Power Analysis Attack Based on BS-XGboost Scheme

Yiran Li(✉)

Inner Mongolia University, Hohhot, Inner Mongolia Autonomous Region, China
18910280986@189.cn

Abstract. The power attack is a type of side-channel attack that involves measuring the power consumption of a device to extract secret information. By analyzing power consumption variations, an attacker can deduce the secret key used in the operation. In a class-imbalanced dataset, where the number of samples in one class is much smaller than the other, the power consumption patterns during cryptographic operations may be different for each class. The BorderLine-SMOTE data enhancement scheme was used to generate synthetic samples near the boundaries or at a greater distance from the existing samples, and through these modifications it helps to increase the diversity of the synthetic samples and reduce the risk of overfitting. XGBoost is then used as a classifier to classify the power curves. To evaluate the efficacy of the proposed method, it was applied to the DPA V4 dataset. The results indicated that the original data, when augmented using the Borderline-SMOTE + XGBoost approach, exhibited a substantial improvement in classification precision of up to 34%, outperforming DUAN's method.

Keywords: Borderline-SMOTE · Power Analysis Attack · Data Unbalanced · Data Augmentation Technology

1 Introduction

Side Channel Attack (SCA) is an attack to obtain the secret information of a cryptographic device by analyzing the power, operation time, electromagnetic radiation and other information leaked during the operation of the cryptographic device. The power attack is a type of side-channel attack that involves measuring the power consumption of a device to extract secret information. By analyzing power consumption variations, an attacker can deduce the secret key used in the operation. Power analysis attacks can be divided into two categories, one is non-profiling attacks and the other is profiling attacks.

In non-profiling attacks, the attacker accesses the target device, obtains the power consumption of the target device, and then analyzes the relationship between power consumption and key by statistical analysis. The typical one is a differential power analysis attack [1], in which the attack method is used to obtain sensitive information by obtaining a large number of power curves and dividing the power curves into two groups according to the distinguisher results, and then differencing the two groups of data according to

S. Goel and P. R. Nunes de Souza (Eds.): ICDF2C 2023, LNICST 571, pp. 186–196, 2024.
https://doi.org/10.1007/978-3-031-56583-0_12

the size of the difference value. Correlation Power Analysis (CPA) [2], an attack method to obtain sensitive information by calculating the magnitude of the Pearson Product-moment Correlation Coefficient (Pearson's r) between the power consumption curve and the hypothetical sensitive information. Mutual Information Analysis (MIA) [3], an attack method to obtain sensitive information by calculating the mutual information between the power consumption curve and the hypothetical sensitive information.

Profiling attacks are the most threatening of the power profiling attacks. In this attack, the attacker needs physical access to a pair of identical (similar) devices, which we call the profiling device and the target device. The whole attack consists of two phases (analysis and attack phases). In the first phase, the attacker analyzes the profiling device and uses it to determine the characteristics of the power leak; in the second phase, the attacker uses the model built in the first phase to attack the target device. Typical profiling attacks include Template attacks [4] (TA) and Stochastic models [5]. And in recent years, machine learning is widely used in profiling attacks.

In 2012, HEUSER [6] applied SVM in power analysis attacks and compared SVM with template attacks. 2017, MARTINASEK [7] compared the effectiveness of k-Nearest with other machine learning models such as traditional SVM in attacks and obtained that k-Nearest is a better model choice for power analysis attacks. In 2018, Benadjila [8] introduced a common framework to study and compare the effectiveness of Machine Learning methods against embedded implementations of cryptographic algorithms. In 2020, Perin [9] introduced a power analysis attack based on integrated learning, through which the problem of difficulty in obtaining hyperparameters when using deep learning large models can be solved. 2021, in [10] Lu used end-to-end models, the architecture could directly classify the traces that contain a large number of time samples while whose underlying implementation is protected by masking.

In 2019, Benjamin [11] successfully implemented non-profiling attacks using machine learning instead of Pearson correlation coefficients. 2021, Moos [12] used deep learning to evaluate the leakage of cryptographic devices.

