Design of Convolutional Neural Networks Architecture for Non-Profiled Side-Channel Attack Detection

Amjed Abbas Ahmed[1, 2], Mohammad Kamrul Hasan[1, *], Shayla Islam[3], Azana Hafizah Mohd Aman[1], Nurhizam Safie[1]
[1]Center for Cyber Security, Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia (UKM), Bangi 43600, Malaysia
[2]Department of Computer Techniques Engineering, Imam Al-Kadhum College (IKC), Baghdad, Iraq
[3]Institute of Computer Science and Digital Innovation, UCSI University, 56000 Kuala Lumpur, Malaysia
amjedabbas@alkadhum-col.edu.iq, [*]mkhasan@ukm.edu.my, shayla@ucsiuniversity.edu.my, azana@ukm.edu.my, nurhizam@ukm.edu.my

***Abstract*—Deep learning (DL) is a new option that has just been made available for side-channel analysis. DL approaches for profiled side-channel attacks (SCA) have dominated research till now. In this attack, the attacker has complete control over the profiling device and can collect many traces for a range of critical parameters to characterise device leakage before the attack. In this study, we apply DL algorithms to non-profiled data. An attacker can only retrieve a limited number of side-channel traces from a closed device with an unknown key value in non-profiled mode. The authors conducted this research. Key estimations and deep learning measurements can reveal the secret key. We prove that this is doable. This technology is excellent for non-profits. DL and neural networks can benefit these organisations. Neural networks can provide a new technique to verify the safety of hardware cryptographic algorithms. It was recently suggested. This study creates a non-profiled SCA utilising convolutional neural networks (CNNs) on an AVR microcontroller with 8 bits of memory and the AES-128 cryptographic algorithm. We used aligned power traces with several samples to demonstrate how challenging CNN-based SCA is in practise. This will help us reach our goals. Here is another technique to create a solid CNN data set. In particular, CNN-based SCA experiment data and noise effects are examined. These experiments employ power traces with Gaussian noise. The CNN-based SCA works well with our data set for non-profiled attacks. Gaussian noise on power traces causes many more issues. These results show that our method can recover more bytes successfully from SCA compared to other methods in correlation power analysis (CPA) and DL-SCA without regularisation.**

***Index Terms*—Non-profile side-channel attack; AES; CNN.**

I. INTRODUCTION

Researchers and security experts worldwide have recently paid close attention to side-channel attacks (SCA) [1]. To devise countermeasures, they first apply various cipher-cracking strategies and then provide recommendations to make the ciphers more secure [2]. Some researchers use deep learning models [3] to carry out SCA [4]. Convolutional neural networks (CNNs) [5] were the primary tool used to demonstrate the efficacy of their attacks and describe how they were carried out. During the evolution of SCA throughout history, there have been three distinct stages.

The beginning of the SCA (from 1996 to 2000), the main feature of this stage is the identification and use of different forms of side-channel information for vital analysis. In 1996 [6], it was found that the execution time of the algorithm could be attacked to break Rivest-Shamir-Adleman (RSA). In 1998 [7], the power consumption leakage model was applied to the problem of breaking data encryption standard (DES). One of the vulnerabilities of DES is its susceptibility to SCA, which exploit information leaked through unintended channels such as timing, power consumption, or electromagnetic radiation. According to the findings of research conducted by Quisquater [8] in 2000, electromagnetic radiation can also be used effectively for SCA.

The first phases of forming the SCA (from 2001 to 2010). In this stage, the primary distinguishing characteristic is the increasing emphasis placed on SCA assessment, countermeasures, and applications, in addition to the discovery of novel leakage models. 2008 saw the beginning of the side-channel analysis competition known as the differential power analysis (DPA) contest [9]. The traces that collected from this DPA competition were used as a basis for several subsequent studies that were based on machine learning [10]. In 2010, SCA that used flash memory pumping, SCA that relied on watermarks, and SCA that exploited fault sensitivity were prevalent.

The pinnacle of advancement for SCA (after 2011) [11].

Manuscript received 27 January, 2023; accepted 15 April, 2023.
This work has been supported by the Universiti Kebangsaan Malaysia (UKM) under Grant No. TAP-K023208.

