engineering proceedings

MDPI

Proceeding Paper

Application of Neural Networks to Power Analysis †

Alla Levina and Roman Bolozovskii *

Laboratory of Fundamentals of Intelligent Systems "LETI", Saint Petersburg Electrotechnical University, Professora Popova Str., 5, 197376 Saint-Petersburg, Russia

* Correspondence: bolozovskii@gmail.com

† Presented at the 15th International Conference "Intelligent Systems" (INTELS'22), Moscow, Russia, 14–16 December 2022.

Abstract: The purpose of this work is to research the possibility of a side-channel attack, more precisely power consumption attack (recovering the encryption key according to the board's power consumption schedule) on the AES-128 algorithm implemented in hardware. Basically, various methods can be used to make an attack, including SPA (Simple Power Consumption Attack) and DPA (Differential Power Consumption Attack). SPA methods involve a simple visual analysis of energy consumption graphs, while DPA involves the use of statistical methods to recover the encryption key. One way to make side-channel attacks more effective is to implement machine learning methods for the described purposes.

Keywords: cryptography; AES; side-channel attacks; power analysis attack; neural networks

Citation: Levina, A.; Bolozovskii, R. Application of Neural Networks to Power Analysis. *Eng. Proc.* **2023**, *33*, 27. https://doi.org/10.3390/engproc2023033027

Academic Editors: Askhat Diveev, Ivan Zelinka, Arutun Avetisyan and Alexander Ilin

Published: 15 June 2023

1. Introduction

The purpose of this research work is to study the possibility of implementing machine learning to side-channel attack, more precisely to speed up the process of analyses of the oscillogram of the power consumption (power consumption) of the encryption board (traces) [1].

Side-channel attacks (SCA) are a powerful type of cryptographic attack that takes advantage of physical leakages, such as electromagnetic radiation or changes in power consumption. Early SCAs involved timing attacks to break Rivest–Shamir–Adleman (RSA), Diffie–Hellman (DH), and Digital Signature Standard (DSS) encryption by measuring the execution time of different code branches and operations. Later, it was discovered that device power consumption leaks information about Data Encryption Standard (DES) and Advanced Encryption Standard (AES) by using both simple [2] and differential analysis [3–6]. Modern SCAs use many non-invasive ways to measure leakage, for example, electromagnetic signals [7,8] or acoustic cryptanalysis [9,10]. More works about different SCA on various cryptographic algorithms can be found in [11–14].

In this work, we will be focusing on the power consumption attack on algorithm AES-128. During the operation, bits of the key used by the encryption board will be restored from the original traces. As initial data, power consumption traces, and encryption keys are needed, they were used during the operation of the board. These data can be obtained using an oscillogram from a programmable logic integrated circuit, on which any encryption algorithm is above.

For the purposes of this work, a TinyAES dataset was selected, it was taken from the board on which the AES-128 algorithm is implemented. Connection data from plaintext, encryption key, ciphertext, and a set of 20,000 points representing a power consumption waveform. In the end, the implementation of research involves obtaining a ready-made algorithm. The extraction of the generated key from the original traces using statistical methods, as well as machine learning methods and neural networks.

The paper is organized as follows. Section 1 describes basic concepts for the work of neural networks and the method of its implementation, Section 2 demonstrates the

results of the implementation of neural networks to power analysis attack on AES-128, and Section 4 summarizes the results of presented researches and plans for future studies.

2. Neural Networks

In this work, a machine learning method will be used as neural networks [15,16]. Neural networks are based on the principle of connectivism—a large number of relatively simple elements are connected in them, and learning comes down to building the optimal structure of connections and setting the connection parameters. To build an artificial neural network (ANN) we will use the same structure. Like a biological neural network, an artificial one consists of neurons interacting with each other, but it is a simplified model. So, for example, an artificial neuron that makes up an ANN has a much simpler structure, it has several inputs on which it receives various signals, transforms them, and transmits them to other neurons. In other words, an artificial neuron is such a function $R^T -> R$, which converts multiple inputs into one output.

