Performance Analysis of Non-profiled Side Channel Attack Based on Multi-layer Perceptron Using Significant Hamming Weight Labeling

Ngoc-Tuan Do[1], Van-Phuc Hoang[1(✉)], and Van Sang Doan[2]

[1] Institute of System Integration, Le Quy Don Technical University, Hanoi, Vietnam
phuchv@lqdtu.edu.vn

[2] Faculty of Communications and Radar, Vietnam Naval Academy, Nha Trang, Vietnam

Abstract. Deep learning (DL) techniques have become popular for side-channel analysis (SCA) in the recent years. This paper proposes and evaluates the applications of multilayer perceptron (MLP) models for non-profiled attacks on the AES-128 encryption implementation in different scenarios, such as high dimensional data, imbalanced classes, and the impact of additive noise. Along with the designed models, a labeling technique called significant Hamming weight (SHW) and dataset reconstruction method are introduced for solving the imbalanced dataset problem. In addition, using SHW in the non-profiled context can reduce the number of measurements needed by approximately 30%. The experimental results show that the DL based SCA with our reconstructed dataset for different targets of ASCAD, RISC-V microcontroller has achieved a higher performance of non-profiled attacks. Comparing to the binary labeling technique, SHW labeling provides better results with the presence of the additive noise.

Keywords: Hardware security · Non-profile side channel attack · Advanced Encryption Standard (AES) · Multilayer perceptron (MLP) · Imbalanced classes

1 Introduction

Side channel attack (SCA) is a serious threat, which exploits weaknesses in the physical implementation of cryptographic system. Recently, the hardware security research communities have focused their attention on deep learning (DL) based SCA (DLSCA). This is a promising technique for implementing powerful attacks against cryptographic devices. In general, DL based SCA is divided into profiling attacks and non-profiling attacks according to the attack environment [2].

Profiling DLSCA is an attack that can be implemented when the clone device, which is the same device as the target device, is available. In this case, the

N.-S. Vo et al. (Eds.): INISCOM 2022, LNICST 444, pp. 241–254, 2022.
https://doi.org/10.1007/978-3-031-08878-0_17

attacker has a full control over the clone device. In particular, profiled DLSCA functions in two phases:
- *Profiled phase:* For the profiling phase, the power traces of the clone device are obtained for training a neural network. A template model is then constructed, which is able to identify the correlation between the intermediate values of the cryptography algorithm and the power traces of the target device.
- *Attack phase:* the trained model is used to analyze the power trace recorded from the actual target device.

In contrast, the non-profiled DLSCA can be performed directly on the target device without profiled phase and clone device. Indeed, the authors in TCHES 2019 [15] presented the efficiency of DL based SCA in the non-profiled attack. Accordingly, they introduced the metrics based on sensitivity analysis to extract the confidential key value and the POIs (such as masks and leaks positions in power traces). This attack technique was so-called differential DL analysis (DDLA). More especially, the authors provided two labeling techniques like Hamming weight labeling and Binary labeling. The efficiency of this technique was proved on both types of synchronized and non-synchronized power traces.

Even thought profiled attacks are considered the most powerful form of side-channel attacks, the profiling phase requires to have access to a profiling device, which is a strong assumption that cannot be always met in practice. Interestingly, non-profiled attacks like DDLA can still threaten the device. Therefore, we are more interested in DL based non-profiled attacks, especially DDLA technique.

1.1 Related Works and Motivation

While proven as a successful attack, DDLA has some problems that need to take into account.

Firstly, DDLA is necessary to perform a DL training for each key guess. It means a complex architecture will cause the time-consuming for taking the secret key, especially in the case of high dimension data input. The authors in [15] have not considered the high dimension data in DL based non-profiled SCA context. Their experiments used a small number of measurements with a few hundred samples per trace that contained the copies of S-box function in memory. Moreover, the authors introduced a technique called sensitivity analysis based on network input to reveal the correct key. Therefore, the higher the dimension data input, the more complex the model. It leads to an increased computation time for each hypothesis key.

In general, several techniques have been proposed to deal with high dimension data input problem in DLSCA. Picek *et al.* [13] used the correlation analysis to extract the most relevant samples. This technique exploits the correlation between power traces with a power consumption model. Another technique called principle component analysis (PCA) which is usually used in deep learning. However, the main drawback of PCA is that the calculation time increases quadratically relative to the number of samples on a power trace. Despite the efficiency of reducing the data dimension, these techniques mentioned above are only applied in a profiled context.

Secondly, the effectiveness of DDLA depends on data labeling techniques. As indicated in [15], two main labeling techniques (Hamming weight and binary) have been applied in a non-profiled context. In which, the efficiency of binary labeling method is proven in many works [1,15,16]. In contrast, HW labeling has not been considered. Moreover, the authors in [16] have also indicated that HW model in non-profiled SCA causes the imbalance dataset problem. To the best of our knowledge, only one report of using HW labeling method has been published in non-profile deep learning context [4].