The greater use of cross-domain technology for SCA is the primary feature of this stage. In particular, deep learning methods such as multi-layer perceptron (MLP) [12] and CNN are becoming more popular. CNNs have been shown to defeat jitter-based countermeasures, power trace misalignment, and disguised Advanced Encryption Standard (AES) implementations. As a result, this research uses CNNs.

II. LITERATURE REVIEW

Maghrebi, Rioul, Guilley, and Danger [3] were the main investigators in exploiting CNNs for side-channel attacks (SCA), although they were not the learning methods that deployed deep learning approaches such as MLP, CNN, and long short-term memory (LSTM) [4]. These techniques include random forest and support vector machine (SVM) [5]. The findings of their study show that deep learning is superior to more conventional approaches to machine learning and, as a result, produces good outcomes. The authors show this using two different data sets, one of which is an implementation that does not have any kind of protection, and the other utilises a countermeasure for masking. In addition, the results show that the CNN database, sometimes referred to as the side-channel analysis data set, is in [6]. This database has been used in the investigations of various researchers and was first presented by the authors. After introducing the data set, they investigate the effect of hyperparameters to find the CNN and MLP architectures [6] that will be the most effective. According to the findings of their study, Masure, Canovas, and Prouff [7] reveal the increase in the volume of the CNN kernel, resulting in better behaviour if the network is confronted with misaligned traces. However, they do not explain why increasing the kernel makes the attack more effective, which is strange. This discovery, in our view, is fascinating and certainly deserves more discussion.

Since both studies reveal that CNN performs successfully in various scenarios, further study was conducted on CNN's behaviour. Picek, Samiotis, Kim, Heuser, Bhasin, and Legay [9] compared CNNs' performance against machine learning methods such as Random Forest, XGBoost, and Naive Bayes. Their main objective is to investigate the circumstances under which CNNs perform better than the other techniques described. According to the findings of their study, CNNs can only improve performance in the aggregate. According to the authors, CNNs are most effective when the traces are not pre-processed, when noise levels are lowered, and when information dimensions are higher (i.e., their many features with many traces). On the contrary, machine learning (ML) schemes could achieve performance that is almost on par with that of CNNs. The discovery that ML methods need noticeably fewer processing resources than CNNs is a significant result. As a result, the researchers have severe reservations about the usefulness of CNNs.

After further research, CNNs were shown to have the potential to surpass existing specific data sets with state-of-the-art solutions. An implementation with a covert countermeasure was the source of the measurements for each data set. The authors in [10] performed tests to show that CNNs can synchronise non-aligned traces by identifying the properties of the most significant trace, enabling grouping to be carried out by applying the chosen characteristics. The findings of such experiments are presented in the article. In addition to these discoveries, the scientists explain that this attack is carried out by using raw trace data without any pre-processing.

In contrast to a template attack, which generally includes the adversary realigning the traces and selecting the points of interest on their own, this one does not. Because of this, the findings show that CNNs are beneficial even when the traces are misaligned. On the other hand, overfitting is potential due to the size and complexity of the CNN architecture that lies under the surface. They provide two data augmentation algorithms for misaligned traces as a means of generating more training data to do this. Experiments were carried out to illustrate the efficacy of data augmentation options for misaligned traces.

The findings corroborated by Kim, Picek, Heuser, Bhasin, and Hanjalic [11] show that their CNN framework performs at the leading edge in the random delays (RD) data set. This gives more credence to the findings in [12]. In particular, compared to DPAv4a data set considered a fundamental information set, an ideal network of its needs fewer attack traces to recover the key of the RD data set [13]. In [14], the researchers experimented with a wide variety of topologies and sets of pieces of information. The results of such experiments showed that no single design succeeds with all data sets. Hence, this remains very necessary in selecting a structure appropriate for issues present at that time. In addition, the authors provide evidence that including distortion within the primary substrates of the networks helps performance by reducing the amount of overfitting that occurs. When working with smaller data sets, this is suggested by using increased noise levels, whereas working with more extensive data sets requires a lower noise level to get the best results.