Although many machine learning models have achieved a lot in power analysis attacks, when using machine learning as a classifier for power consumption profiles, the Hamming weight leakage model is usually used (Hamming weight leakage model is the Hamming weight of an intermediate byte when using a cryptographic algorithm run in the target device). However, due to the unbalanced nature of Hamming weight (classes with Hamming weight equal to 4 account for 70/256 of the total data, and classes with Hamming weight equal to 0 and 8 account for only 1/256 of the total data, respectively), this can cause an imbalance in the data. In a data set with class imbalance, the sample size of one class is much smaller than the other, which means that the power consumption pattern during the encryption operation may be different for each class.

To mitigate this risk, it is necessary to balance the classes in the dataset, which in 2020 DUAN [13] makes into a minority oversampling technique (SMOTE) to generate synthetic samples for minority classes.

SMOTE is a popular oversampling technique that generates synthetic samples for classes with a small number of samples by interpolating between existing ones. However, SMOTE is too random, and its principle is to randomly select a sample a from the minority class, find K nearest neighbors, then randomly select a sample b from the

nearest neighbors, connect samples a and b, and select a point on the straight line of a and b as the oversampling point, which is easy to generate wrong class samples, and the generated samples go into the majority class, they may not add any new information to the model, and may even lead to overfitting.

1.1 Our Contribution

In order to solve the problem that the generated samples in SMOTE are too similar to existing samples, which do not add new information to the model and may even lead to overfitting, the Borderline-SMOTE (BS) + XGBoost (XGB) scheme is proposed in this paper. The Borderline-SMOTE data enhancement scheme is used to generate synthetic samples near the boundary or at a large distance from existing samples The Borderline-SMOTE data enhancement scheme is used to generate synthetic samples near the boundary or at a large distance from the existing samples, and these modifications help to increase the diversity of synthetic samples and reduce the risk of overfitting. The scheme solves the class repeatability problem in Duan's scheme by using XGBoost as a classifier, which can perform leaf splitting optimization calculation without selecting the specific form of the loss function and relying only on the value of the input data [14], and can effectively improve the classification precision and the model success rate.

1.2 Structure of This Article

This paper is structured as follows. In Sect. 2, analyze the unbalanced feature of the output hamming weight of SBOX, power analysis attack based on machine learning and the imbalance of data class. In Sect. 3, introduce the BS + XGBoost scheme. Section 4 experimental results discussed, analysis and comparison are shown. Section 5 conclusion presented.

2 Background

2.1 Hamming-Weight Model

The Hamming (HM) weight model defines the Hamming weight of a binary data as the number of bits that are compared to 1 in this binary data. In power analysis attacks, the attacker assumes that the power consumption is proportional to the number of bits that are set in the processed data value, so the Hamming weight model is often used to represent the power consumption of the attacked data [13].

2.2 Power Analysis Attack Based on Machine Learning

Power analysis attack based on machine learning can be represented as it constructs a possible template for each possible class $c \in \{1, \ldots, C\}$, where the number of classes C depends on the assumed leakage model. Assume that for each class $c \in \{1, \ldots, C\}$, the attacker obtains a power consumption trace vector $\left\{l_c^i\right\}^{N_c}$, where N_c denotes the number of power consumption trace vectors for class C. Since the template attack relies on a

multivariate Gaussian noise model, the power consumption trace vectors are considered to be drawn from a multivariate distribution. Equations (1) and (2) give a more precise expression [14].

$$N(l_c|\mu_c, \Sigma_c) = \frac{1}{(2\Pi)^N 1/2|\Sigma_c|^{1/2}} \exp\left\{-\frac{1}{2}(l_c - \mu_c)^T \Sigma_x^{-1}(l_c - \mu_c)\right\} \quad (1)$$

$$\tilde{\mu}_c = \frac{1}{N_c}\sum_{n_c=1}^{N_c} l_{n_c}, \tilde{\Sigma}_c = \frac{1}{N_c}\sum_{n_c=1}^{N_c}\left(l_{n_c} - \tilde{\mu}_c\right)\left(l_{n_c} - \tilde{\mu}_c\right)^T \quad (2)$$

These templates are constructed based on the estimation of the expectation $\hat{\mu}_c$ and covariance matrices $\hat{\Sigma}_c$. The secrets recovery in the attack phase is performed using maximum likelihood estimation or the equivalent log-likelihood rule as shown in Eq. (3).

$$\log L_{k^*} \equiv \log \prod_{i=1}^{N_2} P(l_i|c) = \sum_{i=1}^{N_2} \log N(l_i|\mu_c, \Sigma_c) \quad (3)$$

where class C is calculated based on the given secrets guesses k^* and the input leakage model.

