IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2023.0322000

Improving Location-based Thermal Emission Side-Channel Analysis Using Iterative Transfer Learning

KAI ZHANG[1,2], TUN-CHIEH LOU[3], CHUNG-CHE WANG[3], JYH-SHING ROGER JANG[3], (Member, IEEE), HENIAN LI[4], LANG LIN[5], AND NORMAN CHANG[5]

[1]Yiwu Industrial and Commercial College, Yiwu, Zhejiang 322000, China
[2]Graduate Institute of Network and Multimedia, National Taiwan University, Taipei 106, Taiwan
[3]Department of Computer Science & Information Engineering, National Taiwan University, Taipei 106, Taiwan
[4]Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611, USA
[5]Ansys Inc, Canonsburg, PA 15317, USA

Corresponding author: Kai Zhang (e-mail: d13944013@ntu.edu.tw)

This work was supported in part by Yiwu City Science and Technology Program under Grant 24-3-160, and in part by Jinhua City Science and Technology Program under Grant 2024-4-265.

ABSTRACT This study applies machine learning to side-channel attacks and proposes an iterative transfer learning method for deep learning models. This study leverages the similarity in training patterns across bytes by first training on a single byte and then using the resulting model as a pretrained foundation for the remaining bytes. This approach enables effective model training with smaller amounts of data while reducing the measurement-to-disclosure (MTD, i.e., the minimum number of traces needed for successful key recovery) in the attack phase. With sufficient data, iterative transfer learning reduces MTD from 55 to 54 using MLP and from 125 to 83 using CNN. Even under limited data conditions, it successfully breaks AES-128 while reducing training samples from 13,600 to 2,000, achieving an average MTD of 635, whereas traditional methods fail. Experimental results demonstrate that the iterative transfer learning approach addresses the persistent data scarcity challenge in deep learning, significantly expanding the applicability of deep learning methods in side-channel attack scenarios.

INDEX TERMS Side-channel Attack, Iterative Transfer Learning, Deep Learning, Thermal Map Image, Power Consumption Map Image

I. INTRODUCTION

A. RESEARCH MOTIVATION

As technology advances, people rely more on electronic devices in daily life. However, as semiconductor manufacturing improves, security concerns also increase. Hackers can cause serious consequences by attacking everything from personal accounts to military systems. Therefore, it is important to identify vulnerabilities in integrated circuit (IC) design.

Side-channel attacks (SCA) are among the most prominent threats, as they can efficiently extract encryption-related information from ICs. During encryption operations, physical data leakage—such as electromagnetic emissions, power consumption, or temperature variations—can be observed. These physical signals are closely related to the underlying algorithm, hardware, and processed data. By analyzing these signals, attackers can potentially retrieve encryption keys and other confidential information.

Machine learning is a technique that learns from data to identify underlying patterns, enabling efficient feature selection to accomplish specific tasks. Due to this capability, machine learning has been widely applied in various fields, such as speech recognition and image classification. In recent years, it has also been utilized in side-channel attacks, where the attack process can be framed as a classification problem. By analyzing physical leakage data, machine learning models predict key-related information. For instance, in this study's attack on AES-128 (Advanced Encryption Standard-128), the input consists of observed physical signals, and the model aims to predict the output values of each byte after passing through the substitution box (S-box).

Recent deep learning models, including multilayer perceptrons (MLP) and convolutional neural networks (CNN), have become focal points in side-channel attack research. Their complex architectures allow for more effective feature

extraction, enhancing attack performance. Most studies aim to improve key recovery efficiency, assuming that enough training data is available. However, in practice, collecting enough data is often challenging. Without sufficient data, deep learning models may not generalize well, which reduces attack performance. Simulating data also requires a lot of time and computing resources. If side-channel attacks could work well with limited data, it would increase their real-world and research value.

Most existing studies train separate models for each byte without considering inter-byte correlations. This paper introduces an iterative transfer learning approach to deep learning-based side-channel attacks, aiming to reduce training data requirements while improving attack efficiency. We compare conventional machine learning models, deep learning models, and the proposed method in terms of attack performance. Furthermore, while power-based side-channel attacks and countermeasures have been extensively studied, research on temperature-based side-channel analysis remains limited. Therefore, this study also explores the performance differences of models trained on different datasets.

