incorrect training label. Therefore, both relevant and unrelated elements acquire similar attention scores. Reflected in the standard deviation of the attention score vector, the metrics of the wrong key hypotheses will not exceed that of the correct key hypothesis. At this point, the correct key can be distinguished by locating the maximum standard deviation value. Algorithm 2 describes the specific non-profiled attack process with the attention metric in detail.

Algorithm 2 Proposed non-profiled attention-based side-channel attack (NASCA).

Input: N power traces $\{T_i\}_{1\leq i\leq N}$ with corresponding plaintexts $\{P_i\}_{1\leq i\leq N}$, an attention- based network $ANet$, number of epochs Ne , substitution box $Sbox$
Output :The correct key $\hat{k}$

for $k \in (0, 255)$ **do**
  Compute the hypothetical intermediate values $(H_{i,k})_{1\leq i\leq N} = Sbox(P_i \oplus k)$
  Compute the training labels $L_{i,k} = LSB(H_{i,k})$
  Initialize the network parameters of $ANet_k$
  for $e < Ne$ **do**
    Perform Deep Learning Training Process: $DL_k = ANet\{T_i, L_{i,k}\}$
    Calculate and save the standard deviation of attention score vector $\alpha_{k,e}$
  end for
  Compute metric m_k according to (2) and (3)
end for
$M_k = max(m_k)$
return $\hat{k} = argmax_k M_k$

3.2. Proposed NASCA Based on CNN Architecture

After their remarkable success in the field of computer vision, CNNs have gained increasing popularity in the domain of side-channel analysis, as the task of classifying a side-channel trace is similar to classifying an image in various aspects.

In [36], it is observed that attention can serve as an indicator of whether the network is converging towards the desired direction. Inspired by this insight, a CNN network integrated with an attention mechanism is constructed.

3.2.1. Architecture of NASCA-CNN

The input of the proposed attention-based CNN architecture is the raw data from datasets, which are essentially measured power traces, while the output is the estimated LSB of a hypothetical intermediate value. The novel architecture architecture can be broadly divided into three components as illustrated in Figure 4.

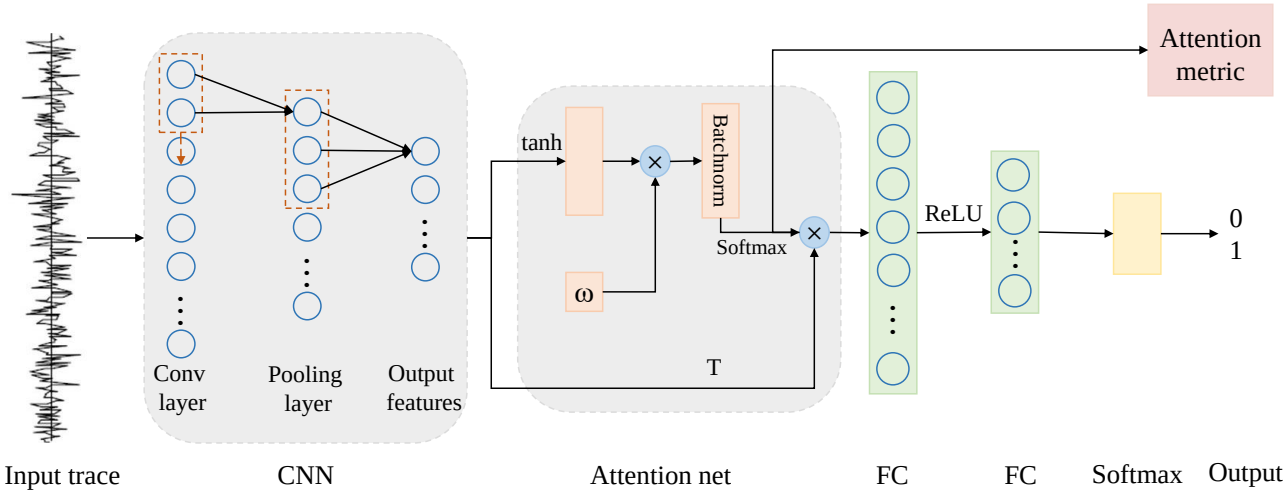

Figure 4. The architecture of NASCA-CNN.

1. The first component of the architecture comprises stacked convolutional layers and pooling layers. Specifically, three convolutional layers are employed to extract features from input traces, and each of them is followed by an average pooling layer. The first convolutional layer consists of four filters with a size of 32 and a stride of 1, followed by pooling layers with a size of 2. The second convolutional layer consists of eight filters with a size of 16 and a stride of 1. Subsequently, pooling layers with a size of 4

are applied. The final layer employs 16 filters with a size of 8 and a stride of 1. It is then followed by pooling layers with a size of 8.

2. The second part is the attention network, which evaluates the weight of each feature vector generated by the CNN. The CNN has an input size of 700, resulting in an output size of (16, 9). To calculate the attention weights, the feature vectors are multiplied by their respective weights, leading to a weighted sum output of size 16. Let $\mathbf{V}$ denote the feature vector produced by CNN, ω denote the trainable parameter vector, α denote the attention score vector and γ be the output generated by the weighted sum operation:

$$\mathbf{M} = tanh(\mathbf{V}) \tag{4}$$

$$\boldsymbol{\alpha}^{'} = batchnorm(\boldsymbol{\omega}^{T}\mathbf{M}) \tag{5}$$

$$\boldsymbol{\alpha} = softmax(\boldsymbol{\alpha}^{'}) \tag{6}$$

$$\boldsymbol{\gamma} = \mathbf{V}\boldsymbol{\alpha}^{\mathbf{T}} \tag{7}$$

3. The third part consists of two FC layers. As a result of the attention network, the dimension of the data input to these FC layers is reduced to 16. The first FC layer is composed of 16 input neurons and 8 output neurons, with rectified linear unit (ReLU) activation. The ReLU activation function introduces non-linearity and helps in capturing complex relationships within the data. The second FC layer is composed of 8 input neurons and 2 output neurons, utilizing SoftMax activation. The SoftMax activation function is commonly used in multi-class classification tasks, as it produces a probability distribution over the classes, allowing for the selection of the most probable class based on the network's outputs.