A common approach to reduce the computational complexity is to use the Hamming weight leakage model. Using the Hamming weight leakage model, the problem of reducing the computational complexity can be achieved by making assumptions about the entire intermediate values instead of just a few bits of the intermediate values.

2.3 The Imbalance of Data Class

There are two types of imbalances in a dataset. One is class imbalance, which occurs when some classes have more samples than others. The other type is an imbalance within the same class, where there are significantly fewer samples of some subsets than others in the same class. In an unbalanced dataset, the classes with more samples are called majority classes, while classes with fewer samples are called minority classes. Most research in the imbalanced domain focuses on these two classes because multi-class problems can be simplified into two-class problems. Generally, the minority class labels are positive, while the majority class labels are negative. Table 1 shows a confusion matrix for a two-class problem. The first column of the table displays the real class label of the sample, and the first row shows its predicted class label. TP and TN represent the number of correctly classified Positives and Negatives, respectively, whereas FN and FP represent the number of incorrectly classified Positives and Negatives, respectively [14].

Accuracy = (TP + TN)/(TP + FN + FP + TN)
Recall = TP/(TP + FN)
Precision = TP/(TP+FP)

When the data is extremely unbalanced, the samples of the majority class are easier to predict, while the samples of the minority class are less predictable. In situations where the dataset is highly unbalanced, even if the classifier correctly classifies the majority class samples and incorrectly classifies all the minority class samples, the accuracy of the classifier remains high. In this case, the accuracy does not reflect the reliable prediction of the minority class, which leads to a biased model learning.

Table 1. Confusion matrix for two-class problem

	Predicted Positive	Predicted Negative
Positive	TP	FN
Negative	FP	TN

2.4 The Imbalance of Hamming-Weight Value

Since the Hamming weight model is defined as the number of bits 1 in a byte, and a byte has 256 possibilities corresponding to the values 0 to 255, but the Hamming weight is only 0, 1, 2, 3, 4, 5, 6, 7 and 8, which are the nine cases, the distribution of Hamming weight for 256 data is unbalanced, as shown in Table 2. The classification of Hamming weight of 0 and 8 is the smallest, which is only 1/256 of the data respectively, while the Hamming weights of 4 is the most, which is 70/256 of the data.

Table 2. The number of HM weight values in one byte

HM weight	0	1	2	3	4	5	6	7	8
number	1	8	28	56	70	56	28	8	1
P_i	$\frac{1}{256}$	$\frac{8}{256}$	$\frac{28}{256}$	$\frac{56}{256}$	$\frac{70}{256}$	$\frac{56}{256}$	$\frac{28}{256}$	$\frac{8}{256}$	$\frac{1}{256}$

3 BS-XGboost Scheme

In order to solve the problem of model bias caused by data imbalance in the process of using machine learning for power analysis attacks, the authors achieve this through the BS-XGboost Scheme, which improves interpolation and random under-sampling experimental sample rebalancing at the data side and increases the fit using a combined optimization model at the model side.

3.1 Borderline-SMOTE

Borderline-SMOTE is an improvement based on the random sampling algorithm. It fully considers the problem of class repeatability caused by the distribution characteristics of adjacent samples and uses the method of identifying minority class samples to avoid such repeatability. The synthesis principle of boundary samples is illustrated in Fig. 1.