B. RESEARCH CONTRIBUTIONS

The contributions of this paper are as follows:

1) Proposed iterative transfer learning to reduce data requirements and accelerate training for deep learning models like multilayer perceptron (MLP) and convolutional neural network (CNN).
2) Proposed a Progressive Feature Selection method to balance efficiency and selection quality through staged feature reduction.
3) Compared different models and loss functions in side-channel attacks to provide a reliable benchmark for future research.
4) Explored various feature selection strategies to identify optimal machine learning features for improving attack accuracy.
5) Analyzed different physical leakage channels by applying side-channel attack models to power and temperature data for chip security insights.

C. SECTION OVERVIEW

This paper is divided into six sections:

- Section 1: Introduction. Provides an overview of the research topic, research motivation, and contributions of this study.
- Section 2: Related Work. Introduces various machine learning classification models used in side-channel attack research.
- Section 3: Dataset. Describes the datasets used in this study, including power consumption and temperature datasets.
- Section 4: Research Methodology. Details the methods and relevant knowledge applied in this study.
- Section 5: Experimental Design and Results Discussion. Presents the experimental setup, parameter configurations, model architectures, and a comparative analysis of the results.
- Section 6: Conclusion and Future Work. Summarizes the findings of this study and suggests potential future improvements.

II. RELATED WORK

This section introduces the fundamentals of side-channel attacks, relevant research on deep learning in side-channel analysis, the commonly used correlation power analysis method, optimized transfer learning approaches for deep learning applications, and the various loss functions used in the experiments.

A. SIDE-CHANNEL ATTACKS

Side-channel attacks can be categorized into profiling attacks and non-profiling attacks. Profiling attacks assume that the attacker possesses an identical device to the target and has full control over it. By adjusting various parameters, the attacker can collect a large amount of physical information to accurately break the target device. Template attacks (TA) [1] are a well-known example of this type of attack.

Non-profiling attacks, on the other hand, assume that the attacker can only collect physical leakage data without direct control over the target device. These attacks rely on statistical analysis to compute the correlation between hypothetical leakages and actual leakages. Differential power analysis (DPA) [2] and correlation power analysis (CPA) [3] are common examples of such attacks.

Since Paul Kocher first introduced side-channel attacks [4], they have been recognized as a powerful and practical method for breaking encryption. These attacks are non-invasive and do not require brute-force computation. Research on side-channel attacks has evolved from traditional techniques such as DPA, CPA, and template attacks to modern deep learning-based approaches. In recent years, deep learning has demonstrated outstanding performance in fields like speech and image recognition, leading to its application in side-channel attacks as well. This section provides a literature review on related research in this domain.

B. SIDE-CHANNEL ATTACKS USING MULTILAYER PERCEPTRONS

The multilayer perceptron (MLP) is one of the most common architectures in deep learning models. As shown in Figure 1, it consists of three main layers: the input layer, hidden layers, and the output layer. The input layer receives physical data, while the output layer predicts the probability of class labels. The neurons in the hidden layers update their weights based on the model's predictions and the actual labels, allowing the model to improve its accuracy over time.

In the context of side-channel attacks, MLPs require the selection of points of interest (POIs). POI selection is a preprocessing step that removes noise from the data, improving model performance while reducing input dimensionality. This

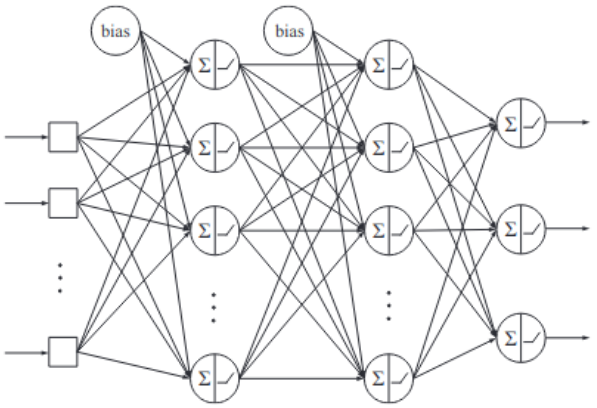


FIGURE 1. Multilayer Perceptron (MLP) Architecture for Side-Channel Analysis [5]