As you can see in Figure 1, the neuron has n inputs x_i, each of which has a weight w_i, by which the signal passing through the connection is multiplied. After that, the weighted signals $x_i \cdot w_i$ are sent to the adder, which aggregates all the signals into a weighted sum. This sum is also called net. In this way:

$$net = \sum_{i=1}^{n} x_i \cdot w_i = x \cdot w^T. \quad (1)$$

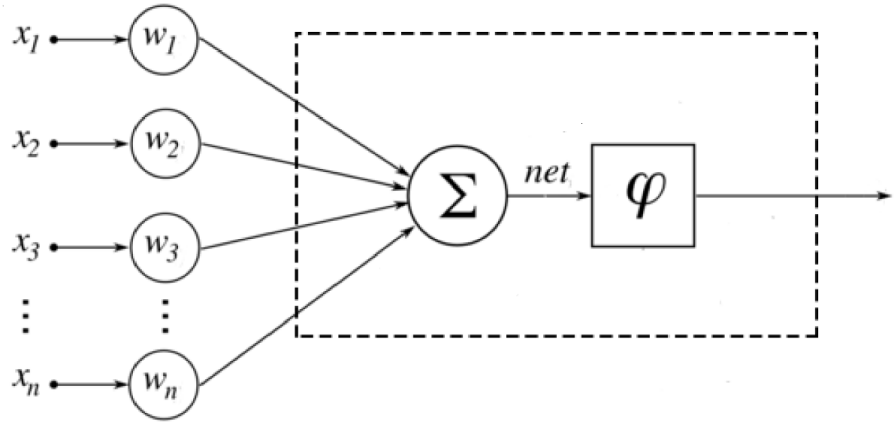

Figure 1. Scheme of an artificial neuron.

Just like that, it is rather pointless to transfer the weighted sum net to the output—the neuron must somehow process it and form an adequate output signal. For these purposes, an activation function is used, which converts the weighted sum into some number, which will be the output of the neuron. The activation function is denoted $\phi(net)$. Thus, the output of an artificial neuron is $\phi(net)$.

Neural network training is the search for such a set of weight coefficients, in which the input signal, after passing through the network, is converted into the output we need.

This definition of "training a neural network" is consistent with biological neural networks. Our brain consists of a huge number of interconnected neural networks, each of which individually consists of neurons of the same type (with the same activation function). Our brains learn by changing synapses, elements that increase or decrease the input signal.

If we train the network using only one input signal, then the network will simply "remember the correct answer", and as soon as we give a slightly modified signal, instead of the correct answer, we will receive nonsense. We expect the network to be able to generalize some features and solve the problem on various input data. It is for this purpose that training samples are created.

A training sample is a finite set of input signals (sometimes along with the correct output signals) on which the network is trained. After the network has been trained, that is, when the network produces correct results for all input signals from the training sample, it can be used in practice. However, before immediately using a neural network, the quality of its work is usually assessed on the so-called test set. A test sample is a finite set of input signals (sometimes together with the correct output signals), which are used to evaluate the quality of the network. Neural network training itself can be divided into two approaches, supervised learning, and unsupervised learning. In the first case, the weights are changed so that the network's answers are minimally different from the ready-made correct answers, and in the second case, the network independently classifies the input signals.

3. Results

In this research work, a model was developed based on the theory of neural networks. An implemented neural network model allows one to obtain information about the secret key that was used for encryption from the captured waveforms of the power consumed by the encryption device.

The initial oscillograms were loaded and normalized to values from −1 to 1, the graph of one of the traces is shown in Figure 2.