Assuming that S is a sample set, S_{min} is a minority class sample set, $S_{max\,i}$ is an adjacent majority sample set, m is the total number of the adjacent samples, x_i is all attributes of the sample, x_{ii} is all attributes of the adjacent sample, x_n is the adjacent sample, R_{ii} is taken as 0.5 or 1, The steps of the synthesis algorithm are as follows [15]:

1) Assume that $x_i \in S_{min}$, and determine the nearest sample set S_{NN}, and $S_{NN} \subset S$;

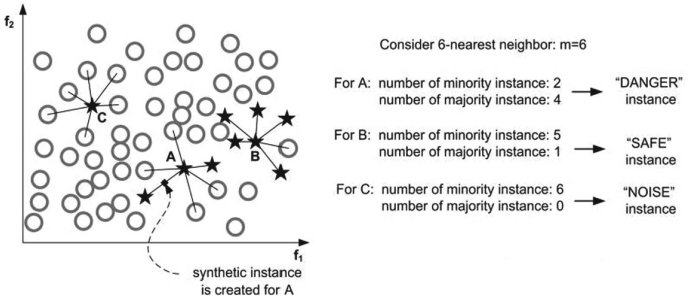

Fig. 1. The synthesis of boundary samples

2) For each sample x_i, determine the number of the nearest sample set that belongs to the majority class, that is $\left|S_{\mathrm{NN}} \cap \mathrm{S}_{\max j}\right|$;
3) Select, $x_i : \frac{m}{2} < \left|S_{NN} \cap S_{\max j}\right| < m$, and composite minority class samples. The difference between attribute j of x_i and x_n is recorded as $d_{ij} = x_i - x_{ij}$, so the new synthetic minority sample $h_{ij} = x_i + d_{ij} \times rand\left(0, R_{ij}\right)$.

3.2 XGBoost

The XGBoost algorithm was proposed by Chen et al. [16] and is widely used in regression and classification problems based on classification and regression powers. The objective function of the XGBoost algorithm consists of two components, the loss function and the regularization, with the expression

$$o_{bj} = \sum_{i=1}^{n} \ell\left(y_i, \hat{y}_i^t\right) + \sum_{i=1}^{l} \Omega(f_i) \tag{4}$$

$$\Omega(f_i) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} \varpi_j^2 \tag{5}$$

where: $\sum_{i=1}^{n} \ell\left(y_i, \hat{y}_i^t\right)$ forms the loss function of the training sample; $\sum_{i=1}^{t} \Omega(f_i)$ forms the regularization term; γ and λ are the regularization coefficients; the number of leaf nodes is labeled T; ϖ_j is the vector value of the j^{th} node of the decision tree.

The XGBoost objective function is expanded by Taylor's formula to obtain a convex optimization function, so in order to find the ϖ_i that minimizes the objective function, the derivative of ϖ_i and make the derivative function equal to zero can be obtained.

$$\varpi_j = -\frac{\sum_{i \in I_j} g_i}{\sum_{i \in I_j} h_i + \forall\lambda} \tag{6}$$

$$o_{bj}^{*} = -\frac{1}{2}\sum_{j=1}^{T} \frac{\left(\sum_{i \in S_j} g_i\right)^2}{\sum_{i \in I_j} h_i + \lambda} + yT \tag{7}$$

where: I_i denotes the set of samples of leaf nodes; g_i denotes the number of first-order partial layers of samples contained in leaf node i; h_i denotes the number of second-order braiding derivatives of samples contained in leaf node i; o_{hi} can indicate the superiority of a tree model, and the smaller the o_{hi} value, the better the model. By leading the greedy algorithm and iteratively selecting the optimal structure, the final gain formula is derived as Eq. (8)

$$Gain^* = \frac{1}{2}\left(\frac{\left(\sum\limits_{i\in I_i} g_i\right)^2}{\sum\limits_{i\in I_L} h_i + \lambda} + \frac{\left(\sum\limits_{i\in I_R} g_i\right)^2}{\sum\limits_{i\in I_n} h_i + \lambda} - \frac{\left(\sum\limits_{i\in I_j} g_i\right)^2}{\sum\limits_{i\in I_j} h_i + \lambda}\right) - \gamma \tag{8}$$

where I_L and I_R denote the set of left and right subtree leaf nodes, respectively.

3.3 BS-XGBoost Scheme

The BS-XGBoost Scheme in this paper is shown in Fig. 2. The whole model consists of two parts.

1. data augmentation of the input data by the Boorderline_SMOTE algorithm;
2. Classification of the augmented data using the XGBoost algorithm as a classifier.

