



## UvA-DARE (Digital Academic Repository)

### Tipping the Balance

*Imbalanced Classes in Deep Learning Side-channel Analysis*

Picek, S.; Heuser, A.; Jovic, A.; Bhasin, S.; Regazzoni, F.

#### DOI

[10.1109/MDAT.2023.3288808](https://doi.org/10.1109/MDAT.2023.3288808)

#### Publication date

2024

#### Document Version

Final published version

#### Published in

IEEE Design and Test

#### License

Article 25fa Dutch Copyright Act (<https://www.openaccess.nl/en/in-the-netherlands/you-share-we-take-care>)

[Link to publication](#)

#### Citation for published version (APA):

Picek, S., Heuser, A., Jovic, A., Bhasin, S., & Regazzoni, F. (2024). Tipping the Balance: Imbalanced Classes in Deep Learning Side-channel Analysis. *IEEE Design and Test*, 41(2), 32-38. <https://doi.org/10.1109/MDAT.2023.3288808>

#### General rights

It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

#### Disclaimer/Complaints regulations

If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: <https://uba.uva.nl/en/contact>, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.

# Tipping the Balance: Imbalanced Classes in Deep-Learning Side- Channel Analysis

## Stjepan Picek

Faculty of Science  
Radboud University  
6500 GL Nijmegen, The Netherlands

## Annelie Heuser

CNRS, Inria, IRISA  
University of Rennes  
35000 Rennes, France

## Alan Jovic

Faculty of Electrical Engineering and Computing  
University of Zagreb  
10000 Zagreb, Croatia

## Shivam Bhasin

Temasek Laboratories  
Nanyang Technological University  
Singapore 637553

## Francesco Regazzoni

Security by Design  
University of Amsterdam  
1012 WP Amsterdam, The Netherlands

## ALaRI Institute

Universita Della Svizzera Italiana  
6900 Lugano, Switzerland

### *Editor's notes:*

This article reports the latest progress on the imbalance problem in evaluating the performance of machine-learning-based side-channel analysis.

—Gang Qu, University of Maryland, USA

■ **MACHINE LEARNING, AND** more recently, deep learning, have become a standard option for profiling side-channel analysis (SCA) to evaluate the worst-case security. Machine-learning-based SCA has advantages over previous approaches like the template attack [1], especially in practical settings where the number of training traces is limited. The advantages of deep-learning-based approaches are even more

pronounced as such techniques can break protected implementations without feature selection and by using relatively small models (neural networks) [2]. However, the use of popular device leakage models brings in the issue of imbalanced data sets. For instance, Hamming weight (HW) or distance (HD) model follows a binomial distribution resulting in significantly more training samples in central classes. Furthermore, evaluating the performance of machine-learning-based SCA with standard machine-learning metrics like accuracy can be misleading. Unfortunately, this problem is not trivial to circumvent by “just” using the SCA metrics as the training process with them is difficult.

This work was originally published in IACR Transactions on Cryptographic Hardware and Embedded

*Digital Object Identifier 10.1109/MDAT.2023.3288808*

*Date of publication: 26 June 2023; date of current version:*

*21 February 2024.*

Systems 2019 [3]. Since then, multiple works have explored the influence of class imbalance on the performance of machine learning (and, most dominantly, deep learning). Before this work, it was not uncommon to report only the accuracy metric and, based on it, estimate the attack performance. Following this work, side-channel metrics are almost exclusively used when assessing SCA performance. There are multiple attempts to further reduce the impact of class imbalance through, for example, custom loss functions or data augmentation.