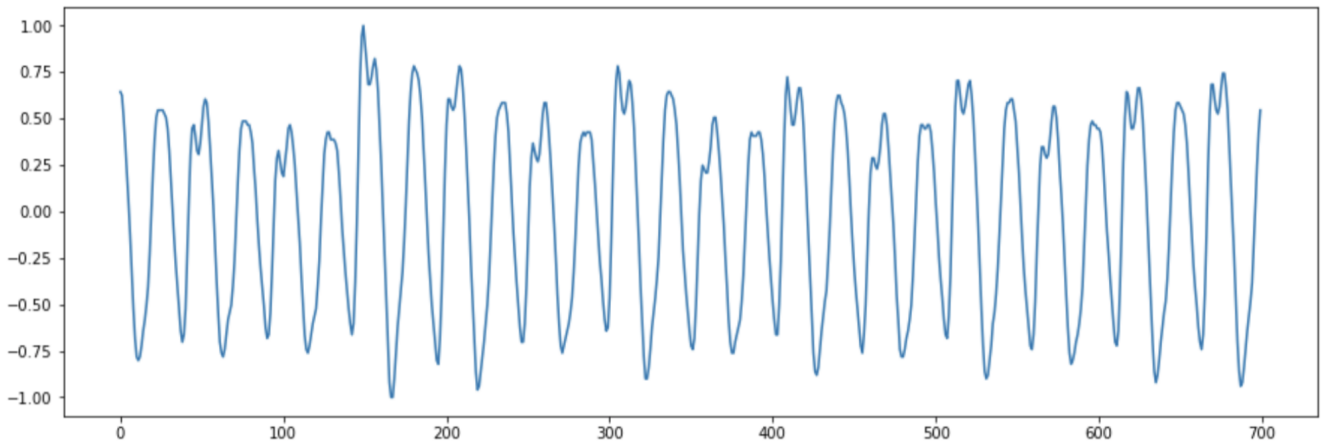

Figure 2. Example of an oscillogram of the power consumption of an encryption module.

Further, for each oscillogram, the bytes of interest were allocated-outputs from the Sbox for the third round. After that, these bytes were transferred to one main vector (vector with dimension 256, where 1 is at the position corresponding to the value of this byte and 0 in all other positions).

The resulting waveforms and one main vector were used to train the SCANet neural network. Graphs of changes in the loss function, and classification accuracy are shown in Figure 3.

To evaluate the final result, the prediction of the neural network, it is necessary to use the rank function, its graph is shown in Figure 4. This function indicates how reliable the values provided by the neural network are. The lower the rank, the easier it is to pick up the bytes that were used in the encryption. As can be seen from the graph, with an increase in the number of traces, the rank decreases, which means a decrease in the complexity of determining the initial key.

When using an untrained neural network to determine the initial key, the rank function does not fall with an increase in the number of traces, that is, the complexity of obtaining the initial key remains at the level of random predictions, which is shown in Figure 5.