Finally, DL based non-profiled SCA technique is sensitive with the additive noise. In profiling DLSCA context, several works have investigated the effect of noise addition [8,11]. Kim et al. have demonstrated in [8] that adding the artificial noise on the input signal can actually improve the performance of neural networks. In [11], Maghrebi has proved that noise addition can avoid over-fitting in DL-based SCA techniques. In contrast, the authors in [1] have indicated that a additive noise may provide higher protection against non-profiled DLSCA. Therefore, it is necessary to proposed a DL model against noise generation countermeasure in a non-profiled context.

Due to the mentioned issues above, the motivation of this research is to introduce a new type of HW labeling technique which can deal with the imbalanced dataset problems. Furthermore, based on the correlation between HW labels and the raw data input, we can reduce the dimension of data input significantly. Consequently, a new multi-layer perceptron model working on new dataset reconstructed is proposed.

The contributions of this paper can be summarized as follows:
- We proposed a new type of HW label based on significant Hamming weight (SHW) to fight against imbalance classes in a non-profiled context. We demonstrate SHW labeling works in both TCHES 2019 model and our proposed architecture. More interestingly, by using SHW labeling, the number of measurements needed for DLSCA attack reduces approximately 30% compared to Binary labeling.
- We introduce a new multi-layer perceptron model instance that applied SHW labeling to reveal the correct key in a non-profiled context. Compared to the state-of-the-art non-profiled DLSCA model (TCHES 2019), our proposed shows better results in discriminating correct key.
- The balancing technique based on SHW enables us to reach better results than Binary labeling with the presence of additive noise.

1.2 Paper Outline

The rest of this paper is organized as follows. In Sect. 2, data preparation, including test platforms, imbalanced data problems, dataset reconstruction based on SHW labeling are described in detail. Section 3 presents our proposed MLP architectures which applied SHW labeling to reveal the correct key. In Sect. 4, we will give detailed results from various experiments implemented on raw power traces collected from RISC-V MCU or ASCAD database. Finally, we conclude the paper in Sect. 5.

2 Data Preparation

2.1 Attack Datasets

As we dealing with the imbalanced classes in DL based non-profiled SCA context. Our proposed labeling method is based on leakage function. We are more interested in the first-order leakage rather than higher order. Therefore, countermeasures like masking are out of scope in this paper. For experimental validation, we use two first-order leakage datasets like unmasked ASCAD and RISC-V data.

Unmasked ASCAD Dataset: ASCAD is a public dataset which is introduced by Prouff *et al.* [14]. This database provides side-channel power traces of an 8-bit ATMega8515 board with a first-order protected software AES implementation. The main database ASCAD is composed of two sets of traces: a profiling set of 50,000 traces to train Deep Learning architectures and an attack set of 10,000 traces to test the efficiency of the trained neural networks. It is worth noting that 700 samples corresponding to the output of the third Sbox being processed during the first round. As we are interested in an unmasked implementation, we consider the mask to be known and thus can easily turn it into an unprotected scenario. Accordingly, the leakage model is calculated as follows:

$$H_{p_3,k} = HW\left(Sbox\left[p_3 \oplus \boldsymbol{k}^*\right] \oplus \underbrace{m_3}_{\text{known mask}}\right) \tag{1}$$

where p_3, $\boldsymbol{k}^*$ and m_3 are respectively the third byte of plaintext, the key and the third byte of known mask.

RISC-V Dataset: Next, the power traces of Sakura-G board with unprotected AES-128 are captured using an oscilloscope at sampling rate of 250 MS/s. It is worth noting that the target device is a 32-bit RISC-V MCU Murax operating at 48 MHz, as shown in Fig. 1. Accordingly, 10,000 power traces of the Sakura-G board are collected. Each power trace contains 9,919 samples. In this platform, we chose the sixth byte for investigating proposed techniques. The leakage model on RISC-V power traces is similar to (1) without mask value, see (2):

$$H_{p_6,k} = HW\left(Sbox\left[p_6 \oplus \boldsymbol{k}^*\right]\right) \tag{2}$$

2.2 Imbalanced Data Problem

Imbalanced data is an issue often occurring in classification applications where the distribution of classes is not balanced. In such a case, deep learning classification algorithms (e.g., multi-layer perceptron, convolutional neural network, etc.) have difficulties since they will be biased towards the majority class.