## Background

### Machine learning

The capability of a system to acquire its knowledge by extracting patterns from data is known as machine learning. Machine-learning algorithms can be divided into several categories based on their learning style. The most popular machine-learning algorithms in SCA follow the supervised learning paradigm. Today, the most popular machine-learning algorithms are various types of neural networks such as multilayer perceptron (MLP) and convolutional neural networks (CNNs). Before, other common techniques included random forest and support vector machines. For a survey on “traditional” machine learning in SCA, we refer readers to [4], while for a more recent overview of deep-learning-based SCA, we recommend [5]. A common metric to assess the performance of machine-learning algorithms is accuracy. Furthermore, precision, recall,  $F_1$ ,  $G$ -mean, Matthew’s correlation coefficient, and Cohen’s kappa score are widely used in machine learning when dealing with imbalanced data sets.

### Side-channel analysis

The SCA considers attacks that do not aim at the weaknesses of an algorithm but is implemented [6]. SCA shows that even for theoretically secure algorithms, observing the unintentional physical or side channel leakages (e.g., timing, power, and electromagnetic emanation) from their implementations could lead to the potential recovery of secret information. More precisely, the core idea of SCA is to compare some secret data-dependent predictions of the physical leakages and the actual (measured) leakage to identify the data most likely to have been processed. In practice,

SCA requires the ability to model the leakage (to come up with the predictions for data) and to have a good comparison tool (so-called distinguisher) to extract the secret information efficiently. In the context of machine-learning-based SCA, the two most common leakage models are the HW or HD and the identity (ID) leakage models. For the HW leakage model, the adversary assumes that the leakage is proportional to the sensitive variable’s HW. This leakage model results in nine classes for a single intermediate byte for the AES cipher. The HD leakage model results from value updates in a register. A typical approach for AES is to compute the value update to the final output from the S-box input of the last round in the state register. Like the HW leakage model, the HD leakage model results in nine classes for a single intermediate byte for the AES cipher. HW and HD model a binomial distribution where the central classes are much more populated than the corner classes. For the ID leakage model, an adversary considers the leakage as an intermediate cipher value. This leakage model results in 256 classes for a single intermediate byte for the AES cipher.

### Profiling SCA

Profiling SCA represents the strongest SCA type since it assumes that an adversary has access to a clone device. Moreover, the adversary can control all the inputs, such as random plaintexts and keys, to the clone device and observe the corresponding leakage. With this knowledge, the adversary builds a model of a device during the training phase. Then, in the attack phase, the adversary collects only a small number of traces from the attack device with an unknown secret key. By comparing the attack traces with the characterized model, the secret key is revealed. Due to the divide-and-conquer approach, where small parts of the secret key can be recovered independently, the attack becomes practical. Machine-learning techniques were adopted for profiling attacks, with the statistics of the unknown leakage distribution automatically learned from the profiling set. Informally, the main advantages of simpler machine-learning techniques over deep learning are fewer hyperparameters, which means less tuning effort. At the same time, they commonly require feature engineering and are less powerful than deep learning when dealing with protected targets.

## Machine-learning-based SCA

As commonly done in machine-learning-based SCA, we follow the supervised learning paradigm where the goal is classification into a finite number of discrete classes (labels). In SCA, the label is derived from the key and input through a cryptographic function (e.g., the `S-box` operation) and a leakage model. The objective of a learning algorithm  $Alg$  is to learn a parameterized function  $f_\theta$ . To train the machine-learning model  $f_\theta$ , the learning algorithm  $Alg$  has access to the data set drawn from a certain distribution. As we assume supervised learning, the data set is partitioned into disjoint subsets denoted as the training set (size  $N$ ), validation set (size  $V$ ), and test set (size  $Q$ ).

*Profiling phase.* In the profiling phase, the goal is to train a model  $f$  by adjusting its parameters  $\theta$ . This is done by evaluating the performance of the training data set and minimizing the error as recorded by a loss function. During the training, a machine-learning model is tested based on how well it generalizes on the unseen examples from the validation set.

*Attack phase.* In the attack phase, the goal is to predict label  $y$  based on previously unseen (attack) traces and the trained model  $f_\theta$ . The SCA metrics (e.g., key rank and guessing entropy) are used to assess the test outcomes. More formally, with  $Q$  traces in the attack phase, an attack outputs a key guessing vector  $g$  in decreasing order of probability where  $g_1$  denotes the most likely and  $g_{|K|}$  the least likely key candidate. The position of the correct key in the key guessing vector is called a key rank. To avoid the effects of randomness due to specific choices of attack traces, it is possible to run multiple independent evaluations and then average the results, which is the guessing entropy metric.