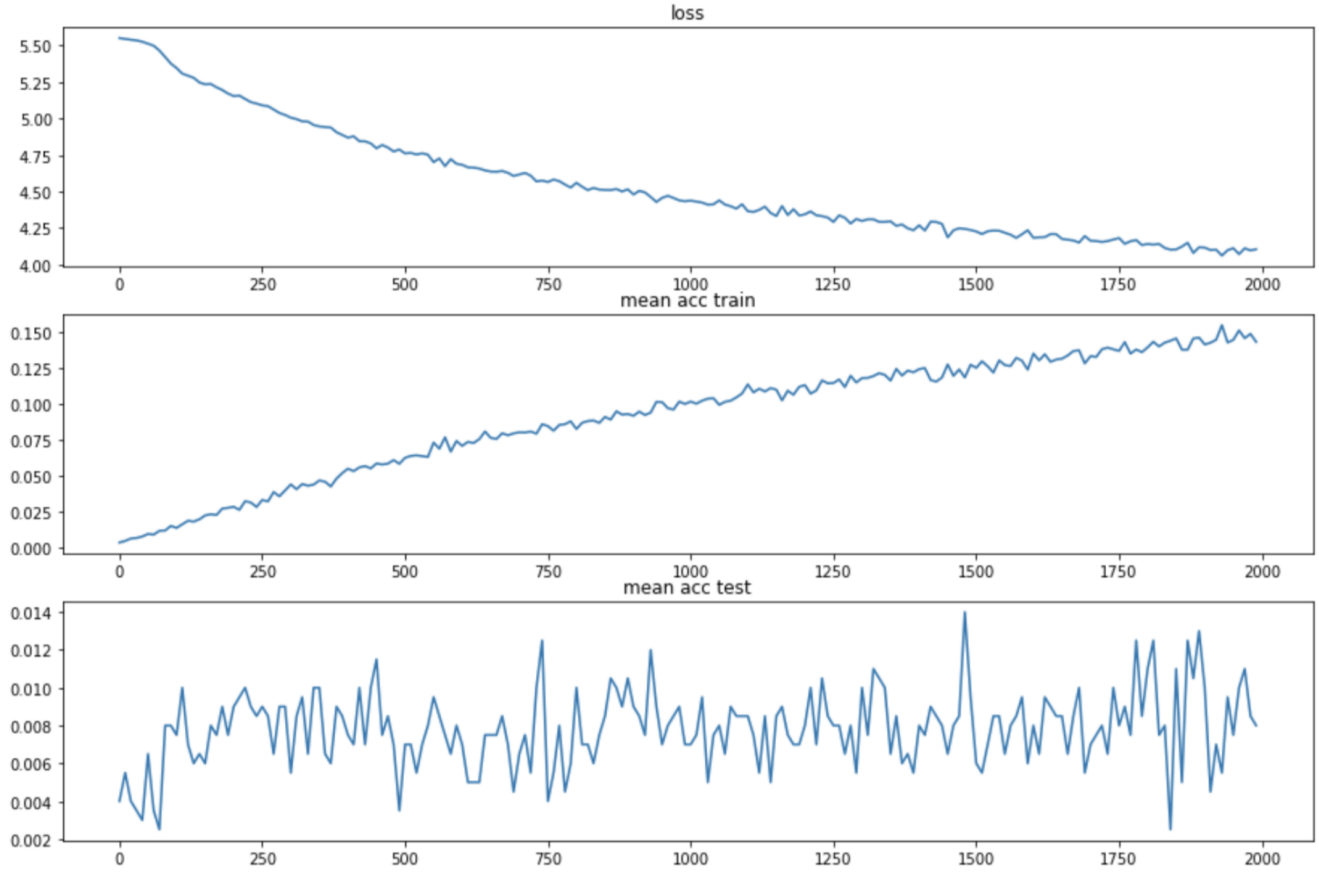

Figure 3. Loss function and classification accuracy plots.

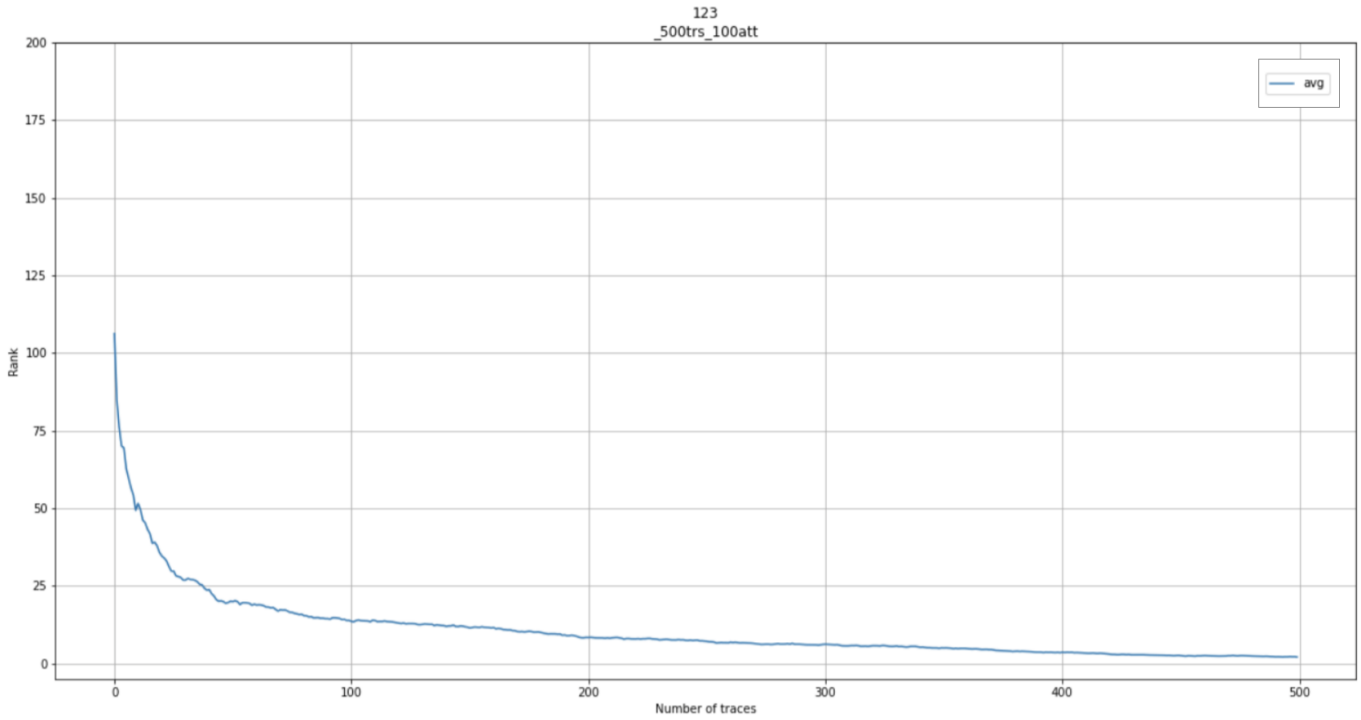

Figure 4. Rank function of the trained network.