These studies suggest that CNNs include two essential qualities that make them suitable for side-channel analysis. To begin with, they can determine the most critical features independently and without any guidance. As a consequence of this, prior processing on the traces was not needed to gain greater behaviour. Compared to more conventional approaches, we consider this to be a considerable advantage. According to the authors in [15], pre-processing is prone to errors, and poor selection of Points of Interest (PoI) leads to lower performance. Because CNNs are spatially invariant, they can identify characteristics regardless of their position within feature vectors. This is the second benefit of using CNNs. As a result of this quality, CNNs can perform at the cutting edge of the field when it comes to data sets originating from implementations that use a concealed countermeasure. The methodologies used in the study that we have discussed up to this point are standard practises in deep learning. According to further research, it has been recommended that new innovative tactics be used, designed explicitly for the side-channel attack, aiming to take advantage of a few qualities.

The researchers in [16] suggested a completely new CNN framework that uses more domain information obtained through a side-channel attack. Data provided for creating neural networks can be plaintext or ciphertext, and this

distinction is determined by the leaky model. The classification block of a CNN architecture is the component that is given the domain information to use as a new feature vector. In the work, the authors compare several architectural concepts offered by various works of literature with and without the architecture that they have provided. They show how a design that uses domain knowledge can improve performance for protected information and not protected information. However, if profile traces are generated using a fixed key, this method is not proper.

Zaid, Bossuet, Dassance, Habrard, and Venelli [17] strongly focus on the need for fine-tuning architecture and hyper-parameters; models do not operate correctly without an appropriate configuration. They point out that we cannot realise the full potential of architecture if we do not understand the influence of a hyperparameter and explain why this is the case. The authors provide three visualisation methods to solve this problem: weight, gradient, and heatmap. These methods are utilised to improve the readability and interpretability of each hyperparameter. These approaches make it simpler to set the hyperparameters by allowing an opponent to determine the influence of each one individually, which in turn makes it easier to tune the hyperparameters. Using these three visualisation approaches, they also propose implementation options for protected and unprotected environments. In particular, for data sets that include a concealed countermeasure, it is recommended that the CNN kernel measure be modified to equal 50 % of the highest delay of randomised delay. It remains one of the guidelines provided by their method.

In contrast to the content provided in articles produced by deep learning communities, the increase in substrate is recommended as opposed to the number of neurones contained within each layer [18]. The authors improved the state-of-the-art work on entire information sets by developing architectures and conducting tests with all publicly available data sets using the methodologies described, which led to an increase in overall performance. On the other hand, the choice of hyperparameters is occasionally made without enough rationale, even though their method offers cutting-edge performance for all publicly available data sets. For example, the authors do not explain how certain learning rates were determined for a few specific data sets or why they were used in the first place. They also do not explain why they were used at all.

Pfeifer and Haddad [19] propose using a deep learning layer known as the spread layer. This layer would be the first to be explicitly designed for side-channel attacks. As demonstrated in their study, Haddad and Pfeifer depicted with this layer that some substrate was needed for better outcomes. Furthermore, the profiling phase needs fewer traces, which speeds up the learning procedure. Such findings were intriguing about side-channel analysis communities because they suggest motivation to create substrates specially made to take advantage of the side-channel properties of traces. This is because these findings indicate a motivation to create layers specially made to take advantage of the side-channel properties of traces. On the other hand, the authors do not provide much information about how to establish the hyperparameters of the layer or why this layer can offer the results it does. These questions will be addressed in Section IV, at which point we will investigate the spread layer in great detail and solve some of its faults that it has.

According to Jin, Kim, Kim, and Hong [20], the deep CNN framework works admirably for SCAs. Despite this, some issues remain concerning the training process for deep neural networks. The primary issue is that training deep neural networks can be complicated since gradients can either vanish or grow as the training progresses. In the sections concerned, we will explain the latest advancements with the initiation of deep neural networks to solve the issues above.