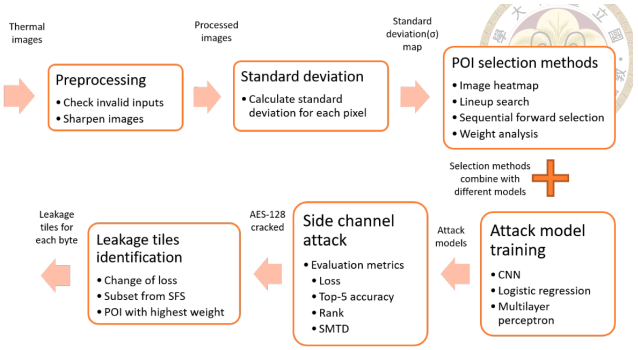


FIGURE 2. Experimental Workflow for Side-Channel Attack Using Machine Learning [8]

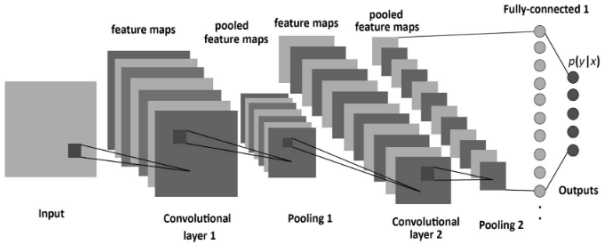


FIGURE 3. Convolutional Neural Network (CNN) Architecture for Image-Based Side-Channel Analysis [9]

process also shortens the training time, making the attack more efficient.

[6] pioneered the application of multilayer perceptrons (MLP) in side-channel attacks through a regression-based approach to predict S-box byte outputs. This methodology supersedes the conventional Hamming weight metric in Correlation Power Analysis (CPA) frameworks by employing MLP-generated predictions as refined power consumption estimators. The enhanced predictive capability of MLPs demonstrates superior alignment with actual physical leakage characteristics compared to traditional CPA implementations. [7] introduced the seminal classification-based MLP architecture for S-box analysis in side-channel contexts, implementing 256-class categorical labeling of byte values - an approach that has become paradigmatic in contemporary research.

Current MLP-based side-channel analyses predominantly utilize power radiation traces as primary datasets. However, [8] innovatively employed localized thermal variation data represented as two-dimensional thermal images corresponding to physical chip layouts. The study proposed advanced preprocessing techniques combining Laplacian filters and standard deviation metrics for feature selection, effectively reducing dimensionality from 40,000 pixels to 200 optimized input features. Furthermore, the research conducted comparative analyses of various operational points of interest (POIs) identification methods - critical vulnerabilities where information leakage manifests most significantly. Figure 2 illustrates the experimental workflow. Building upon this foundation, our current investigation extends the methodology through localized power consumption and thermal variation analysis, introducing novel training data reduction techniques.

C. SIDE-CHANNEL ATTACKS USING CONVOLUTIONAL NEURAL NETWORKS

Convolutional Neural Networks (CNNs) are a common machine learning model in the field of image processing. Unlike multilayer perceptrons (MLPs), which process one-dimensional data, CNNs can handle two-dimensional data, enabling them to capture spatial information. This capability allows CNNs to perform well in image-related tasks. The structure of a CNN, as shown in Figure 3, consists of an input layer, convolutional layers, pooling layers, and fully connected layers. The input layer processes raw physical data, while convolutional layers are composed of multiple kernels. The purpose of these kernels is to learn different features, with each kernel represented as a matrix that performs convolution operations on the input data to extract feature maps, such as horizontal, vertical, or diagonal patterns.