## Data sets

The experiments consider three data sets representing AES128 processing. The 50 most important features for all data sets are selected using the Pearson correlation coefficient.

### DPAcontest v4

The DPAcontest v4 data set provides measurements of a masked AES software implementation. Still, in this work, we consider the mask to be known, making the implementation unprotected, and thus can easily turn it into an unprotected scenario. The

measurements consist of 4,000 features around the S-box part of the algorithm execution.

### AES\_RD

The second data set is a software AES implementation protected with the random delay countermeasure. Each measurement consists of 3,500 features.

### AES\_HD

The third data set represents a hardware AES unprotected implementation. Each measurement consists of 1,250 features.

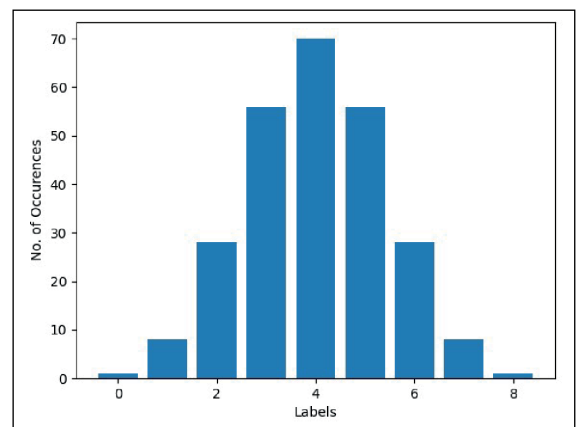
## Imbalance problem in SCA

### HW (HD) leakage model

A common option to mount a side-channel attack on AES is to consider the HW (HD) of the S-box output. Indeed, those leakage models work on a range of devices, even when not many details about the underlying implementations are known

$$y(p,k) = \text{HW}(\text{S-box}[p \text{ XOR } k]) \quad (1)$$

where  $p$  denotes the plaintext and  $k$  is the key. Then, considering i.i.d. values for plaintexts  $p$ , the labels  $y$  follow a binomial distribution  $\mathcal{B}(n,t)$ , where  $t = 0.5$  and  $n = 8$  for AES. Figure 1 gives the resulting numbers of class occurrences. There is a large imbalance among various labels (and thus, the expected number of measurements belonging to each label). The consequences for successful SCA become clear if we consider the performance of a well-performing algorithm versus an algorithm that always predicts the most occurring class. For a well-performing



**Figure 1. Distribution of class labels for AES with the HW/HD leakage models.**

algorithm, we would hope to reach accuracy better than random guessing (which equals 11.1% since we have nine classes). Moreover, even lower accuracy is often sufficient for a successful attack as the attack performance is evaluated over a number of measurements (accumulating the probabilities). On the other hand, an algorithm that would always predict the label to be equal to HW = 4 would have an accuracy of 27.34%. Naturally, an algorithm that always predicts the same HW label would never be able to break the target, despite the good accuracy result. Thus, the question is, can we make the algorithm perform better by considering some different metric or providing a similar (same) number of measurements for each label?

### Fighting the imbalance

There are two main approaches to improving the model performance and avoiding overfitting to the majority class when dealing with imbalanced data sets.

- *Data-level methods*: modify the measurements by balancing distributions (i.e., data augmentation).
- *Algorithm-level methods*: modify classifiers to remove (or reduce) the bias toward majority classes. Here, we provide details on the most common approaches and methods.

### Cost-sensitive learning and weight balancing

In this approach, the importance of a class is equal to its weight, determined as the combined weight of all the instances belonging to that class. Balancing the classes before classification can be made by assigning different weights to instances of different classes so that the classes have the same total weight.