Much work has been done on parameter initialisation topics; variables would often be picked randomly from a Gaussian distribution. This was significantly reworked by Glorot and Bengio [21], who also introduced the latest initialisation technique called “Xavier’s initialisation” simultaneously. This method considers the number of inputs and outputs associated with the parameter while simultaneously deriving the parameter values from a Gaussian distribution. This method is currently considered standard practise and is used to initialise the parameters of several extensive deep-learning libraries. When academics began looking into the architectures of deep neural networks, they found that several works ran into problems with the convergence of their designs. Convergence problems were experienced, e.g., by the well-known visual geometry group (VGG) architecture, which is trained in four phases. The network is then enlarged with additional layers, and training is performed at each stage to ensure that it converges correctly [22].

A novel strategy for deep CNN initialisation is presented by F.-X. Standaert in [23]. According to his research findings, even though the Xavier initialisation was designed to work with linear activations, it is not appropriate for use with the rectified linear unit (ReLU). In addition, they argue that deeper networks have a more difficult time reaching a point of convergence. A solution to such issues, provided by them, is the initialisation of “He”, which was developed specifically for CNNs that use ReLU and, compared to other initialisation methods, results in an improvement in the degree to which deep neural networks converge. layer sequential unit variance (LSUV) initialisation is an alternative method of initialisation that is proposed in [24]. Rather than being developed explicitly for designs that use ReLU as an activation function, this approach exhibits an additional generic character and is appropriate for various architectural kinds. They provide evidence of the viability of their approach by conducting experiments to validate their claims. Both sets of research have shown how important it is to have accurate initialisation of the network parameters for deep neural networks to converge. The published research has only recently begun to do investigations in actual SCA circumstances where the attack traces and the profiling traces were obtained from identical gadgets. This was not unusual for people to use the same key for both the attack path and the profiling track. The results of these studies can, as a direct consequence of this fact, provide an inaccurate image of the effectiveness of several therapies, including template attack (TA), ML, and DL. Consequently, the SCA

community has begun to construct a more realistic environment in which various gadgets are used to acquire attack and profiling traces [25]. A comparison of various research methods on SCA is shown in Table I.

TABLE I. COMPARISON OF VARIOUS RESEARCH METHODS ON SCA.

Research works	Attacked Network	Physical Measurement	Limitations
Wu *et al.* [13], 2023	MLP, CNN	EM	Minimal (black box)
Maji, Banerjee, Fuller, and Chandrakasan [14], 2022	CNN, BNN	SPA	Methodology specific to µC
Shimada, Kuroda, Fukuda, Yoshida, and Fujino [15], 2022	MLP	EM	Intention paper
Sako, Kuroda, Fukuda, Yoshida, and Fujino [22], 2022	Systolic array	CPA	Only the systolic array is implemented
Shi, Sun, Wang, and Hu [24], 2020	BNN	Power	Specific to the line buffer
Yang, Xiang, Huang, Fu, and Yang [25], 2023	CNN	Power	Using non-fine-tuned models once trained

III. PROPOSED METHODOLOGY

One of the most popular uses of CNN is image recognition [16]. They are effective in dividing time series [17]. CNNs are great models for the extraction of features and categorisation of complex data because they are invariant to translation. As a result, our attack into side-channel data attacks benefits from the use of CNNs. CNN has the drawback of being trained for each major theory separately. Our best guesses for the 8-bit key will require 256 trials of practise.

CNNs use layers of computation known as convolutional and pooling layers. The batch normalisation layers will complete these processes today. The batch normalisation of Ioffe and Szegedy [18] reduces the internal covariate shift of neural networks. The authors claim that this leads to more efficient learning. When we put CNN through its paces, we use a series of aligned power wires as our test subject. One power trace sample would include too much data to be used as CNN input features. Therefore, we use the correlation coefficient in the first phase of power-trace processing. There are typically three parts to a CNN data set: the training set, which is used to teach the network, the validation set, which is used to test the accuracy of the network on unseen data, and the test set, which is used to assess the quality of the final prediction or classification. Details of the network architecture will be covered in a subsequent section.