The pooling layer serves to reduce computational complexity while preserving crucial information. Common pooling methods include max pooling and average pooling. The max pooling layer iterates over the feature map with a defined window size and retains only the maximum value within the window, whereas the average pooling layer retains the average value within the window. If the pooling layer operates with a 2*2 window and no overlap between iterations, the feature map size is reduced to half of its original dimensions, significantly decreasing computational cost. The fully connected layer is similar to that in MLPs, where it connects all neurons to predict the probability of each class label.

Since CNNs demonstrate strong performance in image-related tasks, and the dataset used in this paper is also image-based, CNNs will be employed for side-channel attacks.

The study in [10] is one of the early works that applied CNNs to side-channel attacks. It compared CNNs with other commonly used machine learning classifiers, including ran-

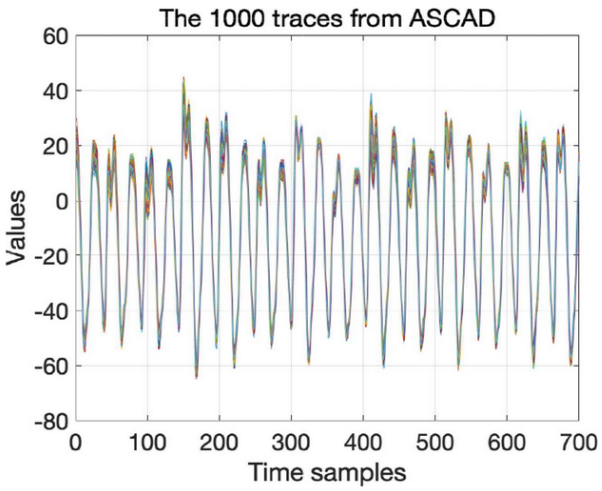


FIGURE 4. Example Power Trace from the ASCAD Public Dataset [11]

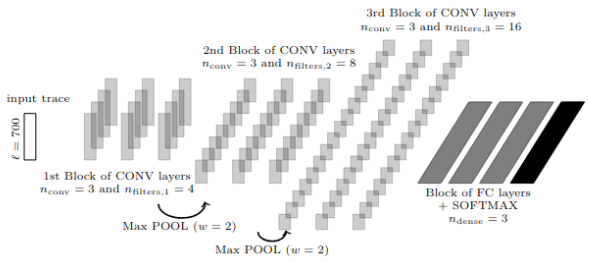


FIGURE 5. CNN Model Architecture Used for ASCAD Dataset Evaluation [11]

dom forests, autoencoders, and MLPs. In [11], CNNs were also used for side-channel attacks, and the authors introduced the publicly available ASCAD dataset, which provides a benchmark for side-channel attack research. The CNN-based benchmark model used in [11] is illustrated in Figure 5, while Figure 4 presents an example of the ASCAD dataset. Unlike the localized dataset used in this paper, ASCAD consists of time-frequency data.

The study in [12] introduced an innovation in CNN-based side-channel attacks by converting plaintext into a one-hot representation and concatenating it with the first layer of the fully connected network, as shown in Figure 6. In this approach, plaintext is treated as domain knowledge that aids CNNs in improving performance. Furthermore, unlike traditional side-channel attacks that use S-box outputs as labels, this study directly predicts the encryption key and successfully breaks the target system.

D. LOSS FUNCTIONS APPLIED TO SIDE-CHANNEL ATTACKS

In machine learning, the loss function is crucial as it measures the gap between the predicted probabilities and the actual labels. In other words, whether the model can successfully

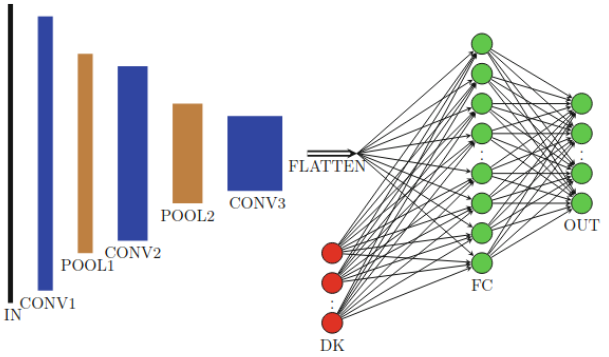


FIGURE 6. Model Structure Combining Domain Knowledge and CNN [12]

learn is highly dependent on the loss function. The following section will introduce the loss functions applied to side-channel attack classification problems in recent years.