In profiling SCA context, Pikek *et al.* in [12] indicated that the commonly used Hamming weight model leads to imbalanced training datasets. Indeed, by observing Table 1, it is obvious that the distribution of intermediate values

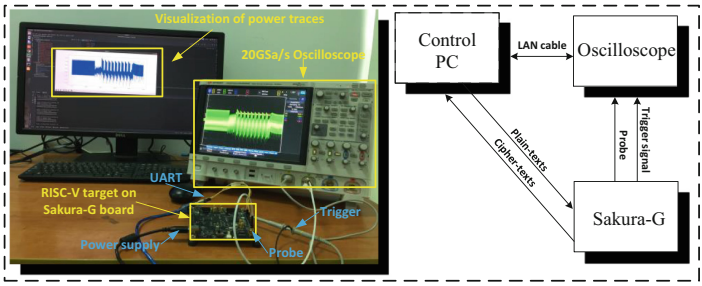

Fig. 1. Test platform: RISC-V power traces acquisition on Sakura-G board.

on each HW is imbalanced and symmetric about $HW4$ in the case of AES-128. To tackle this problem, the authors in [12] proposed a method based on the data re-sampling technique. Accordingly, they used a random oversampling method called SMOTE to oversamples for each class. In practice, SMOTE can be considered as a general case of the data augmentation (AU) technique proposed in [3].

Table 1. Probability distribution of the HW of a uniformly distributed 8-bit value.

HW	0	1	2	3	4	5	6	7	8
Probability	$\frac{1}{256}$	$\frac{8}{256}$	$\frac{28}{256}$	$\frac{56}{256}$	$\frac{70}{256}$	$\frac{56}{256}$	$\frac{28}{256}$	$\frac{8}{256}$	$\frac{1}{256}$

2.3 Significant HW Labeling

Similarly, non-profiling SCA like DDLA used HW model also suffers from imbalanced data problems. However, unlike profiled DL based SCA, DDLA uses the training metrics instead of the model's output for discriminating the correct key. According to [15], training with the correct key always has better learning ability than incorrect ones. It means that if we use only three classes for training instead of nine classes, the DL model using correct key still has better training metrics than wrong key. As shown in Table 1, there are three significant HW ($HW = 3, 4, 5$) that contain the most distribution of intermediate values. Moreover, the distribution of three significant HWs (SHW) are nearly the same. Therefore, we assume that it is possible to use SHW for classification in a non-profiled context.

Apart from balancing data, SHW labeling reduces significant measurements needed for the attack. Indeed, from Table 1, it is clearly shown that SHW discards the intermediate values corresponding to $HW = 0, 1, 2, 6, 7, 8$. It is meaningful in the case of using adaptive chosen plaintext method in a non-profiled context [7]. Next, we will present the procedure in order to reconstruct new datasets from original ones for using SHW labeling.

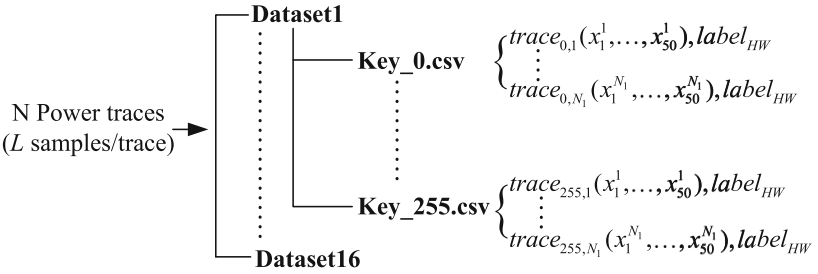

Fig. 2. Structure of the new datasets: There are 16 folders (**Dataset1 to Dataset16**) corresponding to 16 bytes of secret key, each folder contains 256 files in **.csv** format which correspond to 256 hypothesis keys. N original power traces (L samples/trace) are calculated to form N_1 new traces and labeled ($HW = \{3, 4, 5\}$). Each new trace contains 50 samples which are highest correlation values.

2.4 Dataset Reconstruction

As mentioned in the previous section, non-profiled attacks do not need the profiling phase as profiled attacks; therefore, the datasets that are used for deep learning in non-profiled attacks are different. It means that we can use power trace values as input for training neural networks, which can learn to classify the power traces in the group that corresponds to the power consumption model. However, a power trace often contains thousands of samples, whereas only a part of them serves for key prediction. Therefore, using all of these samples as the input features of ML model leads to an increment in the ML complexity and the time-consuming [6,10,13].

In order to handle this issue, we use the correlation characteristic to find the most useful samples on a power trace to feed up the neural network. Recently, in our previous work [5], correlation coefficients are employed for taking the most relevant samples in the power trace by computing correlation between real traces with their model. This method is suitable for a large number of samples because attackers do not need to know detail about the AES implementation. Accordingly, the samples which have high correlation values with the power model will be selected as strong features for training a neural network. As mentioned above, HW model is used as a power consumption model. However, we use formula (1) and (2) for ASCAD dataset and RISC-V data, respectively.