### Data resampling approaches

Data resampling techniques can usually be divided into undersampling and oversampling. In undersampling, the number of instances for a majority class is reduced to become the same or similar to the minority class. In an imbalanced multiclass setting, undersampling reduces the number of instances in all classes except the one with the smallest number of instances. In oversampling, the number of instances in the minority class is increased to become equal or similar to the majority class. In an imbalanced multiclass setting, oversampling

increases the number of instances of all classes except the one with the highest number of instances. Oversampling may lead to overfitting when samples from the minority class are repeated; thus, synthetic samples could be used to prevent it.

### Random undersampling

Random undersampling undersamples all classes except the least populated one. While a simple technique, its main drawback is that it removes the measurements from the other classes.

### Random oversampling with replacement

Random oversampling with replacement oversamples the minority class by generating instances selected randomly from the initial set of minority class instances with replacement. All minority classes are oversampled to reach the number of instances equal to the highest majority class.

### Synthetic minority oversampling technique—SMOTE

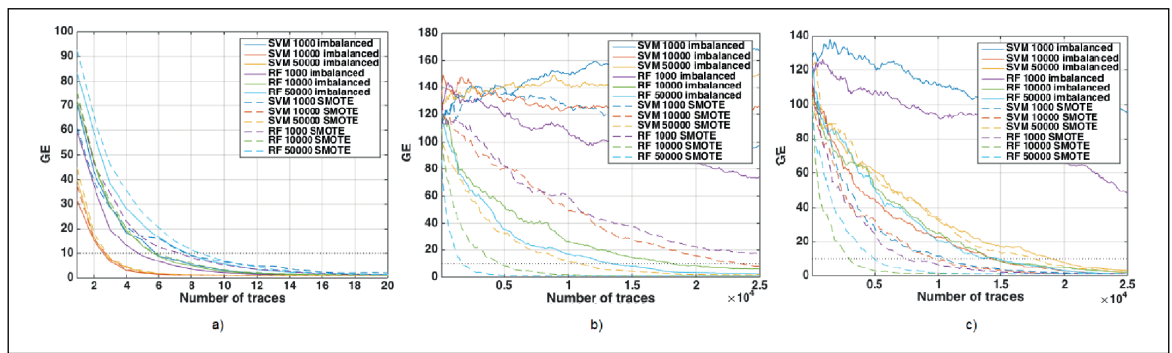
SMOTE is a resampling method that oversamples by generating synthetic minority class instances. The process works by taking each minority class instance and introducing synthetic instances along the line segments joining any/all of the  $k$  minority class' nearest neighbors (using Euclidean distance).

### SMOTE with an edited nearest neighbor

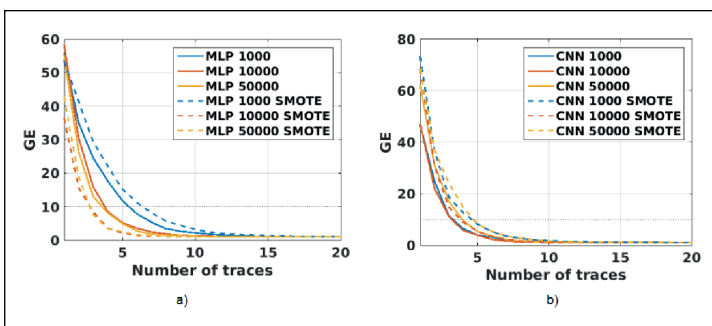
SMOTE + ENN combines oversampling used by SMOTE and data cleaning by the ENN method. The ENN cleaning method works by removing from the data set any instance whose class differs from the classes of at least two of its three nearest neighbors.