1) Cross entropy (CE)

Cross entropy is one of the most commonly used loss functions in machine learning classification problems. Its computation is given by the formula in equation 1, where $p(x)$ represents the true label distribution, and $q(x)$ is the predicted probability distribution by the model. During training, the goal is to minimize cross entropy so that the distributions of $p(x)$ and $q(x)$ become as close as possible.

$$Cross\ entropy = -\sum_{z} p(x)\log(q(x)) \tag{1}$$

2) Cross Entropy Ratio (CER)

Cross Entropy Ratio (CER) is an improvement of cross entropy that separates the computation of the positive labels (true labels) and negative labels (false labels). It places the cross entropy of negative labels in the denominator and the cross entropy of positive labels in the numerator, as shown in equation 2. The aim is for the positive label and its predicted probability to be as close as possible, while the negative label and its predicted probability should differ as much as possible.

In [13], the authors proposed the Cross Entropy Ratio (CER) loss function and provided comprehensive experiments showing that CER can significantly improve model performance under conditions of imbalanced training data. However, in our study, since the label distribution is relatively uniform due to dataset generation design, we did not observe the same degree of improvement when applying CER. This suggests that the benefits of CER highlighted in [13] are highly dependent on the data distribution, and that for balanced datasets, conventional cross-entropy or ranking loss may remain preferable.

$$Loss(y, \hat{y}) = \frac{CE(k^*)}{\frac{1}{n}\sum_{i=1}^{n} CE(k)} \tag{2}$$

3) Ranking Loss

The two loss functions mentioned above both calculate the error between the true labels and the predicted probabilities. However, [14] proposes the ranking loss, which is computed as shown in equation 3. Here, $s(k^*)$ represents the model's predicted probability for the correct label, and $s(k)$ represents the predicted probability for the incorrect labels. In side-channel attacks, it is difficult to predict the correct label from a single data point. Typically, multiple data points are used, and their predicted probabilities are aggregated before ranking. Therefore, the concept of ranking loss is to compare the rankings of the true label and other labels, with the goal of ensuring that the true label has a higher ranking than the other labels.

$$Loss(s) = \sum_{\substack{k \in K \\ k \neq k^*}} \log_2 \left(1 + \exp\left(-\alpha(s(k^*) - s(k))\right)\right) \quad (3)$$

E. CORRELATION POWER ANALYSIS ATTACK

Correlation Power Analysis (CPA) is a type of non-invasive side-channel attack. It calculates the correlation between hypothetical leakages (predicted data) and actual leakage data to identify the most probable key values. The hypothesis in CPA assumes that the Hamming weight of the S-box byte output is correlated with the actual physical leakage data. The Hamming weight is computed as the number of "1"s in a byte. For example, the Hamming weight of 10010001 is 3, and for 11110001, it is 5.

To attack byte-0 using CPA in side-channel attacks, the steps are as follows:

1) Using a plaintext byte-0 and all possible keys (0-255), perform AddRoundKey and SubBytes operations, then calculate the Hamming weight of the S-box output.
2) Compute the Pearson product-moment correlation coefficient between all Hamming weights and every piece of physical data from the trace, as shown in Equation 4.
3) Repeat the above steps with more data samples.
4) The key with the highest correlation coefficient is the predicted key.