To select the useful features, Pearson correlation coefficients formula is applied:

$$\rho_{k,i} = \frac{\sum_{n=1}^{N'} (h_{n,k} - \bar{h}_k)(t_{n,i} - \bar{t}_i)}{\sqrt{\sum_{n=1}^{N'} (h_{n,k} - \bar{h}_k)^2 \sum_{n=1}^{N'} (t_{n,i} - \bar{t}_i)^2}} \tag{3}$$

where $\bar{h}_k$ and $\bar{t}_i$ are the average values of the power consumption model and real power at instant i, respectively.

Let's consider N random plaintexts corresponding to N power traces in which each power trace has L samples. It is noted that $t_{i,j}$ is the value of j^{th} sample in the i^{th} trace ($1 \leq j \leq L$, $1 \leq i \leq N$), $d_{i,B}$ is the byte value of byte B ($B \in [1;16]$) in the i^{th} plain-text. In order to collect the useful features from the power traces, a half of power traces is used and denoted as $N'(N' = \frac{1}{2}N)$. Afterward, the formula (3) is applied to select the samples of high correlation from all guess keys ($Key = [0;255]$).

For determining the positions of high correlation, the correlation value of real power trace with its model is calculated on half of power traces. As a result, a matrix of correlation coefficients with a size of $256 \times L$ is produced, in which each row corresponds to a hypothesis key. By determining 50 useful sample points based on the 50-top highest correlation values, a smaller dataset of power traces is generated and reconstructed follows hypothesis keys and HW values, in which HWs plays a role of labels as shown in Fig. 2. By this way, a numerous power traces corresponding to other remain HW values can be ignored; therefore, the dataset continues to be reduced (about one-third). Three new datasets are reconstructed from the original ASCAD database and RISC-V data as presented in Table 2.

Table 2. The details of reconstructed dataset.

Dataset	Unmasked ASCAD		RISC-V data		Labeling technique
	Number of traces	Dimension	Number of traces	Dimension	
Original	3000	700	10000	9919	LSB
Dataset 1	≈2000	700	x	x	SHW
Dataset 2	≈2000	50	x	x	SHW
Dataset 3	x	x	≈7000	50	SHW

3 Proposed Deep Learning Architecture

As demonstrated in [15], multi-layer perceptron (MLP) is the simple and efficient architecture to perform DDLA on synchronous power traces. In this work, we propose a new MLP instance for non-profiled SCA attack. Our novelty is to apply the SHW labeling technique. In addition, we used Exponential Linear Unit (ELU) activation function instead of RELU as in TCHES2019. The details of the proposed architecture are depicted in Fig. 3.

Our proposed network comprises an input layer, output layers, and six hidden layers. As presented in Sect. 2, the reconstructed dataset consists of 50 samples each power trace. The number of nodes in the input layer is assigned according to the number of samples in a power trace. Therefore, the input layer of the proposed MLP has 50 nodes corresponding to 50 features of the extracted power trace. As depicted in Fig. 3, all arrows represent the weights. Prior to implement training phase, the values of weights and bias are randomly chosen from a normal distribution using Xavier scheme.

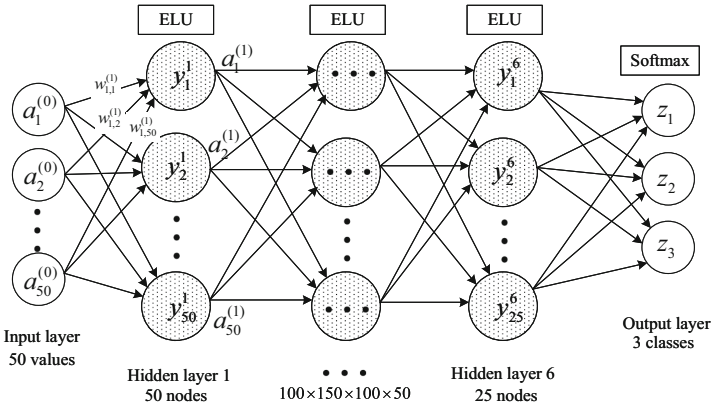

Fig. 3. The proposed multi-layer perceptron architecture.

To obtain the output, a procedure called *forward propagation* is performed. *Forward propagation* can be implemented as illustrated in Fig. 3, the weighted sum is calculated as follows:

$$y_i^{(l)} = b_i + \sum_{j=1}^{l^{(l-1)}} a_j^{(l-1)} \times w_{ji}^l \tag{4}$$

where b_i is the bias of node i^{th}, w_{ji}^l is the weight which connect node i^{th} of layer $l-1$ to node j^{th} of layer l, $a_j^{(l-1)}$ is the output of activation function $\boldsymbol{F}(y)$ on node j^{th} of layer $l-1$ and calculated as (5).

$$a_j^{(l)} = \boldsymbol{F}(y_j^{(l)}) \tag{5}$$

It is worth noting that formula (5) does not apply on the input layer. The output $a_j^{(l)}$ is then used as the input of the next neuron on next layer. This procedure is performed from the input layer to the output layer.