### Experimental results

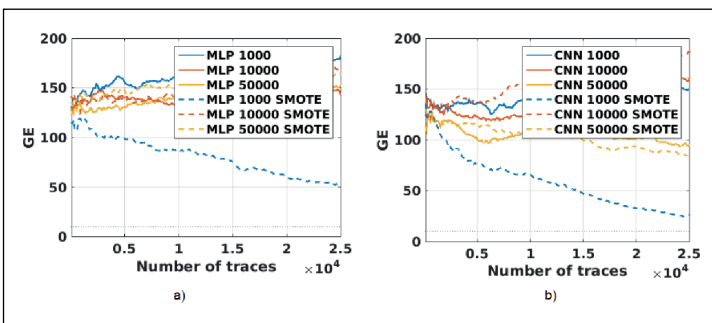
The experiments are conducted for the random forest, support vector machines, MLP, CNNs, and template attacks. The number of attack traces for all three data sets equals 25,000. For AES\_HD and DPAcontest v4, we evaluate with 1,000, 10,000, and 50,000 measurements in the profiling phase while for AES\_RD, we consider 1,000, 10,000, and 25,000 profiling measurements. The results indicate that machine-learning metrics designed to fight imbalance do not help much, except in the easiest cases. Next, when comparing techniques for data resampling, SMOTE performs the best, followed by



**Figure 2. Performance of imbalanced data set and balanced one with SMOTE for (a) DPAcontest v4, (b) AES\_RD, and (c) AES\_HD. Random forest and support vector machines.**



**Figure 3. Performance of imbalanced data set and balanced one with SMOTE for (a) MLP v4 and (b) CNN. DPAcontest v4.**



**Figure 4. Performance of imbalanced data set and balanced one with SMOTE for (a) MLP v4 and (b) CNN. AES\_RD.**

random oversampling, class weight balancing, and SMOTE + ENN. Hence, we provide detailed results for SMOTE and imbalanced data sets with various machine-learning approaches. In Figure 2, we provide results with random forest and support vector machines. For DPAcontest v4, all results indicate

good performance, which means that imbalance is not always a problem (even when considering a small training data set). Moreover, SMOTE does not seem to provide advantages in such cases. Already for the AES\_RD data set, the problem becomes more difficult for machine learning due to the random delay countermeasure. There, we see a clear advantage when balancing data with SMOTE and having a larger training data set. Similar results are obtained for AES\_HD since this data set does not have a countermeasure. Still, due to the hardware platform, the SNR is much lower (making it in a way to behave like a hiding countermeasure in the amplitude domain). Again, we notice that SMOTE significantly improves the performance, and a larger training data set gives better results. Also, the random forest seems to be a better option than support vector machines for the more difficult data sets.

When considering deep learning (MLPs and CNNs), we again notice that DPAcontest v4 is an easy data set with no need to balance it (Figure 3). On the other hand, for AES\_RD (Figure 4) and AES\_HD (Figure 5), we can notice how balancing the data set provides better attack performance. This is especially pronounced for AES\_HD as there, without balancing the data set, we cannot break the target with the allocated number of attack traces, while after balancing it, we require “only” around 10,000 measurements. Here, CNNs need fewer measurements than MLPs in most cases.

### Perspectives and long-term impact

This work [3] is one of the first works examining the benefits of data augmentation for SCA and the first work discussing how machine-learning metrics are not suitable to assess side-channel performance.

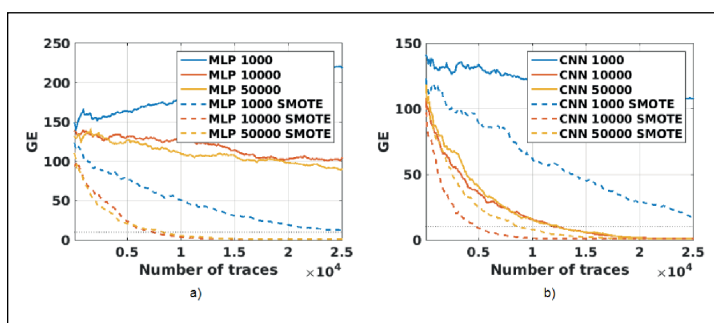


As such, this work has motivated several directions for further research. Already Cagli et al. [7] discussed the relevance of data augmentation for defeating targets protected with hiding countermeasures. The authors used shifting and addition transformation to generate synthetic measurements. While they worked in the HW leakage model, they did not report issues due to class imbalance, indicating data augmentation prevented overfitting.