3.2.2. The Working Mechanism of NASCA-CNN

CNNs possess a natural property of translation invariance, making them well suited for extracting information, even from desynchronized time traces. Therefore, the translation-invariance property is primarily achieved through the utilization of convolutional layers in NASCA-CNN. In the context of neural networks, convolution is defined as the detection of features at different positions. This means that regardless of the location of the leakage point within the input power trace, the convolutional layer will identify the same features and produce consistent responses. For instance, if the input power trace is shifted by a certain offset, the convolutional kernel will still be able to detect the features associated with the leakage point after undergoing certain shift. Consequently, even when the features associated with power leakage are not located at the same position within the input power traces, the features can always be detected and passed on to the subsequent FC layers. Additionally, as a result of the weighted sum calculation performed by FC layers, features that have been activated by the CNN can be transmitted to the subsequent layers.

The attention mechanism in this architecture performs a weighted sum operation described in (7), which combines all the feature vectors and produces a single vector as a representation. This mechanism plays a significant role in controlling the weights of different feature vectors through the attention probabilities. By assigning higher attention probabilities to certain feature vectors, the attention mechanism regulates the proportion of backpropagated gradients during the training process. Consequently, the updates of network parameters primarily depend on the gradients associated with the feature vectors that have higher attention probabilities. This enables the network to focus more on the relevant and informative features, leading to more effective learning and optimization.

3.3. Proposed NASCA Based on LSTM Architecture

It should be noted that CNNs and MLPs may not be exactly suited for learning time-series data. Conversely, the RNN algorithm demonstrated its suitability for non-profiled side-channel attacks based on NDLPA in [17] due to its memory effect on sequential data.

In [37], the attention mechanism is employed in LSTM to address the gradient problem encountered in profiled attacks. Hence, an LSTM architecture integrated with an attention mechanism is proposed in this paper.

3.3.1. Architecture of NASCA-LSTM

The proposed attention-based LSTM architecture takes power traces as input and generates an estimated LSB of a hypothetical intermediate value (0 or 1) as the output. The architecture can be roughly segmented into three components as depicted in Figure 5.

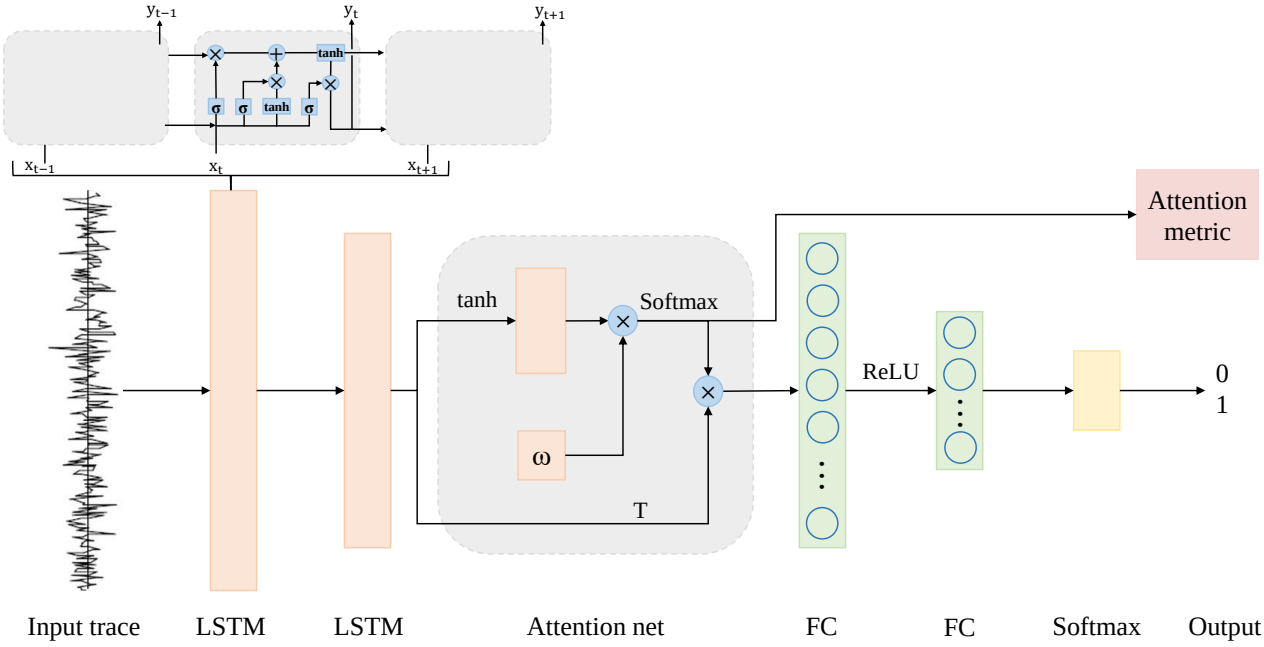

Figure 5. The architecture of NASCA-LSTM.

1. The first part is the non-bidirectional LSTM layer with one hidden layer, whose input size is 70, hidden size is 64 and sequence length is 10. For non-bidirectional LSTM, the hidden size equals the output size.
2. The second component is the attention network, which evaluates the importance of each feature vector generated by the LSTM. The attention network takes inputs with a size of 64, with an input sequence length of 10, and generates an output with a size of 64. The attention network multiplies the weights of the feature vectors to calculate a weighted sum output. Let $\mathbf{V}$ denote the feature vector produced by LSTM, ω denote the trainable parameter vector, α denote the attention score vector and γ be the output generated by the weighted sum operation:

$$\mathbf{M} = tanh(\mathbf{V}) \tag{8}$$

$$\boldsymbol{\alpha} = softmax(\boldsymbol{\omega}^T \mathbf{M}) \tag{9}$$

$$\boldsymbol{\gamma} = \mathbf{V}\boldsymbol{\alpha}^{\mathbf{T}} \tag{10}$$

3. The third component consists of two FC layers. Thanks to the attention network, the dimension of the data input to these FC layers is reduced. The first layer comprises 64 input neurons and 32 output neurons, employing ReLU activation. Subsequently,

the second layer consists of 32 input neurons and two output neurons, utilizing SoftMax activation.

3.3.2. The Working Mechanism of NASCA-LSTM

When employing a feed-forward neural network, the assumption is that the current input is entirely independent of the previous input. However, when working with time-series data, it is more reasonable to consider the input information from the current and previous moments in a connected way. The LSTM is a type of RNN that effectively utilizes the useful information across different time steps within an input time series. When dealing with input traces consisting of 700 time samples, the LSTM can effectively capture and represent the information contained in long power traces without ignoring (or forgetting) useful details over a long duration. Additionally, the LSTM can avoid issues, such as vanishing or exploding gradients, that may arise in traditional RNNs.