In DL based SCA context, the popular activation functions used in hidden layer are ELU and RELU which are computed as formula (6) and (7), respectively. Our proposed model used ELU instead of ReLU to avoid the vanishing problem and produce negative outputs for each node in the hidden layer.

$$\text{ReLU}: F_{(y)} = \begin{cases} y: & y > 0 \\ 0: & y \leq 0 \end{cases} \tag{6}$$

$$\text{ELU}: F_{(y)} = \begin{cases} y: & y > 0 \\ \alpha \cdot (e^y - 1): & y \leq 0 \end{cases} \tag{7}$$

For classification, the Softmax function is used in the output layer for calculating the probability of each HW label. This function is calculated as

$$\text{SoftMax}: z\,(y)\,[i] = \frac{e^{y[i]}}{\sum_{j=1}^{K} e^{y[j]}} \tag{8}$$

where K is number of classes, in our case, $K = 3$ since our proposed model uses SHW label.

Finally, *backward propagation* is implemented in order to update the weights to obtain the expected results. Since we have three labels, the categorical crossentropy loss between the labels and predictions are computed as

$$\mathcal{L}_X(\boldsymbol{w}) = -\sum_{j=1}^{3} y_{true} \ln(z) \tag{9}$$

where y_{true} is the encoded values of HW.

Then, we use Adam optimizer to find the optimal minimizing the loss function. Deep learning will do a series of iteration t, and in each iteration, the gradient of loss function $\nabla\mathcal{L}_X(\boldsymbol{w})$ is computed. After that, $\boldsymbol{w}$ is updated by using the following formula described in [9]:

$$\boldsymbol{w}_t = \boldsymbol{w}_{t-1} - \alpha \cdot \hat{m}_t \Big/ \left(\sqrt{\hat{v}_t} + \varepsilon\right) \tag{10}$$

where α is called the learning rate. In our case, α is chosen equal 0.001, other parameters are chosen as recommended in [9] ($\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\varepsilon = 10^{-8}$)

When the correct hypothesis key $\boldsymbol{k^*}$ is used, the series of intermediate results will be correctly computed. Consequently, the partition and the labels used for our model will be consistent with the corresponding traces. In contrast, for all the incorrect guess keys, the labels used for the training will be incompatible with the traces. By analyzing and optimizing the model, we decided to use six layers with the number of nodes as depicted in Fig. 3. Consequently, our model provide better results like lower loss or higher accuracy than the other candidates. Therefore, the correct key can be obtained easily.

4 Experimental Results

In our experiment, the reconstructed datasets in Sect. 2 are used to perform training the proposed models. We use three models to obtain the results. Firstly, we reconstruct the MLP architecture in [15] called MLP_{ref1}. Next, we use the same as MLP_{ref1}, but SHW labeling is used instead of Binary labeling, denotes as MLP_{ref2}. Finally, we use proposed MLP called $MLP_{proposed}$. To demonstrate the effectiveness of reconstructed datasets and neural networks, the training metrics such as accuracy and loss are used as criterion. All experiments are performed on Keras framework running on a personal computer (Intel Core i5-9500 CPU, DDR4 24GB memory). We provide the detailed experiment setup in Table 3.

Firstly, we use the unmasked ASCAD database to validate the efficiency of SHW labeling techniques. It is important to note that the dimension of data input of MLP_{ref1} is the same as the original one, while MLP_{ref2} and $MLP_{proposed}$ use the reconstruction data. According to Table 2, MLP_{ref1} is