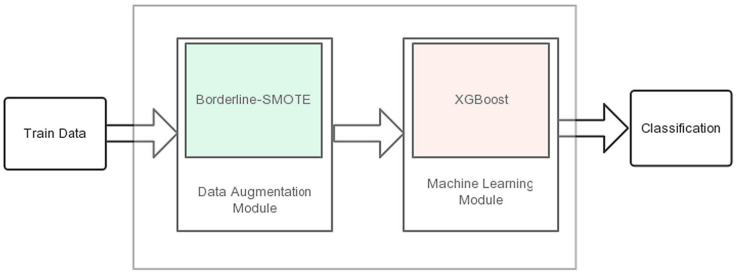

Fig. 2. BX-XGBoost Scheme

4 Experimental Results and Analysis

The data set used in this paper is the DPA Contest v4 data set supplied by the DPA contest. The DPA contest organizes an international academic competition that began in August 2008 and is jointly sponsored by the French National Academy of Sciences and the Paris Institute of Advanced Telecommunications. Its official website is http://www.dpacontest.org [17].

The comparison of the scenarios was evaluated using the classification precision and the model success rate.

4.1 Interesting Points Selection

Due to the high dimensionality of the data in the DPA V4 dataset, with 430,000 dimensions per curve, in order to avoid the problem of high data dimensionality and the model falling into the curse of dimensionality, the correlation analysis between the labels and the data interesting points was performed using the Pearson correlation system [18], calculated as (9)

$$p_{X,Y} = \frac{\mathrm{cov}(X, Y)}{\sigma_X \sigma_Y} = \frac{E[(X - u_X)(Y - u_Y)]}{\sigma_X \sigma_Y} \tag{9}$$

In Eq. (6): cov(X, Y) denotes the difference of agreement between data column X and data column Y samples; σ_X and σ_Y denotes the standard deviation of data column X and data column Y samples, respectively; u_X and u_Y denotes the mean of data column X and data column Y samples, respectively, and E denotes the mean value function.

The 600 dimensions with the largest label-related coefficients are selected as interesting points based on the statistical Pearson's r to achieve the purpose of interesting points selection and avoid the impact of repeated interesting points on machine learning performance, and the calculated Pearson coefficient values are shown in Fig. 3.

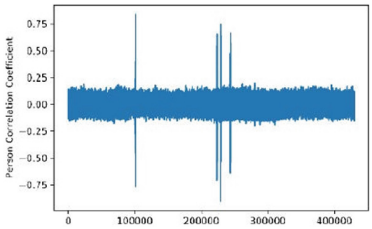

Fig. 3. Pearson's r of traces and labels

4.2 Data Augmentation Based on Borderline-SMOTE

From the DPA V4 dataset, 1000 power consumption curves were selected and the voltage distribution of their different Hamming weight data was viewed through scatter plots. Figure 4-a shows the distribution of the original data, which shows that the data is rather messy and random, and this kind of data is not conducive to machine learning training.

Figure 4-b shows the distribution after SMOTE data enhancement, which can see that the data increased and the distribution has been regular. Since SMOTE is too random, a sample a is randomly selected from the minority class, and after finding K nearest neighbors, a sample b is randomly selected from the nearest neighbors, connecting samples a, b, and choosing a point de on the ab line as the oversampling point. This is easy to generate the wrong class of samples, and the generated samples go into the majority class, as shown in the red figure in Fig. 4-b.

Figure 4-c shows the distribution after Borderline-SMOTE data enhancement. Since Borderline-SMOTE creates synthetic examples only on the decision boundary between

two classes, instead of blindly generating new synthetic examples for the minority class, it eliminates the problem of data overlap, as shown by the red circle in Fig. 4-c, where there are no longer generated samples that have gone into the majority class.

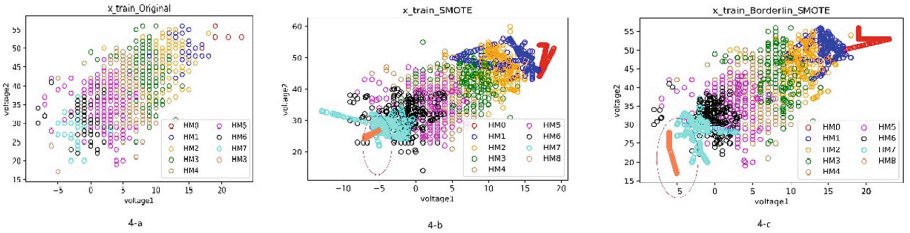

Fig. 4. The Voltage distribution of the HM (original data (4-a), data augmentation by SMOTE (4-b), data augmentation by Borderline-SMOTE (4-c))