Subsequent to the LSTM, the attention network evaluates the weight scores of the feature vectors generated by the LSTM, as mentioned in (9). This evaluation enables the attention network to determine which portions of the entire input hold greater significance for the final classification task. In LSTM, the calculations are performed sequentially; thus, multiple steps of information accumulation are required for long-distance interdependent features to connect each other. Longer distances between such features make less effective capture in the absence of an attention mechanism. Therefore, the inclusion of an attention mechanism makes it easier to capture long-distance interdependent features in traces. Furthermore, during the backpropagation process, neurons with higher weights receive larger updates to their gradients, thus enabling faster learning.

4. Experimental Results

In this section, the performance evaluations of the proposed NASCA architecture are provided. The experiments were conducted on a personal computer with the following hardware configuration: an Intel i5-4590 CPU, an NVIDIA GeForce RTX 2070 SUPER GPU, and 16 GB RAM. The network was implemented using PyTorch (Version 1.12.1) and CUDA (Version 11.6) for efficient computation.

4.1. Datasets

To investigate the proposed architecture with reference value, experiments were conducted on three commonly used diverse datasets in terms of mask and desynchronized countermeasures, namely ASCAD [38], ASCAD_desync50, and AES_RD [13]. All the datasets were collected from power traces obtained during the execution of the protected software AES encryption algorithm running on the real board. In addition, desynchronization and mask are common countermeasures in real-world protection; thus, the ASCAD_decync50 and AES_RD can represent the protected scenarios encountered in real-world settings. The detailed attack settings of these datasets are summarized in Table 1.

- **ASCAD**: The power traces in the ASCAD dataset were collected from a first-order protected software AES encryption running on an 8-bit ATMega8515 board. The ASCAD dataset consists of a profiling set of 50,000 traces, and an attack set of 10,000 traces. Each trace contains 700 samples associated with the SubBytes operation of the third byte during the first round of the AES-128 encryption process. The 16-byte key is the same for all traces, while the plaintexts are randomly generated. Therefore, both the profiling set and the attack set can be used for non-profiled attacks. In this paper, the profiling set is chosen to apply NASCA.
- **ASCAD_desync50**: This dataset was obtained by introducing a misalignment of the samples in the ASCAD dataset, with a maximum window of 50. This dataset also contains 700 samples and the same number of power traces as the ASCAD dataset.
- **AES_RD**: Regarding this dataset, the AES encryption algorithm is implemented on a 8-bit Atmel AVR microcontroller with an encryption key of 0x2b7e151628aed2a6abf715 8809cf4f3c. The random delay countermeasure proposed by Coron and Kizhvatov

in [13] is implemented as the protection mechanism. This dataset consists of 50,000 power traces, and each trace contains 3500 samples, which are compressed by selecting 1 sample (peak) of each CPU clock cycle.

Table 1. Details of datasets.

Dataset	ASCAD	ASCAD_desync50	AES_RD
Traces	60,000	60,000	50,000
Samples	700	700	3500
Countermeasures	Boolean Mask	Boolean Mask	Boolean Mask and Random Delay
Target Byte	Byte 3	Byte 3	All Bytes

4.2. Performance of NASCA on ASCAD

Since the majority of the DLSCA work is carried out under experimental conditions with 50 epochs per key hypothesis and a batch size of 1000, these parameters are adopted for our experiment to provide a reference value for this work. In this experiment, both NASCA-CNN and NASCA-LSTM architectures are applied to the profiling set of ASCAD under the same experimental condition. In actual attack scenarios, obtaining a large number of power traces for training is relatively difficult, and thus both networks are trained with 10,000 power traces with a learning rate of 0.005. The mean squared error (MSE) loss function and adaptive moment estimation (Adam) optimizer are employed.

In the experimental results shown in Figures 6 and 7, the red curves represent the loss and accuracy of the correct key, while the results for other incorrect keys are represented by gray curves. Based on the analysis of the loss and accuracy metrics proposed by Timon, the key value of 224 shows the most distinctive training results and is considered the correct key, which is consistent with the actual key. The observed results indicate that both architectures proposed in this paper are effective in distinguishing the correct keys.

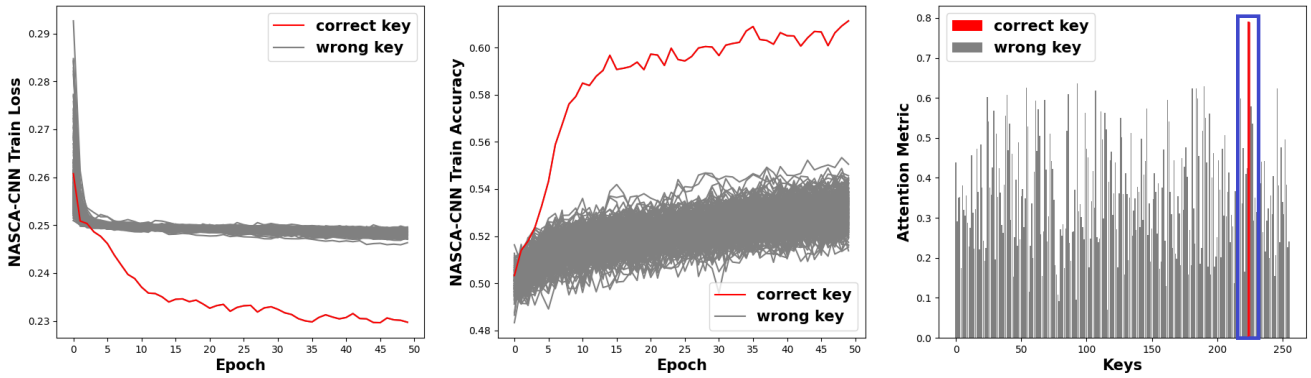

Figure 6. The attack results of NASCA-CNN on ASCAD with 10,000 traces.