While the time required to compute the correlation coefficients in CPA is longer than for machine learning models, the advantage of CPA lies in its not requiring additional training data, thus saving time that would otherwise be spent on training. Although the MTD (Mean Time Delay) value for CPA is generally larger than that of machine learning models, CPA remains a stable and commonly used method for performing side-channel attacks.

$$\rho_{X,Y} = \frac{\mathrm{cov}(X, Y)}{\sigma_X \sigma_Y} = \frac{E[(X - \mu_X)(Y - \mu_Y)]}{\sigma_X \sigma_Y} \quad (4)$$

F. TRANSFER LEARNING

Neural network training for side-channel analysis often demands a substantial amount of labeled data to achieve high accuracy and generalization. However, collecting such extensive datasets can be impractical due to constraints such as hardware variations, limited measurement capabilities, and data privacy concerns. To address this issue, transfer learning has been explored as a technique to enhance data efficiency by leveraging knowledge from related domains or pre-trained models.

Several studies have demonstrated the applicability of transfer learning in mitigating the data requirements for side-channel analysis. Thapar, Alam, and Mukhopadhyay [15], as well as Yu et al. [16], investigated the use of pretraining models with data acquired from different devices before fine-tuning them on target device-specific datasets. Their approach demonstrated improvements in cross-device generalization, showing that shared representations can be learned from diverse hardware sources. Meanwhile, Garg and Karimian [17] explored the potential of leveraging pretrained models originally designed for general image recognition tasks, such as InceptionV3 and VGG16, as feature extractors for side-channel attack analysis. This approach capitalized on the ability of deep convolutional networks to learn hierarchical feature representations, thus reducing the dependence on large-scale domain-specific labeled data.

However, other studies have highlighted certain limitations associated with transfer learning in this domain. For instance, Hettwer et al. [18] pointed out that the effectiveness of transfer learning can be inconsistent, particularly when the feature distributions of the source and target datasets exhibit significant discrepancies. Such differences may lead to negative transfer, where the pre-trained model fails to generalize effectively, potentially degrading performance rather than improving it.

Unlike these existing methods, our proposed Iterative Transfer Learning (ITL) addresses these limitations by focusing on intra-task and intra-device model reuse. Rather than relying on external datasets or domain-agnostic pretraining, ITL captures inter-byte similarities within the same cryptographic operation. The byte-wise sequential reuse of weights allows ITL to significantly reduce training data while maintaining performance, especially in data-scarce conditions. This progressive and adaptive fine-tuning strategy distinguishes ITL as a lightweight and domain-specific alternative to traditional TL techniques.

III. DATASET INTRODUCTION

This section introduces the training datasets used in side-channel attacks, including the temperature dataset and the power consumption dataset.

A. DATASET GENERATION

In this study, we generated chip power maps and thermal maps using ANSYS RedHawk-SC simulation software. We then used these maps as datasets for machine learning–based side-channel attacks. Simulation software lets us create training data more quickly and efficiently.

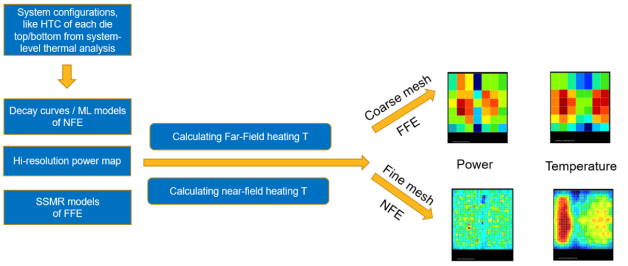


FIGURE 7. Flowchart of Temperature Generation [20]

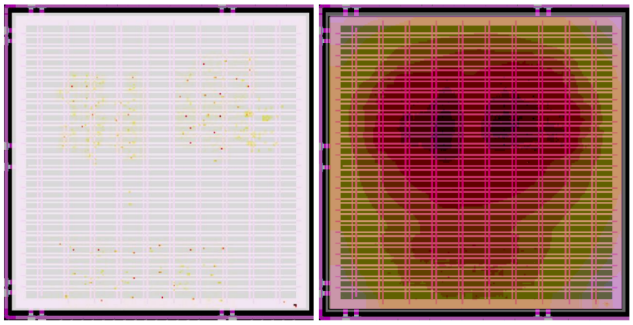


FIGURE 8. Chip Temperature and Power Consumption Map [20]

The process of generating physical data involves executing an AES encryption chip design, where the power consumption of each region is calculated based on the integrated circuit parameters, initial plaintext, and initial key. After generating the power map, [19] mentions that the generation of thermal maps needs to consider both near-field heating effects (NFE) and far-field heating effects (FFE). The far-field heating effect is handled using scaling with multi-dimensional interpolation and remapping (SSMR), while the near-field heating effect is modeled through a pre-trained decay model.