– *Experiment and Equipment Details*

To evaluate our neural network models, we employed MATLAB. Three convolutional layers and three pooling layers precede the fully connected layer and the classification layer in the network, respectively. The first convolution layer has 16 filters, each of which is [11] by [12] in size, and has an output layer of the same size as the input. The subsequent two convolutional layers are the same size as the first but have 24 and 32 filters, respectively. By sliding filters down the edge of the layer below them, convolutional layers can perform convolution on incoming data. The fact that CNN minimises the loss function using filter weights allows it to learn invariant features during translation. Power cords will not hinder the mobility of SCA filters. The max-pooling method with kernel size [12] and stride [12] is used for the first two layers, while the average-pooling technique is used for the third and final layers. Maximum and average pooling are non-linear layers that can bring down data dimensions. When comparing average pooling with maximum pooling, it is important to note that the former determines an average, while the latter determines a maximum. All convolutional layers in our model use ReLU Piecewise linear means that when the input is positive, the output is also positive. Softmax is used to do the categorisation in the output layer. Table II shows the simulation parameters of the proposed CNN model and Fig. 1 details the structural makeup of our convolutional neural network.

TABLE II. SIMULATION PARAMETERS OF THE PROPOSED CNN MODEL.

Layer	Weight Shape	Stride	Activation
Convolutional (1)	1 × 3 × 16	-	-
Batch Normalisation (1)	-	-	ReLU
Max-Pooling (1)	-	[12]	-
Convolutional (2)	1 × 3 × 24	-	-
Batch Normalisation (2)	-	-	ReLU
Max-Pooling (2)	-	[12]	-
Convolutional (3)	1 × 3 × 24	-	-
Batch Normalisation (3)	-	-	ReLU
Average-pooling (1)	-	[12]	-
FC-output	-	-	Softmax

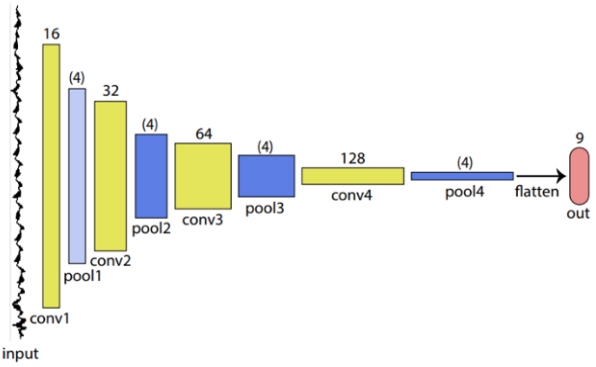


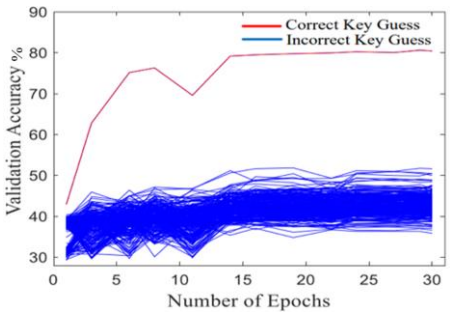
Fig. 1. Proposed model.

IV. RESULTS

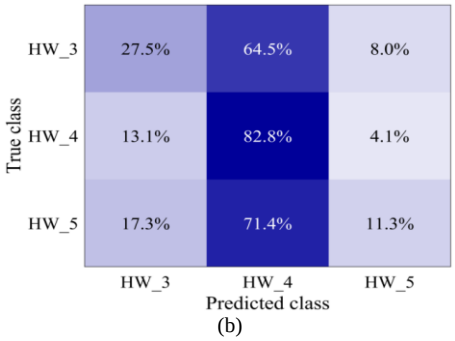
The CW1173 ChipWhisperer board was subjected to our testing [19]. This SCA platform has a target board equipped with an 8-bit Atmel AVR Xmega128 microcontroller capable of executing AES-128. The ChipWhisperer's internal analog to digital converter (ADC) can capture the continuous wave (CW) Lite signal. Because this system is set up, we can deliver the software, the plaintext, and the key to the Xmega board while recording the traces on a laptop. For purposes of conducting tests, we have 5000 power traces available. 10,000 AES Round 1 and Round 2 samples are included in each power trace. An attack that is not profiled will keep the same key throughout and will choose 5000 plaintexts at random. The only round of the AES we focus on attacking is the first.