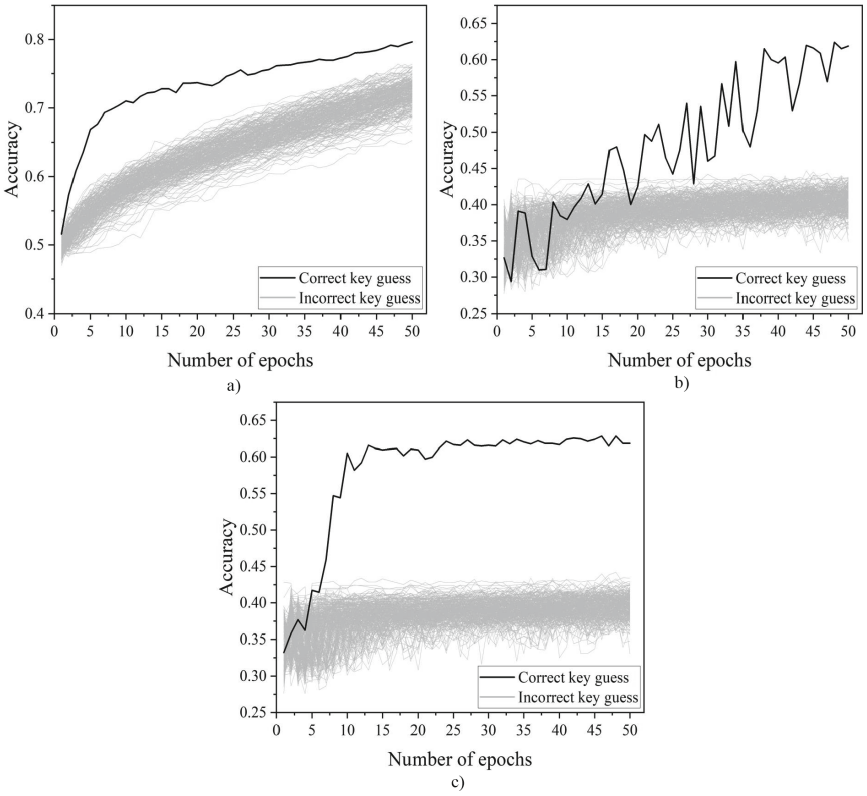

Fig. 4. The attack results on ASCAD database with different models using SHW and LSB labeling a) MLP_{ref1} using LSB; b) MLP_{ref2} using SHW; c) $MLP_{proposed}$ using SHW.

trained by original unmasked ASCAD data, MLP_{ref2} and $MLP_{proposed}$ are trained by Dataset1 and Dataset2, respectively. The attack results on the third byte are illustrated in Fig. 4 where we can see that by using the SHW labels, the model has lower accuracy than MLP_{ref1} because they using only two labels, which leads to results for classification is at least 50%. Despite lower accuracy, SHW labeling method gives a high probability for discriminating the correct key than the model in MLP_{ref1}. Moreover, the result presented in Fig. 4a shows that MLP_{ref1} adapts to the training set too well on both correct key and incorrect key. In contrast, MLP_{ref2} and $MLP_{proposed}$ only adjust training metric on the correct key. More interestingly, the result of MLP_{ref2} indicates that SHW works for TCHES 2019 architecture. However, the accuracy is fluctuate and it takes more epochs (about 20 epochs) to discriminate correct key compared to MLP_{ref1} and $MLP_{proposed}$ (first 5 epochs).

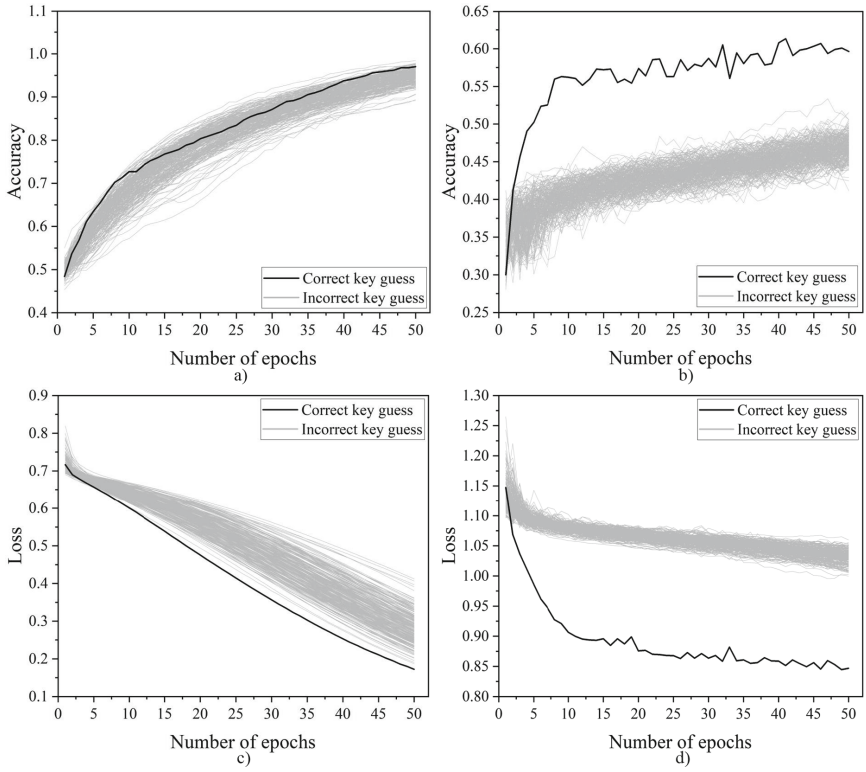

Fig. 5. The attack results of models using LSB and SHW labeling technique on Gaussian noise added ASCAD data (Standard deviation of noise= 3.0) a,c) Accuracy and loss of MLP_{ref1} using LSB label; b,d) Accuracy and loss of $MLP_{proposed}$ using SHW label.