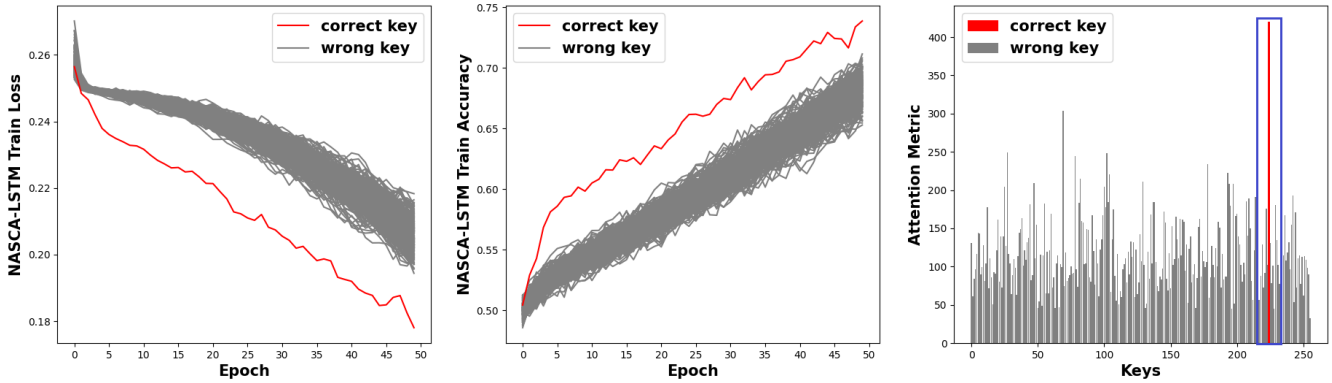

Figure 7. The attack results of NASCA-LSTM on ASCAD with 10,000 traces.

Based on the results of the mean standard deviations of the attention score vectors, the maximum value is achieved when the key value is 224. Therefore, from the view of the proposed attention metric, 224 is considered the correct key in two types of NASCA architectures. This also proves that the proposed metric is effective in distinguishing the correct keys.

4.3. Performance of NASCA on ASCAD _desync50

To evaluate the robustness of the NASCA algorithm on desynchronized datasets, the NASCA-CNN architecture is applied to ASCAD_desync50 dataset. DDLA constructed by CNN can break the desynchronization countermeasure without preprocessing due to the translation-invariance property of the CNN architecture [16]. The training of NASCA-CNN employs the same training parameters as described in Section 4.2, except with the modification of the number of power traces to 20,000. The results of the attack are given in Figure 8.

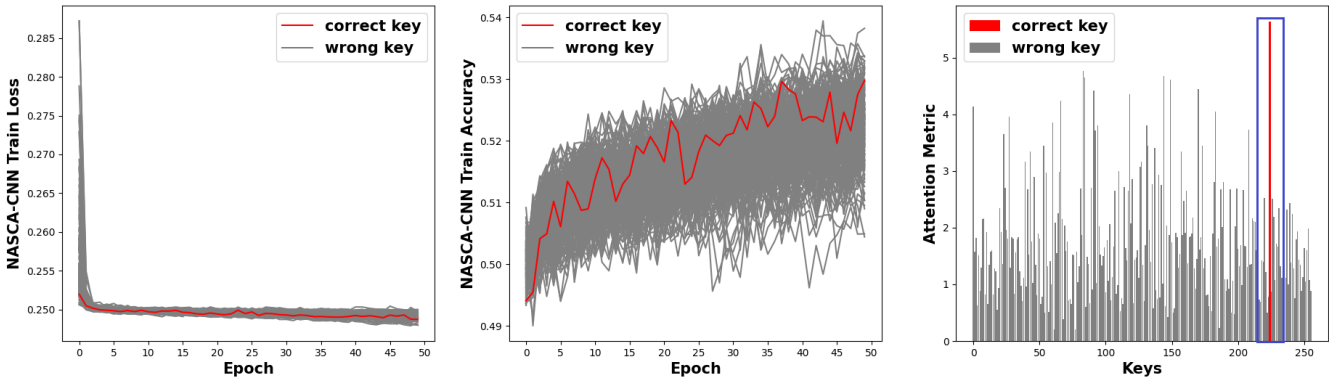

Figure 8. The attack results of NASCA-CNN on ASCAD_desync50 with 20,000 traces.

Regarding the proposed attention metric, the correct key can be recovered obviously by locating the maximum attention metrics yielded by the attention mechanism, which indicates that the NASCA algorithm can handle non-aligned datasets. Furthermore, the structure of the NASCA-CNN structure remains the same, which suggests that the proposed NASCA algorithm exhibits a certain level of robustness. However, the traditional loss and accuracy fail to give a correct result.

Notably, the results of this experiment clearly demonstrate the superiority of quantitative analysis. Unlike traditional loss and accuracy metrics that work by identifying the most distinguishable curve, the proposed metric provides a more quantitative approach by calculating the mean standard deviation. In this experiment, it is not feasible, as shown in Figure 6, to identify the correct key by the loss or accuracy metric. By employing the proposed attention metric, the correct key can be obtained more objectively and accurately, minimizing the uncertainties caused by traditional metrics.

4.4. Performance of NASCA on ASCAD with Additional Noise

In real-world scenarios, the noise generated in traces collection stage is often Gaussian noise. Therefore, in order to assess the robustness of NASCA against power traces with additional noise, two different levels of Gaussian noise are introduced. One with a standard deviation of 0.2 ($\sigma = 0.2$) and the other with a standard deviation of 0.5 ($\sigma = 0.5$). Subsequently, the attacks on the noisy ASCAD datasets are carried out by both NASCA-CNN and NASCA-LSTM under the same training parameters described in Section 4.3. The results of the attack against the noise level of 0.2 are presented in Figures 9 and 10. Figures 11 and 12 show the attack results against the noise level of 0.5.

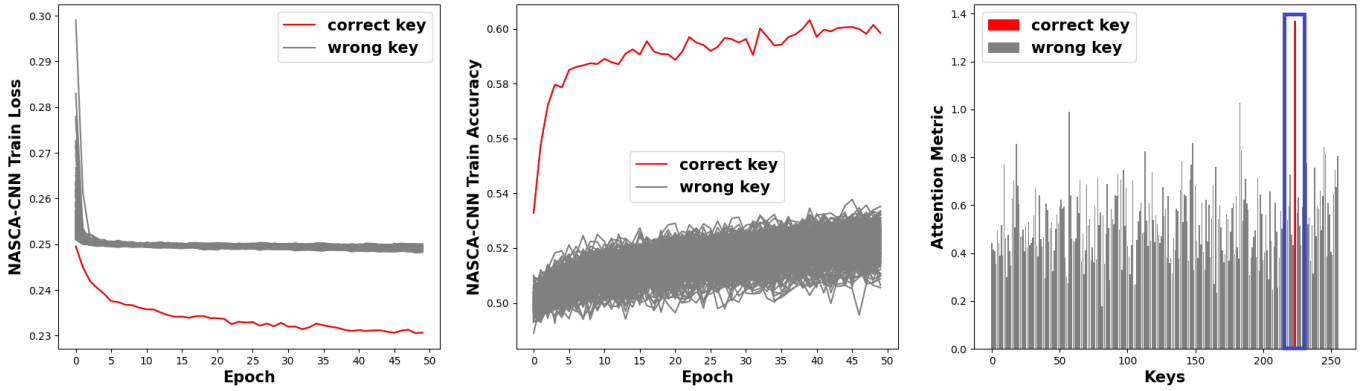

Figure 9. The attack results of NASCA-CNN on noisy ASCAD with 20,000 traces. ($\sigma = 0.2$).