Later, Robissout et al. [8] investigated how to introduce an online evaluation metric developed for the SCA context and how to use it to perform early stopping on existing CNNs found in the literature. This work was motivated by the fact that accuracy can be misleading and should be (potentially) completely removed from the training phase evaluation. Zhang et al. [9] proposed a novel cross-entropy ratio (CER) metric to evaluate the performance of deep-learning models for SCA. The authors showed that the new metric works stable even when the training data is imbalanced. Finally, the authors adapted the CER metric to a new loss function, called the CER loss function, designed specifically for deep-learning-based SCA. The proposed loss function is a good alternative to standard loss functions in deep-learning-based SCA for imbalanced data. Kerkhof et al. [10] continued the direction of the design of custom loss functions for SCA. The proposed loss function, the focal loss ratio, was designed to enable deep-learning models to learn from noisy or imbalanced data efficiently. Ito et al. [11] discussed the reasons for the negative effect of data imbalance in classification for deep-learning-based SCA and introduced the Kullback–Leibler divergence as a metric to measure this effect. Furthermore, the authors proposed a technique to solve data set imbalance at the inference phase (corresponding to data augmentation at the training phase), utilizing a likelihood function based on the key value instead of the HW/HD.

## Discussion

**OUR WORK [3]** selected for the top picks in Hardware and Embedded Security 2021 demonstrates how misleading it can be to use machine-learning metrics to assess SCA performance. Our results show that none of the common machine-learning metrics (including those recommended for the imbalanced



**Figure 5. Performance of imbalanced data set and balanced one with SMOTE for (a) MLP v4 and (b) CNN. AES\_HD.**

scenarios) provide adequate performance. Moreover, the results show that both traditional machine learning and deep learning are susceptible to imbalance problems occurring due to the binomial distribution of HW/HD leakage models. The results suggest that balancing the data set by data augmentation provides good results. More recent results also show the applicability of custom loss functions for deep-learning-based SCA. ■

## Acknowledgments

This work was supported in part by the Singapore National Research Foundation (“SOCure”) under Grant NRF2018NCR-NCR002-0001—www.green-ic.org/socure, in part by the Framework of the NWA Cybersecurity Call with project name PROACT under Project NWA.1215.18.014, in part by the Netherlands Organization for Scientific Research (NWO), and in part by the Netherlands Organization for Scientific Research NWO Project DISTANT (CS.019).

## References

- [1] S. Chari, J. R. Rao, and P. Rohatgi, “Template attacks,” in *Cryptographic Hardware and Embedded Systems—CHES 2002* (Lecture Notes in Computer Science), vol. 2523, B. S. Kaliski Jr., Ç. K. Koç, and C. Paar, Eds. Redwood Shores, CA, USA: Springer, Aug. 2002, pp. 13–28. [Online]. Available: [https://doi.org/10.1007/3-540-36400-5\\_3](https://doi.org/10.1007/3-540-36400-5_3)
- [2] G. Perin, L. Wu, and S. Picek, “Exploring feature selection scenarios for deep learning-based side-channel analysis,” *IACR Trans. Cryptograph. Hardw. Embedded Syst.*, vol. 2022, no. 4, pp. 828–861, Aug. 2022. [Online]. Available: <https://tches.iacr.org/index.php/TCHES/article/view/9842>