For investigating the effect of noise, we performed further experiments, including adding Gaussian noise to power traces and training our model again. We assume raw power traces from ASCAD database are low noise, and they have no impact of measurement equipment. Our method for adding noise is different with [8,11] that the Gaussian noise is added directly on the power traces. It means that we try to simulate the real scenario when an attacker performs the power measurements. The experiment results are shown in Fig. 5. In this case, we observe both the accuracy and loss metrics. It clearly shows that MLP_{ref1} can not distinguish correct key based on accuracy. In contrast, $MLP_{proposed}$ using SHW label provides good performance. As depicted in Fig. 5(b,d), the correct key is easy to be discriminated from the rest. In addition, we can observe that the loss metric gives better results. Therefore, loss metric is the main criterion in our next experiments.

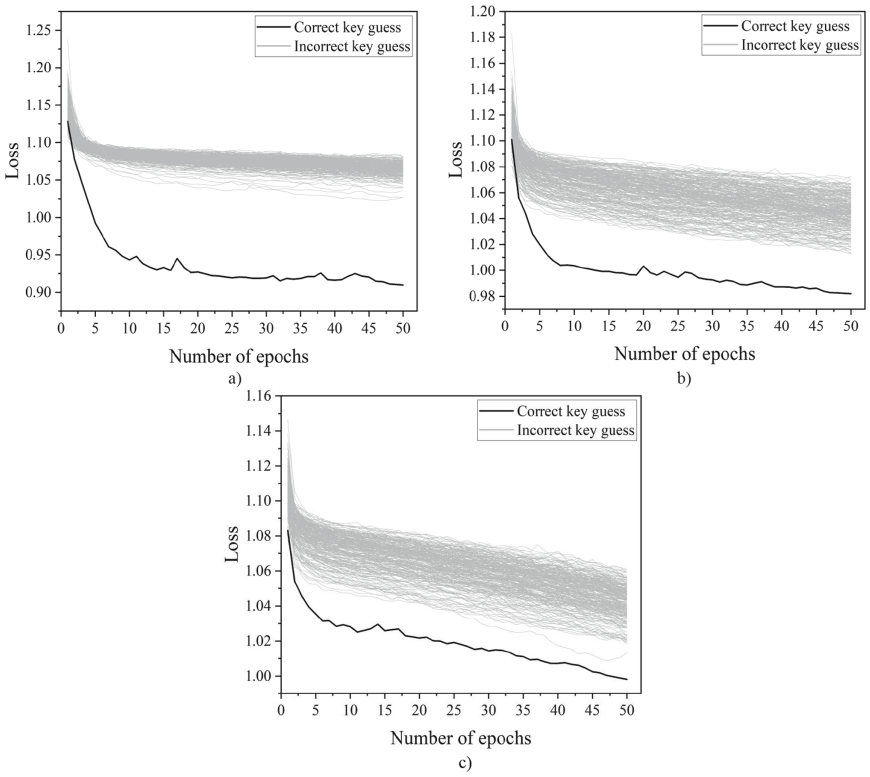

Fig. 6. The attack results on RISC-V data with different standard deviation of additive noise a) 0; b) 0.002; c) 0.004.

To prove the efficiency of SHW labeling and proposed ML architecture on different platforms, we perform the same experiment above on the power trace collected from a 32-bit RISC-V processor, as described in Sect. 2. We will investigate the adaption of those techniques for the new dataset. As a result shown in Fig. 6, it is clear that our proposed MLP models have good performance in discriminating the correct key, even with the presence of small level of noise. Besides the efficiency of SHW labeling technique, we can see that additive noise adversely affects the performance of non-profiling DL-based SCA attacks. The more noise added on power traces, DL is more adaptive with the wrong hypothesis key as depicted in Fig. 6c. It leads to more difficulty in discriminating the correct key.

Table 3. Hyperparameters of MLP models using in experiments.