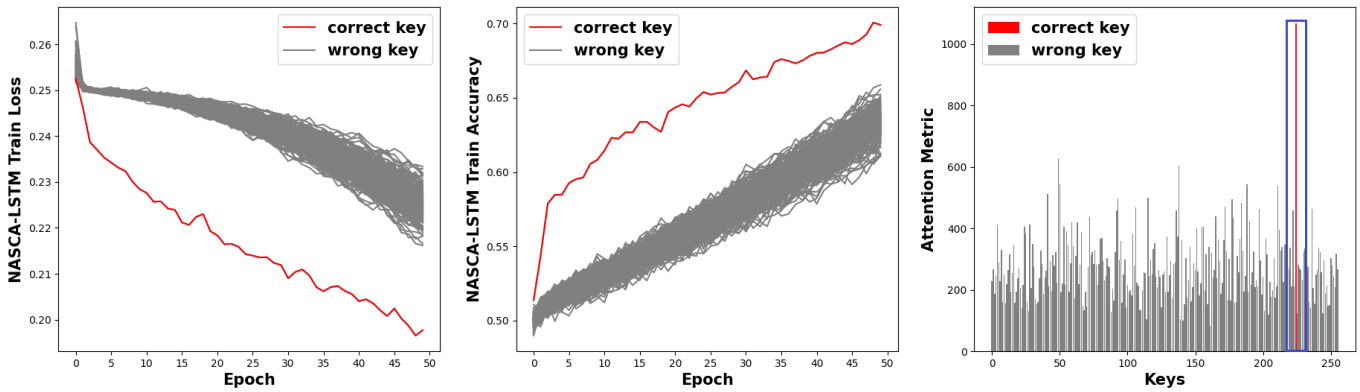

Figure 10. The attack results of NASCA-LSTM on noisy ASCAD with 20,000 traces. ($\sigma = 0.2$).

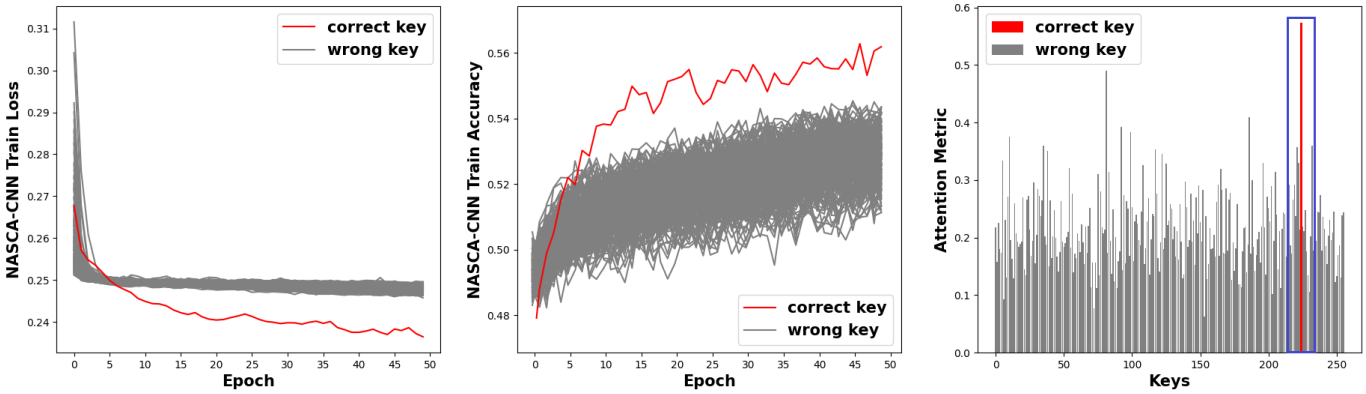

Figure 11. The attack results of NASCA-CNN on noisy ASCAD with 20,000 traces. ($\sigma = 0.5$).

As shown in Figures 9 and 10, it is clear that with a small level of Gaussian noise ($\sigma = 0.2$), both architectures show promising performance in detecting the correct key without preprocessing. Moreover, the attention metric maintains its quantitative identification capability. The attack results indicate that both architectures and the attention metric retain their effectiveness against additional noise. A similar trend can be seen at the higher level of Gaussian noise ($\sigma = 0.5$). Despite the performance of the loss and accuracy metrics declining to a certain extent, which is particularly significant for NASCA-LSTM, the proposed attention metric still recovers the correct key accurately, revealing the good robustness of the proposed metric.

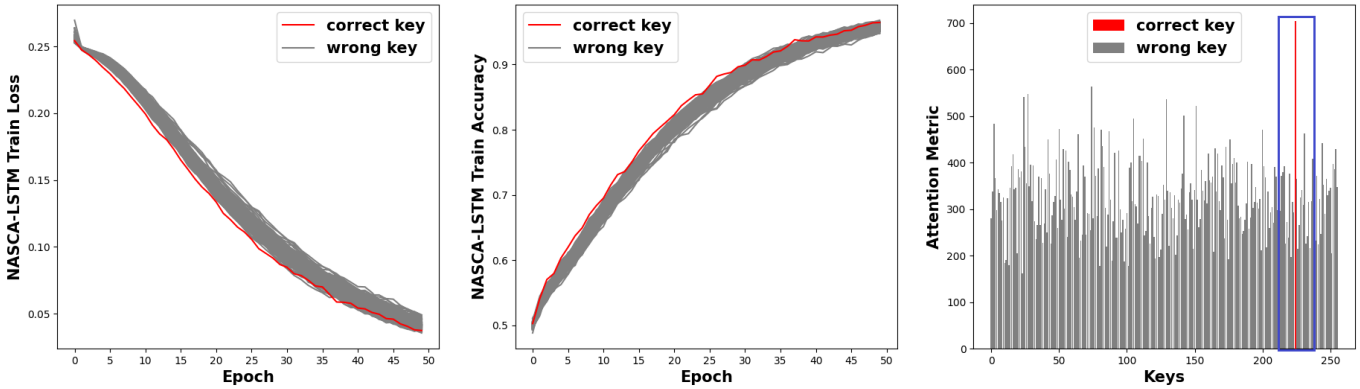

Figure 12. The attack results of NASCA-LSTM on noisy ASCAD with 20,000 traces. ($\sigma = 0.5$).