- [3] S. Picek et al., “The curse of class imbalance and conflicting metrics with machine learning for side-channel evaluations,” *IACR Trans. Cryptograph. Hardw. Embedded Syst.*, vol. 2019, no. 1, pp. 209–237, Nov. 2018. [Online]. Available: <https://tches.iacr.org/index.php/TCHES/article/view/7339>
- [4] B. Hettwer, S. Gehrer, and T. Güneysu, “Applications of machine learning techniques in side-channel attacks: A survey,” *J. Cryptograph. Eng.*, vol. 10, no. 2, pp. 135–162, Apr. 2019. [Online]. Available: <https://doi.org/10.1007/s13389-019-00212-8>
- [5] S. Picek et al., “SoK: Deep learning-based physical side-channel analysis,” *ACM Comput. Surv.*, vol. 55, no. 11, pp. 1–35, Feb. 2023. [Online]. Available: <https://doi.org/10.1145/3569577>
- [6] S. Mangard, E. Oswald, and T. Popp, *Power Analysis Attacks: Revealing the Secrets of Smart Cards*. New York, NY, USA: Springer, Dec. 2006. [Online]. Available: <http://www.dpabook.org/>
- [7] E. Cagli, C. Dumas, and E. Prouff, “Convolutional neural networks with data augmentation against jitter-based countermeasures: Profiling attacks without pre-processing,” in *Cryptographic Hardware and Embedded Systems—CHES 2017* (Lecture Notes in Computer Science), W. Fischer and N. Homma, Eds. Taiwan: Springer, Sep. 2017, pp. 45–68.
- [8] D. Robissout et al., “Online performance evaluation of deep learning networks for profiled side-channel analysis,” in *Constructive Side-Channel Analysis and Secure Design*, G. M. Bertoni and F. Regazzoni, Eds. Cham, Switzerland: Springer, 2021, pp. 200–218.
- [9] J. Zhang et al., “A novel evaluation metric for deep learning-based side channel analysis and its extended application to imbalanced data,” *IACR Trans. Cryptograph. Hardw. Embedded Syst.*, vol. 2020, no. 3, pp. 73–96, Jun. 2020. [Online]. Available: <https://tches.iacr.org/index.php/TCHES/article/view/8583>
- [10] M. Kerkhof et al., “Focus is key to success: A focal loss function for deep learning-based side-channel analysis,” in *Constructive Side-Channel Analysis and Secure Design*, J. Balasch and C. O’Flynn, Eds. Cham, Switzerland: Springer, 2022, pp. 29–48.
- [11] A. Ito et al., “Imbalanced data problems in deep learning-based side-channel attacks: Analysis and solution,” *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 3790–3802, 2021.

**Stjepan Picek** is an associate professor at Radboud University, 6500 GL Nijmegen, The Netherlands. He is a program committee member and reviewer for a number of conferences and journals, and a member of several professional societies. His research interests include security/cryptography, machine learning, and evolutionary computation.

**Annelie Heuser** is a permanent researcher at CNRS, IRISA, 35000 Rennes, France. Her main interests include the security of embedded devices, especially combined with side-channel information/attacks, machine and deep-learning algorithms, and malware.

**Alan Jovic** is an associate professor of computer science at the University of Zagreb, 10000 Zagreb, Croatia. His research interests include data science, machine learning, biomedical engineering, and software engineering. He is the Editor-in-Chief of *CIT Journal of Computing and Information Technology*.

**Shivam Bhasin** is a senior research scientist and program manager in cryptographic engineering at Nanyang Technological University (NTU), Singapore 637553. His research interests include embedded security and trusted computing. Bhasin has a PhD from Telecom Paristech, Paris, France, and a master’s from Mines Saint-Etienne, Saint-Etienne, France.

**Francesco Regazzoni** is a tenured assistant professor in security by design at the University of Amsterdam, 1012 WP Amsterdam, The Netherlands, and also affiliated with Università Della Svizzera Italiana, 6900 Lugano, Switzerland. His research activities are mainly focused on security, in particular, the security of embedded systems and cyber-physical systems, hardware trojans, postquantum cryptography, lightweight cryptography, and design automation for security.

■ Direct questions and comments about this article to Stjepan Picek, Faculty of Science, Radboud University, 6500 GL Nijmegen, The Netherlands; [stjepan.picek@ru.nl](mailto:stjepan.picek@ru.nl).