The predicted temperature is given by Equation 5, where T represents the final generated temperature, DT_{NFE} represents the temperature increase due to near-field heating effects, and DT_{FFE} (also referred to as DT_{SSMR}) represents the temperature increase due to far-field heating effects. T_{AMB} denotes the ambient temperature. Figure 8 illustrates the correspondence between power and temperature in the actual chip layout.

$$T = DT_{NFE} + DT_{FFE} + T_{AMB} \tag{5}$$

B. DATA DISTRIBUTION

The dataset used in this study consists of power data and temperature data, each containing 20,000 samples. Each sample includes a 128-bit plaintext, with the key also being 128-bit and kept consistent as follows:

$$key = [\prime 00\prime, \prime 11\prime, \prime 22\prime, \prime 33\prime, \prime 44\prime, \prime 55\prime, \prime 66\prime, \prime 77\prime, \prime 88\prime, \prime 99\prime, \prime AA\prime, \prime BB\prime, \prime CC\prime, \prime DD\prime, \prime EE\prime, \prime FF\prime] \tag{6}$$

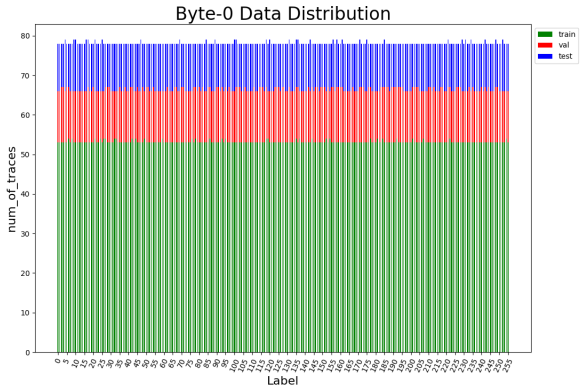


FIGURE 9. Label distribution of AES S-box Byte-0 output. Train/Val/Test = 5:1:1.

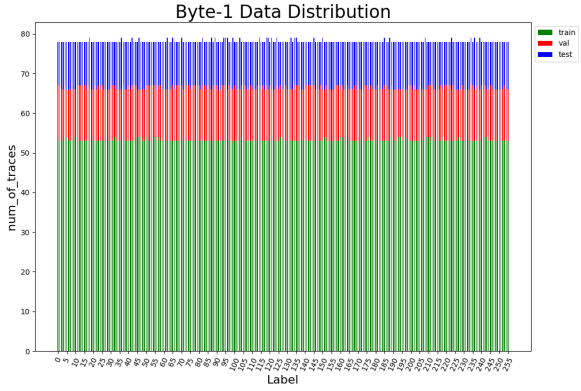


FIGURE 10. Label distribution of AES S-box Byte-1 output. Train/Val/Test = 5:1:1.

In this study, the training labels correspond to the output of the S-box, which consists of 128 bits. These bits are divided into 16 bytes, labeled from Byte-0 to Byte-15. Therefore, each plaintext is associated with 16 labels. Figures 9–24 illustrate the label distributions for the first 8 bytes and the last 8 bytes, respectively. The vertical axis represents the quantity, while the horizontal axis represents the labels. In the figures, the green section represents the training data, the red section represents the validation data, and the blue section represents the test data, which is also used during the attack phase. It can be observed that the label distribution for Byte-0 is relatively uniform.