Hyperparameters	MLP_{ref1}	MLP_{ref2}	$MLP_{Proposed}$
Input	700	700	50
Hidden layer	2	2	6
Neuron	20 × 10	20 × 10	50 × 100 × 150 × 100 × 50 × 25
Label	LSB	SHW	SHW
Optimizer	Adam		
Activation	RELU	RELU	ELU
Learning rate	0.001		
Batch size	1000		
Initializing	Xavier Initialization		

5 Conclusion

In this paper, we have proposed a new MLP instance using the SHW labeling technique to overcome the practical issue like imbalanced classes, high dimensional data in non-profiled DL based SCA context. Our experiments were performed for both types of the original power trace and the power trace with the added Gaussian noise. The results have clarified that our data preparation technique is capable of extracting the useful features for the non-profiled SCA based on neural networks. Furthermore, our proposed model architecture provides reliable results for non-profiled attacks on variety datasets like ASCAD, RISC-V MCU. Significantly, the proposed MLP model using SHW labeling provides better performance than that of Binary labeling with the presence of additive noise. Additionally, the results have shown that the Gaussian noise added on power traces becomes a serious problem compared to the benefits of noise added in the profiling attack. In future work, we will investigate several other activation functions and pre-processing methods to increase neural networks' performance for non-profiled attacks.

Acknowledgment. This research is funded by Vietnam National Foundation for Science and Technology Development (NAFOSTED) under grant number 102.02-2020.14.

References

1. Alipour, A., Papadimitriou, A., Beroulle, V., Aerabi, E., Hély, D.: On the performance of non-profiled differential deep learning attacks against an AES encryption algorithm protected using a correlated noise generation based hiding countermeasure. In: 2020 Design, Automation Test in Europe Conference Exhibition (DATE), pp. 614–617 (2020). https://doi.org/10.23919/DATE48585.2020.9116387
2. Hettwer, B., Gehrer, S., Güneysu, T.: Applications of machine learning techniques in side-channel attacks: a survey. J. Cryptogr. Eng. **10**(2), 135–162 (2019). https://doi.org/10.1007/s13389-019-00212-8

3. Cagli, E., Dumas, C., Prouff, E.: Convolutional neural networks with data augmentation against jitter-based countermeasures. In: Fischer, W., Homma, N. (eds.) CHES 2017. LNCS, vol. 10529, pp. 45–68. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-66787-4_3
4. Do, N.T., Hoang, V.P., Doan, V.S.: Performance analysis of non-profiled side channel attacks based on convolutional neural networks. In: 2020 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS), pp. 66–69 (2020). https://doi.org/10.1109/APCCAS50809.2020.9301673
5. Do, N.-T., Hoang, V.-P.: An efficient side channel attack technique with improved correlation power analysis. In: Vo, N.-S., Hoang, V.-P. (eds.) INISCOM 2020. LNICST, vol. 334, pp. 291–300. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-63083-6_22
6. Gilmore, R., Hanley, N., O'Neill, M.: Neural network based attack on a masked implementation of AES. In: 2015 IEEE International Symposium on Hardware Oriented Security and Trust (HOST), pp. 106–111 (2015). https://doi.org/10.1109/HST.2015.7140247
7. Hu, W., Wu, L., Wang, A., Xie, X., Zhu, Z., Luo, S.: Adaptive chosen-plaintext correlation power analysis. In: 2014 Tenth International Conference on Computational Intelligence and Security, pp. 494–498 (2014). https://doi.org/10.1109/CIS.2014.94
8. Kim, J., Picek, S., Heuser, A., Bhasin, S., Hanjalic, A.: Make some noise. unleashing the power of convolutional neural networks for profiled side-channel analysis. IACR Transactions on Cryptographic Hardware and Embedded Systems, pp. 148–179, May 2019. https://doi.org/10.46586/tches.v2019.i3.148-179
9. Kingma, D., Ba, J.: Adam: a method for stochastic optimization. In: International Conference on Learning Representations, December 2014
10. Lerman, L., Bontempi, G., Markowitch, O.: A machine learning approach against a masked AES. J. Cryptogr. Eng. **5**, 123–139 (2014)
11. Maghrebi, H.: Deep learning based side channel attacks in practice. IACR Cryptol. ePrint Arch. **2019**, 578 (2019)
12. Picek, S., Heuser, A., Jovic, A., Bhasin, S., Regazzoni, F.: The curse of class imbalance and conflicting metrics with machine learning for side-channel evaluations (2018). https://ia.cr/2018/476
13. Picek, S., Samiotis, I.P., Kim, J., Heuser, A., Bhasin, S., Legay, A.: On the performance of convolutional neural networks for side-channel analysis. In: Chattopadhyay, A., Rebeiro, C., Yarom, Y. (eds.) SPACE 2018. LNCS, vol. 11348, pp. 157–176. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-05072-6_10
14. Prouff, E., Strullu, R., Benadjila, R., Cagli, E., Canovas, C.: Study of deep learning techniques for side-channel analysis and introduction to ASCAD database. IACR Cryptol. ePrint Arch. **10**, 53 (2018)
15. Timon, B.: Non-profiled deep learning-based side-channel attacks. IACR Cryptol. ePrint Arch. **2018**, 196 (2018)
16. 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 **9**, 22480–22492 (2021). https://doi.org/10.1109/ACCESS.2021.3055833