The next step is to train a CNN by putting 80 % of the data set through its paces during training and just 20 % through its paces during testing. Selecting the features that have the highest correlation values for I and k is an effective way to assist the CNN model in locating HW labels. The correct value for the key parameter, k, will accurately predict the sequence of intermediate HW values, ultimately leading to the appropriate CNN training labels. If the CNN can acquire the appropriate properties from the correct key, it should train successfully and increase training metrics, such as loss and accuracy, with time. The intermediate values of all other key candidates will be inaccurate and lead to unsuccessful training. The attacker can discover the proper key value if they choose the key with the highest training metrics. The results of the experiment will now be presented. Figure 2 shows clean power traces validation accuracy and confusion matrix. Figure 2(a) and Fig. 2(b) show validation accuracies and incorrect key guesses, respectively.

The accuracy of the training network's validation can judge the success of CNN's non-profiled attacks. Loss and accuracy over epochs are the two primary metrics used to measure CNN training, as previously mentioned. This study focused on precision in finding the appropriate sub-byte key. Validation accuracy is shown in Fig. 2(a), which was derived from an attack on our data set using 30 epochs for each estimate. The number of training epochs is shown along the horizontal axis, and the validation accuracy of the training network is shown along the vertical axis. Because we used the correct sub-byte key value, the validation accuracy of our training was much greater than that of the other companies. Even after ten epochs, the attack manages to discover the hidden key. It is not difficult to determine the suitable key guess if we use the highest accuracy value for each sub-byte key guess. More intriguingly, we can utilise the confusion matrix to differentiate between the distributions of the three HW labels. The incorrect candidates will fall under the HW4 label, while the correct candidate will be marked independently, as shown in Fig. 2(b). After discovering these data, we set out to determine how the power trace noise affected the accuracy of the recommended CNN. The initial power traces consist of three different layers of Gaussian noise. Then came three sets of data. The results of the training are shown in Fig. 3. Even though there is very little noise, as shown in Fig. 3, our CNN can still identify the correct key after ten epochs have passed. The correct key is concealed in the last key byte generated after each epoch with more than 3000 power traces. The greater the noise variance, the less accurate the validation becomes.



(a)



(b)

Fig. 2. Clean power traces validation accuracy and confusion matrix: (a) Validation accuracies; (b) Incorrect key guesses.

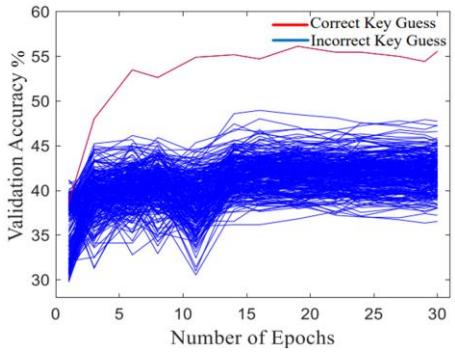


Fig. 3. Validation accuracy when Gaussian noise is added (0.025).

– Comparative Analysis

The potential number of partial keys that can be derived from a sample of 30,000 traces is shown graphically in Fig. 4.

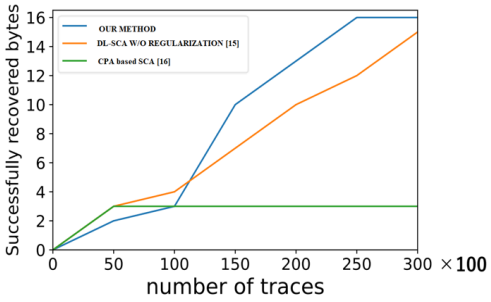


Fig. 4. Comparison of the results of the SCA attack.

The CPA was unsuccessful, except for the unmasked bytes 2 and 5. Regularisation caused our method DL-SCAs (CNN) to attack all 16 bytes in 25,000 traces, whereas they only attacked 15 bytes in 30,000 without regularisation (DL-SCA). Compared to first-order CPAs, non-profiled DL-SCAs performed better in attacks. The disclosed masking SCA countermeasure is vulnerable to attack if a high-order CPA can be utilised to reliably predict the internal mask value. The risk of side-channel attacks is significantly increased since an attacker unfamiliar with the underlying processing of the countermeasure can still get all partial keys using non-profiled DL-SCAs. These results show that our method can recover more bytes successfully from SCA compared to other methods in CPA [14] and DL-SCA without regularisation [15].