4.5. Performance of NASCA on AES_RD

The AES_RD dataset employs the desynchronization countermeasure, similar to ASCAD_desync50. However, the 3500 time samples are not specific to the information leakage of a single byte in this dataset but rather involve the information leakage of all 16 bytes. Since power analysis attacks are typically performed byte by byte, performing attack on this dataset poses a significant challenge. To evaluate the robustness of the proposed architecture, both NASCA-CNN and NASCA-LSTM architectures are applied to this dataset, and the first byte is targeted for attack. Both architectures are trained under the same training parameters as described in Section 4.2, except for the number of time samples (3500) in each input traces. Due to the limit of LSTM in handling the desynchronization countermeasure, the number of traces for NASCA-LSTM is twice that of NASCA-CNN.

Like the results shown in Figure 13, NASCA-CNN performs well in both metrics, which is consistent with the findings presented in Section 4.3. The performance of NASCA-CNN remains consistent across different desynchronized datasets, demonstrating its robustness and capability to handle desynchronized data. Since LSTM does not have the same translation-invariance property as CNN, it can be observed that the traditional loss and accuracy metrics lose their functionality as demonstrated in Figure 14. However, the proposed metric still provides reliable results by an obvious peak for correct key. The superiority of the quantitative analysis realized by the attention metric is demonstrated again.

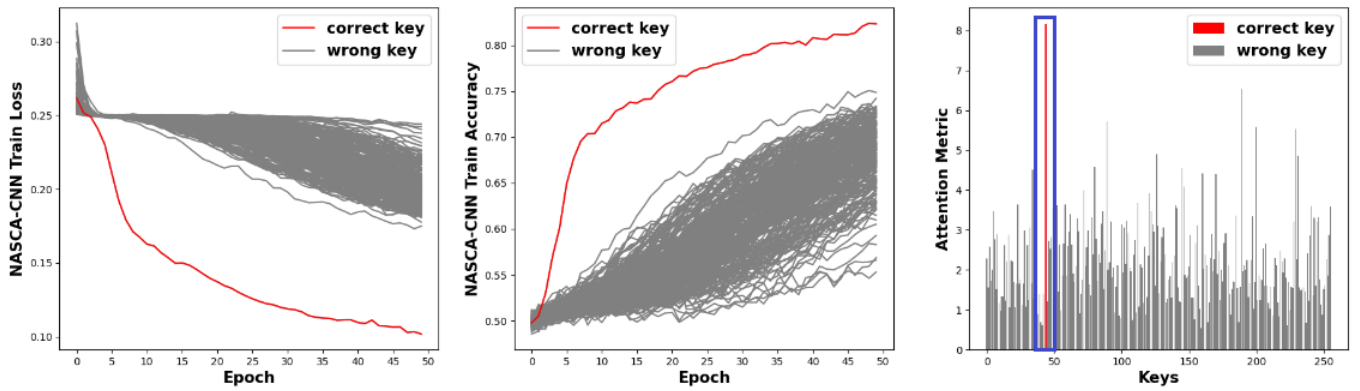

Figure 13. The attack results of NASCA-CNN on AESRD with 10,000 traces.

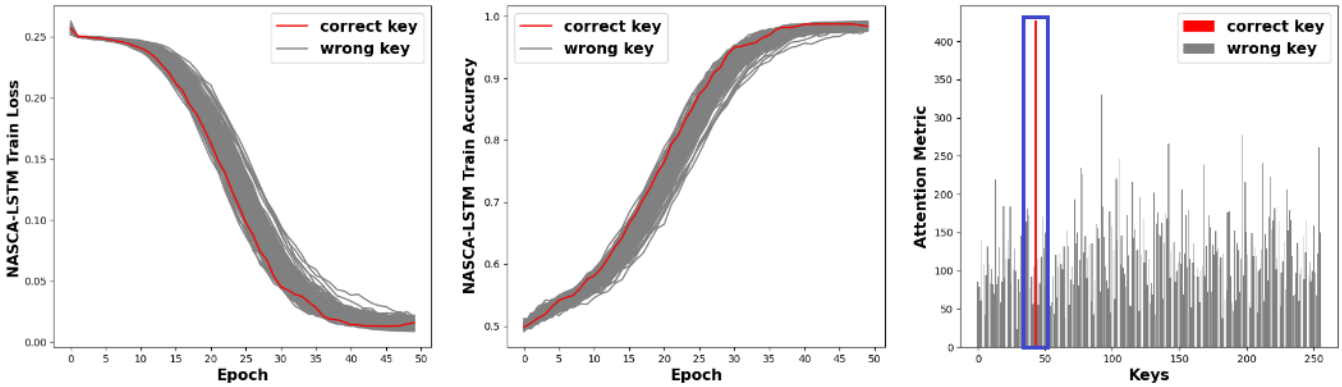

Figure 14. The attack results of NASCA-LSTM on AESRD with 20,000 traces.

4.6. Performance Evaluation

The performance of different metrics in the experiments above is summarized in Table 2. For ASCAD and ASCAD with noise, all the three metrics provide successful attack results. However, when dealing with ASCAD_desync50, loss and accuracy metrics fail, and the proposed metric still succeeds. Regarding AES_RD, loss and accuracy fail to recover the correct key in the LSTM architecture, while the proposed metric succeeds in both architectures.

Table 2. Summary of attack results.

Metric / Dataset	Loss	Accuracy	Proposed Metric
ASCAD	Succeed	Succeed	Succeed
ASCAD_desync50	Fail	Fail	Succeed
ASCAD with 0.2 noise	Succeed	Succeed	Succeed
ASCAD with 0.5 noise	Fail in LSTM	Fail in LSTM	Succeed
AES_RD	Fail in LSTM	Fail in LSTM	Succeed

To provide a more quantitative evaluation of this work, a comparison of the success rate [24] between the proposed architecture state-of-the-art non-profiled SCA methods is shown in Table 3. The success rate refers to the percentage of successful attacks over total attacks. One hundred iterations of attack on each dataset are recorded to obtain the success rate of our architectures. When facing ASCAD, all methods perform well. In the case of ASCAD_desync50, NASCA-CNN provides a higher success rate than DDLA-CNN in [16] and MOR-CNN4 in [23]. When Gaussian noise is introduced to ASCAD, although the performance of all methods experiences a decline, NASCA-CNN still has a slight advantage.