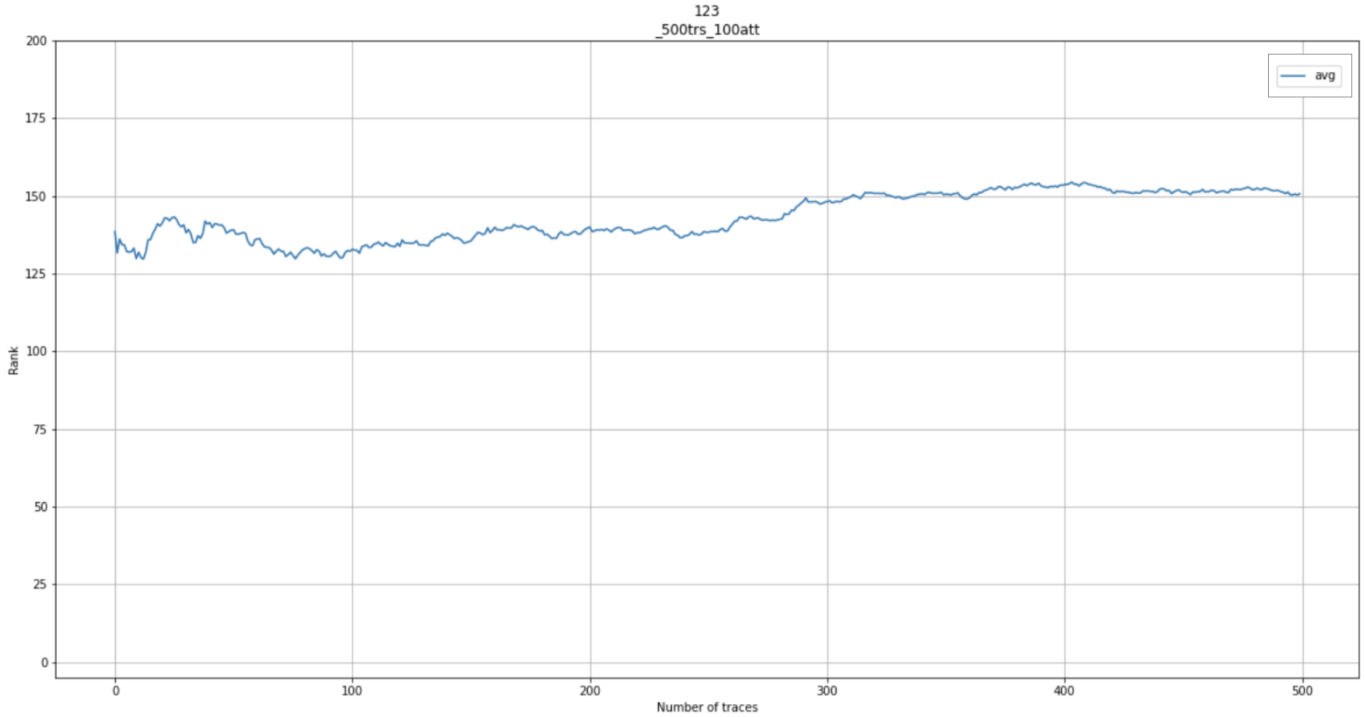

Figure 5. Rank function of an untrained network.

4. Conclusions

In the course of the research work, a neural network model was obtained, which allows, using the oscillogram of the power consumption of the encryption board, to restore the encryption key. The ASCAD dataset was used as the initial data, containing oscillograms, encryption keys, source texts, and points of interest—outputs from the Sbox function of the third round of the AES-128 cipher. The model is a convolutional neural network consisting of four convolutional and two fully connected layers, with Softmax activation function and CrossEntropyLoss as loss function. The quality of the resulting network was verified by the classical method for energy consumption analysis problems, namely, using the rank function.

Author Contributions: Conceptualization, A.L. and R.B.; methodology, R.B.; software, R.B.; validation, A.L. and R.B.; formal analysis, R.B.; investigation, R.B.; resources, R.B.; data curation, R.B.; writing—original draft preparation, A.L.; writing—review and editing, A.L. and R.B; visualization, A.L. and R.B; supervision, A.L.; project administration, A.L.; funding acquisition, A.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Ministry of Science and Higher Education of the Russian Science Foundation (Project "Goszaadanie" No075-01024-21-02 from 29.09.2021, FSEE-2021-0015).