4.3 The Classification Precision

From the DPA contest V4 data set, 800 and 1000 traces are randomly selected as the training set, and 600 interesting points are chosen for each trace by Pearson's r introduced in Sect. 4.1. The classification precision of the model is shown in Fig. 5. The classification Precision is shown in Fig. 5. From Fig. 5, we can see that the classification Precision of BS_XGBoost is higher than that of the SMOTE-RF scheme because the problem of the generated samples entering the majority class is solved in Sect. 4.2, the precision of label 1 and label 8 increases significantly.

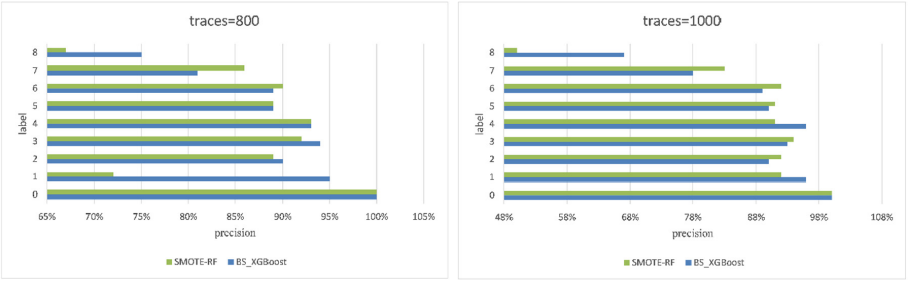

Fig. 5. Classification Precision of the Two Schemes

4.4 Model Success Rate with Different Interesting Points

From the DPA contest V4 data set, 800, 1000, 1500 and 2000 traces were randomly selected as the training set, and 200, 300, 400 and 500 interesting points were selected by Pearson's r as introduced in Sect. 4.1 respectively for each trace. The four sets of

data were augmented with Borderline-SMOTE, and the augmented data were substituted into the XGBoost model for training, and then the trained model was tested with another 1000 curves to obtain the the model success rate. As shown in Fig. 6, the BS-XGBoost scheme outperforms the SMOTE-RF scheme for different interesting points and different numbers of original curves.

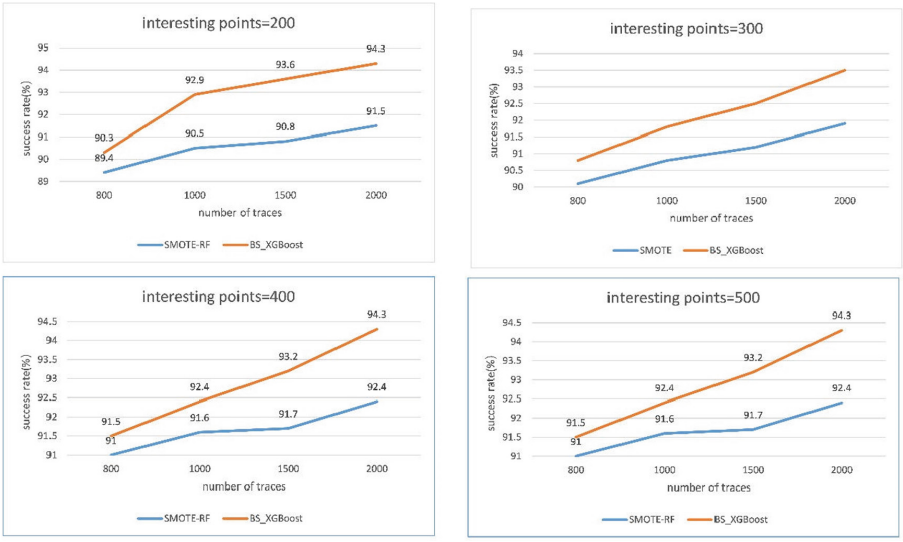

Fig. 6. Model success rate of the two schemes

5 Conclusion

The data in the power analysis attacks may be unbalanced due to classification. In this paper, BS_XGBoost scheme is used to augment the data, which uses the minority class samples on the boundary to synthesize the new samples, so as to improve the class distribution characteristics of samples. The scheme effectively solves the imbalance of data and thus solves the problem of model learning deviation. After the data augmentation of the proposed scheme, the data discrimination between HM = 8 and HM = 7 is significantly augmented. Compared with the scheme in literature [13], using machine learning to carry out model training on the data augmentation of the proposed scheme has significantly improved the classification precision and model success rate not only when the trace number is different (e.g. 800, 1000), but also when the interesting points number is different.