Table 3. Comparison of success rate.

Dataset / Method	ASCAD	ASCAD_desync50	ASCAD with Noise
NASCA-CNN	99%	95%	89%
NASCA-LSTM	98%	-	86%
DDLA-MLP [16]	98%	-	58%
DDLA-CNN [16]	99%	97%	-
MOR-CNN4 [23]	-	84%	-
MOR-MLP3 [23]	98%	-	84%

Additionally, comprehensive evaluations on time complexity, model complexity and traces consumption are summarized in Table 4. Regarding the attack time on ASCAD and model complexity, the two proposed models exhibit moderate performance, while the two MOR models and DDLA-MLP demonstrate the best performance. However, our

proposed models outperform other existing works in terms of trace consumption against mask countermeasures.

Table 4. Summary of comprehensive evaluations.

Mehtod	Attack Time(s) on ASCAD	Model Complexity	Trace Consumption
NASCA-CNN	4133	moderate	10,000
NASCA-LSTM	4090	moderate	10,000
DDLA-MLP [16]	1200	low	20,000
DDLA-CNN [16]	11030	low	20,000
MOR-CNN4 [23]	270	low	40,000
MOR-MLP3 [23]	700	moderate	40,000
ACNN [18]	-	high	100,000

5. Conclusions

In this paper, a novel architecture called NASCA, including an attention metric, is proposed for non-profiled side-channel attacks. With the incorporation of an attention mechanism, this innovative architecture effectively leverages long-term power traces for attacks. The experiments demonstrate that the proposed NASCA consistently provides reliable results under various countermeasures, whereas traditional metrics fail in certain cases. By attaching the attention mechanism after the feed-forward networks, which are realized by CNN and LSTM in this paper, the attention mechanism automatically identifies the most informative feature vectors during the network-training process. When the network converges, the attention metrics show a distinct peak, which indicates the correct key. In the future work, some feature extraction methods can be studied to reduce the time samples required for deep learning. Additionally, we aim to investigate the impact of hyperparameters and training strategies, such as early stopping or learning rate decay, on the performance of NASCA.

Author Contributions: Conceptualization, methodology and writing—original draft preparation, K.P.; software and writing—review and editing, K.P. and H.D.; data curation and validation, K.P., H.D. and F.K.; visualization and project administration, K.P., W.W. and J.Z.; supervision and investigation, H.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: All data can be provided upon reasonable request to the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Kocher, P.C. Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In Proceedings of the Advances in Cryptology—CRYPTO'96: 16th Annual International Cryptology Conference, Santa Barbara, CA, USA 18–22 August 1996; Springer: Berlin/Heidelberg, Germany, 1996; pp. 104–113.
2. Chari, S.; Rao, J.R.; Rohatgi, P. Template attacks. In *Revised Papers 4, Proceedings of the Cryptographic Hardware and Embedded Systems-CHES 2002: 4th International Workshop, Redwood Shores, CA, USA, 13–15 August 2002*; Springer: Berlin/Heidelberg, Germany, 2003; pp. 13–28.
3. El Aabid, M.A.; Guilley, S.; Hoogvorst, P. *Template Attacks with a Power Model*; IACR Cryptology ePrint Archive. 2007. Available online: https://eprint.iacr.org/2007/443 (accessed on 1 June 2023).
4. Schindler, W.; Lemke, K.; Paar, C. A stochastic model for differential side channel cryptanalysis. In Proceedings of the Cryptographic Hardware and Embedded Systems—CHES 2005: 7th International Workshop, Edinburgh, UK, 29 August–1 September 2005; Springer: Berlin/Heidelberg, Germany, 2005; pp. 30–46.
5. Mangard, S. A simple power-analysis (SPA) attack on implementations of the AES key expansion. In Proceedings of the Information Security and Cryptology—ICISC 2002: 5th International Conference, Seoul, Republic of Korea, 28–29 November 2002; Springer: Berlin/Heidelberg, Germany, 2003; pp. 343–358.