V. CONCLUSIONS

According to the findings of this research, CNN creates difficulties for SCA when aligned power traces include a large number of samples. After preparing the CNN training

data, we evaluated the power traces with the original data and those with Gaussian noise. Our non-profiled SCA data preparation method is based on CNN, which allows for extracting key properties. Our method requires fewer power traces for attacks because the power traces are organised into three distinct groups. These findings indicate that our technique can effectively recover an increased number of bytes from SCA compared to previous methods used in CPA and DL-SCA without regularisation. The consistent findings that our CNN architecture produces for attacks that are not profiled highlight the considerable challenge posed by Gaussian noise in power traces. To improve the performance of neural networks when faced with non-profiled attacks, we will investigate several pre-processing strategies that aim to decrease power trace noise.

CONFLICTS OF INTEREST

The authors declare that they have no conflicts of interest.

REFERENCES

[1] G. Yang, H. Li, J. Ming, and Y. Zhou, "Convolutional neural network based side-channel attacks in time-frequency representations", in *Smart Card Research and Advanced Applications. CARDIS 2018. Lecture Notes in Computer Science()*, vol. 11389. Springer, Cham, 2019, pp. 1–17. DOI: 10.1007/978-3-030-15462-2_1.

[2] B. Timon, "Non-profiled deep learning-based side-channel attacks with sensitivity analysis", *IACR Transactions on Cryptographic Hardware Embedded Systems*, vol. 2019, no. 2, pp. 107–131, 2019. DOI: 10.13154/tches.v2019.i2.107-131.

[3] H. Maghrebi, O. Rioul, S. Guilley, and J.-L. Danger, "Comparison between side-channel analysis distinguishers", in *Information and Communications Security. ICICS 2012. Lecture Notes in Computer Science*, vol. 7618. Springer, Berlin, Heidelberg, 2012, pp. 331–340. DOI: 10.1007/978-3-642-34129-8_30.

[4] H. Maghrebi, "Deep learning based side channel attacks in practice", *Cryptology ePrint Archive*, vol. 2019, p. 578, 2019.

[5] B. Timon, "Non-profiled deep learning-based side-channel attacks", *IACR Cryptology ePrint Archive*, vol. 2018, p. 196, 2019. DOI: 10.46586/tches.v2019.i2.107-131.

[6] D. Das, A. Golder, J. Danial, S. Ghosh, A. Raychowdhury, and S. Sen, "X-DeepSCA: Cross-device deep learning side channel attack", in *Proc. of 2019 56th ACM/IEEE Design Automation Conference (DAC)*, 2019, pp. 1–6. DOI: 10.1145/3316781.3317934.

[7] L. Masure, C. Canovas, and E. Prouff, "A comprehensive study of deep learning for side-channel analysis", *IACR Transactions on Cryptographic Hardware and Embedded Systems*, vol. 2020, pp. 348–375, 2019. DOI: 10.13154/tches.v2020.i1.348-375.

[8] J. J. Quisquater, "A new tool for non-intrusive analysis of smart cards based on electro-magnetic emissions. The SEMA and DEMA methods", *Eurocrypt Rump Session*, 2000.

[9] S. Picek, I. P. Samiotis, J. Kim, A. Heuser, S. Bhasin, and A. Legay, "On the performance of convolutional neural networks for side-channel analysis", in *Security, Privacy, and Applied Cryptography Engineering. SPACE 2018. Lecture Notes in Computer Science()*, vol. 11348. Springer, Cham, 2018, pp. 157–176. DOI: 10.1007/978-3-030-05072-6_10.

[10] H. Wang, S. Forsmark, M. Brisfors, and E. Dubrova, "Multi-source training deep-learning side-channel attacks", in *Proc. of 2020 IEEE 50th International Symposium on Multiple-Valued Logic (ISMVL)*, 2020, pp. 58–63. DOI: 10.1109/ISMVL49045.2020.00-29.