Informed Consent Statement: Not applicable.

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

References

1. Mahanta, H.J.; Azad, A.K.; Khan, A.K. Power analysis attack: A vulnerability to smart card security. In Proceedings of the 2015 International Conference on Signal Processing and Communication Engineering Systems, Guntur, India, 2–3 January 2015; pp. 506–510.
2. Bechtsoudis, A.; Sklavos, N. Side channel attacks cryptanalysis against block ciphers based on FPGA devices. In Proceedings of IEEE Computer Society Annual Symposium on VLSI, Kefalonia, Greece, 5–7 July 2010.
3. Alioto, M.; Giancane, L.; Scotti, G.; Trifiletti, A. Leakage power analysis attacks: A novel class of attacks to nanometer cryptographic circuits. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2009**, *57*, 355–367. [CrossRef]

4. Prouff, E.; Rivain, M.; Bevan, R. Statistical analysis of second order differential power analysis. *IEEE Trans. Comput.* **2009**, *58*, 799–811. [CrossRef]
5. Longo, J.; Mulder, E.D.; Page, D.; Tunstall, M. SoC it to EM: Electromagnetic side-channel attacks on a complex system-on-chip. In Proceedings of the International Workshop on Cryptographic Hardware and Embedded Systems, Saint-Malo, France, 13–16 September 2015; Springer: Berlin/Heidelberg, Germany, 2015; pp. 620–640.
6. Kocher, P.; Jaffe, J.; Jun, B. Differential power analysis. In Proceedings of the Annual International Cryptology Conference (CRYPTO), Santa Barbara, CA, USA, 15–19 August 1999.
7. National Security Agency. *NACSIM 5000 Tempest Fundamentals (Report)*; National Security Agency: Fort Meade, MD, USA, February 1982.
8. Genkin, D.; Pipman, I.; Tromer, E. Get your hands off my laptop: Physical side-channel key-extraction attacks on PCs. *J. Cryptogr. Eng.* **2015**, *5*, 95–112. [CrossRef]
9. Backes, M.; Dürmuth, M.; Gerling, S.; Pinkal, M.; Sporleder, C. Acoustic side-channel attacks on printers. In Proceedings of the 19th USENIX Conference on Security (USENIX Security'10), Washington, DC, USA, 11–13 August 2010.
10. Genkin, D.; Shamir, A.; Tromer, E. Acoustic cryptanalysis. *J. Cryptogr. Eng.* **2017**, *30*, 392–443. [CrossRef]
11. Levina, A.; Mostovoi, R.; Sleptsova, D.; Tcvetkov, L. Physical model of sensitive data leakage from PC-based cryptographic systems. *J. Cryptogr. Eng.* **2019**, *9*, 393–400 [CrossRef]
12. Ometov, A.; Orsino, A.; Andreev, S.; Levina, A.; Borisenko, P.; Mostovoy, R. Mobile social networking under side-channel attacks: Practical security challenges. *IEEE Access* **2017**, *5*, 2591–2601. [CrossRef]
13. Levina, A.; Varyukhin, V.; Kaplun, D.; Zamansky, A.; van der Linden, D. A Case Study Exploring Side-Channel Attacks On Pet Wearables. *IAENG Int. J. Comput. Sci.* **2021**, *48*, 878–883.
14. Levina, A.B.; Sleptsova, D. Correlation Side-Channel Attack on Mifare Classic Cards. In Proceedings of the ISPIT 2015 Information Security Conference, St. Petersburg, Russia, 5–6 November 2015; pp. 53–56.
15. Maghrebi, H. Deep Learning based Side-Channel Attack: A New Profiling Methodology based on Multi-Label Classification. *IACR Cryptol. Eprint Arch.* **2020**, *2020*, 436
16. Maghrebi, H.; Portigliatti, T.; Prouff, E. Breaking cryptographic implementations using deep learning techniques. In Proceedings of the International 46 Conference on Security, Privacy, and Applied Cryptography Engineering, Hyderabad, India, 14–18 December 2016; Springer: Berlin/Heidelberg, Germany, 2016; pp. 3–26.