6. Kocher, P.; Jaffe, J.; Jun, B. Differential power analysis. In Proceedings of the Advances in Cryptology—CRYPTO'99: 19th Annual International Cryptology Conference, Santa Barbara, CA, USA, 15–19 August 1999; Springer: Berlin/Heidelberg, Germany, 1999; pp. 388–397.
7. Brier, E.; Clavier, C.; Olivier, F. Correlation power analysis with a leakage model. In Proceedings of the Cryptographic Hardware and Embedded Systems—CHES 2004: 6th International Workshop, Cambridge, MA, USA, 11–13 August 2004; Proceedings 6; Springer: Berlin/Heidelberg, Germany, 2004; pp. 16–29.
8. Maghrebi, H.; Portigliatti, T.; Prouff, E. Breaking cryptographic implementations using deep learning techniques. In Proceedings of the Security, Privacy, and Applied Cryptography Engineering: 6th International Conference, SPACE 2016, Hyderabad, India, 14–18 December 2016; Springer: Berlin/Heidelberg, Germany, 2016; pp. 3–26.
9. Maghrebi, H. *Deep Learning Based Side Channel Attacks in Practice*; Cryptology ePrint Archive 2019. Available online: https://eprint.iacr.org/2019/578 (accessed on 1 June 2023).
10. Lerman, L.; Markowitch, O. Efficient profiled attacks on masking schemes. *IEEE Trans. Inf. Forensics Secur.* **2018**, *14*, 1445–1454. [CrossRef]
11. Nassar, M.; Souissi, Y.; Guilley, S.; Danger, J.L. RSM: A small and fast countermeasure for AES, secure against 1st and 2nd-order zero-offset SCAs. In Proceedings of the 2012 Design, Automation & Test in Europe Conference & Exhibition (DATE), Dresden, Germany, 12–16 March 2012; pp. 1173–1178.
12. Veyrat-Charvillon, N.; Medwed, M.; Kerckhof, S.; Standaert, F.X. Shuffling against side-channel attacks: A comprehensive study with cautionary note. In Proceedings of the Advances in Cryptology—ASIACRYPT 2012: 18th International Conference on the Theory and Application of Cryptology and Information Security, Beijing, China, 2–6 December 2012; Springer: Berlin/Heidelberg, Germany, 2012; pp. 740–757.
13. Coron, J.S.; Kizhvatov, I. An efficient method for random delay generation in embedded software. In Proceedings of the Cryptographic Hardware and Embedded Systems—CHES 2009: 11th International Workshop, Lausanne, Switzerland, 6–9 September 2009; Springer: Berlin/Heidelberg, Germany, 2009; pp. 156–170.
14. Dao, B.A.; Hoang, T.T.; Le, A.T.; Tsukamoto, A.; Suzaki, K.; Pham, C.K. Correlation Power Analysis Attack Resisted Cryptographic RISC-V SoC With Random Dynamic Frequency Scaling Countermeasure. *IEEE Access* **2021**, *9*, 151993–152014. [CrossRef]
15. Jin, S.; Kim, S.; Kim, H.; Hong, S. Recent advances in deep learning-based side-channel analysis. *ETRI J.* **2020**, *42*, 292–304. [CrossRef]
16. Timon, B. Non-profiled deep learning-based side-channel attacks with sensitivity analysis. *Iacr Trans. Cryptogr. Hardw. Embed. Syst.* **2019**, *2019*, 107–131. [CrossRef]
17. Xiangliang, M.; Bing, L.; Hong, W.; Di, W.; Lizhen, Z.; Kezhen, H.; Xiaoyi, D. Non-profiled Deep-Learning-Based Power Analysis of the SM4 and DES Algorithms. *Chin. J. Electron.* **2021**, *30*, 500–507. [CrossRef]
18. Lu, X.; Zhang, C.; Gu, D. Attention-Based Non-Profiled Side-Channel Attack. In Proceedings of the 2021 Asian Hardware Oriented Security and Trust Symposium (AsianHOST), Shanghai, China, 16–18 December 2021; pp. 1–6.
19. Do, N.T.; Hoang, V.P.; Doan, V.S. Performance analysis of non-profiled side channel attacks based on convolutional neural networks. In Proceedings of the 2020 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS), Ha Long, Vietnam, 8–10 December 2020; pp. 66–69.
20. Raffel, C.; Ellis, D.P. Feed-forward networks with attention can solve some long-term memory problems. *arXiv* **2015**, arXiv:1512.08756.
21. Peng, H.; Pappas, N.; Yogatama, D.; Schwartz, R.; Smith, N.A.; Kong, L. Random feature attention. *arXiv* **2021**, arXiv:2103.02143.
22. Kuroda, K.; Fukuda, Y.; Yoshida, K.; Fujino, T. Practical aspects on non-profiled deep-learning side-channel attacks against AES software implementation with two types of masking countermeasures including RSM. In Proceedings of the 5th Workshop on Attacks and Solutions in Hardware Security, Virtual Event, Republic of Korea, 19 November 2021; pp. 29–40.
23. Do, N.T.; Hoang, V.P.; Doan, V.S. A novel non-profiled side channel attack based on multi-output regression neural network. *J. Cryptogr. Eng.* **2023**, 1–13. [CrossRef]
24. Dol, N.T.; Le, P.C.; Hoang, V.P.; Doan, V.S.; Nguyen, H.G.; Pham, C.K. Mo-dlsca: Deep learning based non-profiled side channel analysis using multi-output neural networks. In Proceedings of the 2022 International Conference on Advanced Technologies for Communications (ATC), Hanoi, Vietnam, 20–22 October 2022; pp. 245–250.
25. Won, Y.S.; Han, D.G.; Jap, D.; Bhasin, S.; Park, J.Y. Non-profiled side-channel attack based on deep learning using picture trace. *IEEE Access* **2021**, *9*, 22480–22492. [CrossRef]
26. Daemen, J.; Rijmen, V. Reijndael: The advanced encryption standard. *Dr. Dobb'S J. Softw. Tools Prof. Program.* **2001**, *26*, 137–139.
27. Rioja, U.; Batina, L.; Flores, J.L.; Armendariz, I. Auto-tune POIs: Estimation of distribution algorithms for efficient side-channel analysis. *Comput. Netw.* **2021**, *198*, 108405. [CrossRef]
28. Rolnick, D.; Veit, A.; Belongie, S.; Shavit, N. Deep learning is robust to massive label noise. *arXiv* **2017**, arXiv:1705.10694.
29. Khan, S.; Rahmani, H.; Shah, S.A.A.; Bennamoun, M. A guide to convolutional neural networks for computer vision. *Synth. Lect. Comput. Vis.* **2018**, *8*, 1–207.
30. Kayhan, O.S.; Gemert, J.C.v. On translation invariance in cnns: Convolutional layers can exploit absolute spatial location. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 14274–14285.

31. Cagli, E.; Dumas, C.; Prouff, E. Convolutional neural networks with data augmentation against jitter-based countermeasures: Profiling attacks without pre-processing. In Proceedings of the Cryptographic Hardware and Embedded Systems—CHES 2017: 19th International Conference, Taipei, Taiwan, 25–28 September 2017; Springer: Berlin/Heidelberg, Germany, 2017; pp. 45–68.
32. Hua, W.; Zhang, Z.; Suh, G.E. Reverse engineering convolutional neural networks through side-channel information leaks. In Proceedings of the 55th Annual Design Automation Conference, San Francisco, CA, USA, 24–29 June 2018; pp. 1–6.
33. Li, Y.; Zeng, J.; Shan, S.; Chen, X. Occlusion aware facial expression recognition using CNN with attention mechanism. *IEEE Trans. Image Process.* **2018**, *28*, 2439–2450. [CrossRef] [PubMed]
34. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
35. Elman, J.L. Finding structure in time. *Cogn. Sci.* **1990**, *14*, 179–211. [CrossRef]
36. Whitnall, C.; Oswald, E.; Standaert, F.X. The Myth of Generic DPA...and the Magic of Learning. In Proceedings of the Topics in Cryptology—CT-RSA 2014: The Cryptographer's Track at the RSA Conference 2014, San Francisco, CA, USA, 25–28 February 2014; Springer: Berlin/Heidelberg, Germany, 2014; Volume 8366, pp. 183–205.
37. Lu, X.; Zhang, C.; Cao, P.; Gu, D.; Lu, H. Pay attention to raw traces: A deep learning architecture for end-to-end profiling attacks. *Iacr Trans. Cryptogr. Hardw. Embed. Syst.* **2021**, *2021*, 235–274. [CrossRef]
38. Benadjila, R.; Prouff, E.; Strullu, R.; Cagli, E.; Dumas, C. Deep learning for side-channel analysis and introduction to ASCAD database. *J. Cryptogr. Eng.* **2020**, *10*, 163–188. [CrossRef]