[11] 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 Transactions on Cryptographic Hardware Embedded Systems*, vol. 2019, no. 3, pp. 148–179, 2019. DOI: 10.13154/tches.v2019.i3.148-179.

[12] Y.-S. Won, D.-G. Han, D. Jap, S. Bhasin, and J.-Y. Park, "Non-profiled side-channel attack based on deep learning using picture trace", *IEEE Access*, vol. 9, pp. 22480–22492, 2021. DOI: 10.1109/ACCESS.2021.3055833.

[13] L. Wu *et al.*, "Label correlation in deep learning-based side-channel analysis", *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 3849–3861, 2023. DOI: 10.1109/TIFS.2023.3287728.

[14] S. Maji, U. Banerjee, S. H. Fuller, and A. P. Chandrakasan, "A threshold-implementation-based neural-network accelerator securing model parameters and inputs against power side-channel attacks", in *Proc. of 2022 IEEE International Solid-State Circuits Conference (ISSCC)*, 2022, pp. 518–520. DOI: 10.1109/ISSCC42614.2022.9731598.

[15] S. Shimada, K. Kuroda, Y. Fukuda, K. Yoshida, and T. Fujino, "Deep learning-based side-channel attacks against software-implemented RSA using binary exponentiation with dummy multiplication", *IEICE Technical Report*, vol. 122, no. 11, pp. 13–18, 2022.

[16] B. Sönmez, A. A. Sarıkaya, and Ş. Bahtiyar, "Machine learning based side channel selection for time-driven cache attacks on AES", in *Proc. of 2019 4th International Conference on Computer Science and Engineering (UBMK)*, 2019, pp. 1–5. DOI: 10.1109/UBMK.2019.8907211.

[17] G. Zaid, L. Bossuet, F. Dassance, A. Habrard, and A. Venelli, "Ranking loss: Maximizing the success rate in deep learning side-channel analysis", *IACR Transactions on Cryptographic Hardware Embedded Systems*, vol. 2021, no. 1, pp. 25–55, 2021. DOI: 10.46586/tches.v2021.i1.25-55.

[18] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift", in *Proc. of the 32nd International Conference on Machine Learning*, 2015, pp. 1–9.

[19] C. Pfeifer and P. Haddad, "Spread: A new layer for profiled deep-learning side-channel attacks", *IACR Cryptology ePrint Archive*, vol. 2018, p. 880, 2018.

[20] S. Jin, S. Kim, H. Kim, and S. Hong, "Recent advances in deep learning-based side-channel analysis", *ETRI Journal*, vol. 42, no. 2, pp. 292–304, 2020. DOI: 10.4218/etrij.2019-0163.

[21] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks", in *Proc. of the Thirteenth International Conference on Artificial Intelligence and Statistics*, 2010, pp. 249–256.

[22] M. Sako, K. Kuroda, Y. Fukuda, K. Yoshida, and T. Fujino, "Deep learning side-channel attacks against hardware-implemented lightweight cipher Midori 64", *IEICE Technical Report*, vol. 122, no. 11, pp. 7–12, 2022.

[23] F.-X. Standaert, "Introduction to side-channel attacks", in *Secure Integrated Circuits and Systems. Integrated Circuits and Systems*. Springer, Boston, MA, 2010, pp. 27–42. DOI: 10.1007/978-0-387-71829-3_2.

[24] M. Wei, D. Shi, S. Sun, P. Wang, and L. Hu, "Convolutional neural network based side-channel attacks with customized filters", in *Information and Communications Security. ICICS 2019. Lecture Notes in Computer Science()*, vol 11999. Springer, Cham, 2020, pp. 799–813. DOI: 10.1007/978-3-030-41579-2_46.

[25] W. Yang, X. Xiang, C. Huang, A. Fu, and Y. Yang, "MCA-based multi-channel fusion attacks against cryptographic implementations", *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 13, no. 2, pp. 476–488, 2023. DOI: 10.1109/JETCAS.2023.3252085.