References

1. Kocher, P., Jaffe, J., Jun, B.: Differential power analysis. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 388–397. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48405-1_25

2. Brier, E., Clavier, C., Olivier, F.: Correlation power analysis with a leakage model. In: Joye, M., Quisquater, J.-J. (eds.) CHES 2004. LNCS, vol. 3156, pp. 16–29. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-28632-5_2
3. Batina, L., Gierlichs, B., Prouff, E., Rivain, M., Standaert, F.-X., Veyrat-Charvillon, N.: Mutual information analysis: a comprehensive study. J. Cryptol.Cryptol. **24**(2), 269–291 (2010). https://doi.org/10.1007/s00145-010-9084-8
4. Chari, S., Rao, J.R., Rohatgi, P.: Template attacks. In: Kaliski, B.S., Koç, çK., Paar, C. (eds.) CHES 2002. LNCS, vol. 2523, pp. 13–28. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-36400-5_3
5. Schindler, W., Lemke, K., Paar, C.: A stochastic model for differential side channel cryptanalysis. In: Rao, J.R., Sunar, B. (eds.) CHES 2005. LNCS, vol. 3659, pp. 30–46. Springer, Heidelberg (2005). https://doi.org/10.1007/11545262_3
6. Heuser, A., Zohner, M.: Intelligent machine homicide. In: Schindler, W., Huss, S.A. (eds.) COSADE 2012. LNCS, vol. 7275, pp. 249–264. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29912-4_18
7. Martinasek, Z., Zeman, V., Malina, L., et al.: K-nearest neighbors algorithm in profiling power analysis attacks. Radioengineering **25**(2), 365–382 (2016)
8. Benadjila, R., Prouff, E., Strullu, R., Cagli, E., Dumas, C.: Deep learning for side-channel analysis and introduction to ASCAD database. J. Cryptogr. Eng.Cryptogr. Eng. **10**(2), 163–188 (2019). https://doi.org/10.1007/s13389-019-00220-8
9. Perin, G., Chmielewski, U., Picek, S.: Strength in numbers: improving generalization with ensembles in machine learning-based profiled side-channel analysis (2020)
10. Lu, X., Zhang, C., Cao, P., et al.: Pay attention to raw traces: a deep learning architecture for end-to-end profiling attacks (2021)
11. Timon, B.: Non-profiled deep learning-based side-channel attacks with sensitivity analysis (2019)
12. Moos, T., Wegener, F., Moradi, A.: DL-LA: deep learning leakage assessment: a modern roadmap for SCA evaluations. In: Cryptographic Hardware and Embedded Systems. Universitätsbibliothek der Ruhr-Universität Bochum (2021)
13. Duan, X., Chen, D., Fan, X., et al.: Research and Implementation on power analysis attacks for unbalanced data. Secur. Commun. Netw. **2020**(3), 1–10 (2020)
14. Zhou, Z.: Machine Learning, pp. 29–30. Tsinghua Press, Beijing (2016)
15. Han, H., Wang, W.-Y., Mao, B.-H.: Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning. In: Huang, D.-S., Zhang, X.-P., Huang, G.-B. (eds.) ICIC 2005. LNCS, vol. 3644, pp. 878–887. Springer, Heidelberg (2005). https://doi.org/10.1007/11538059_91
16. Chen, T., Gueslrin, G.: XGBoost: a scalable tree boosting system. In: The 22nd ACM SIGKDD International Conference, pp. 758–794. ACM, New York (2016)
17. http://www.dpacontest.org/home/
18. Rodgers, L., Nicewander, W.A.: Thirteen ways to look at the correlation coefficient. Stat **42**(1), 59–66 (1988)