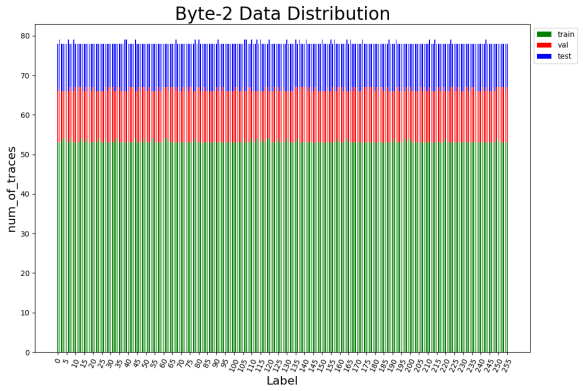


FIGURE 11. Label distribution of AES S-box Byte-2 output. Train/Val/Test = 5:1:1.

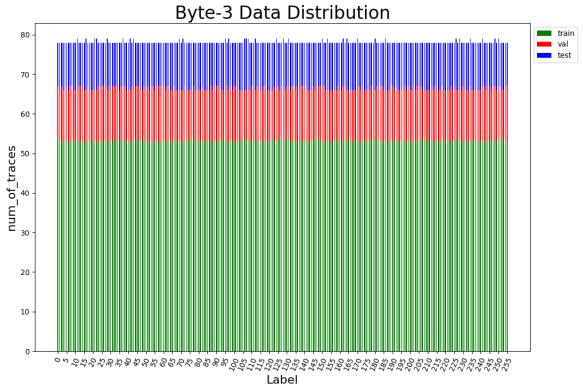


FIGURE 12. Label distribution of AES S-box Byte-3 output. Train/Val/Test = 5:1:1.

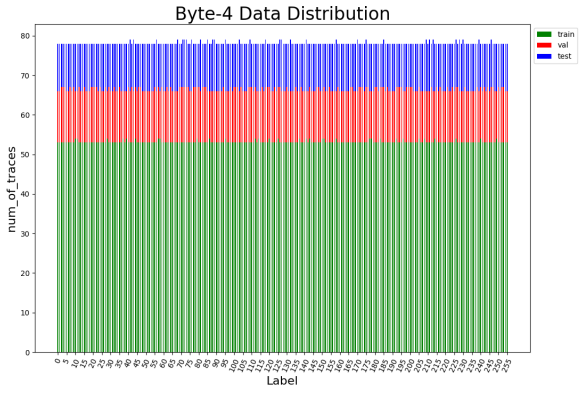


FIGURE 13. Label distribution of AES S-box Byte-4 output. Train/Val/Test = 5:1:1.

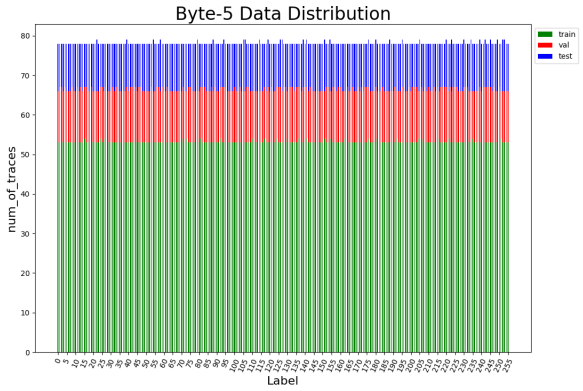


FIGURE 14. Label distribution of AES S-box Byte-5 output. Train/Val/Test = 5:1:1.

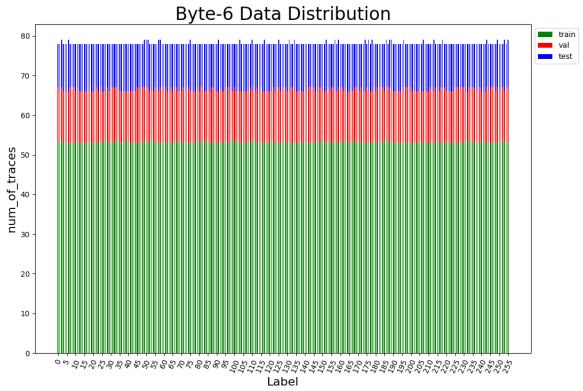


FIGURE 15. Label distribution of AES S-box Byte-6 output. Train/Val/Test = 5:1:1.

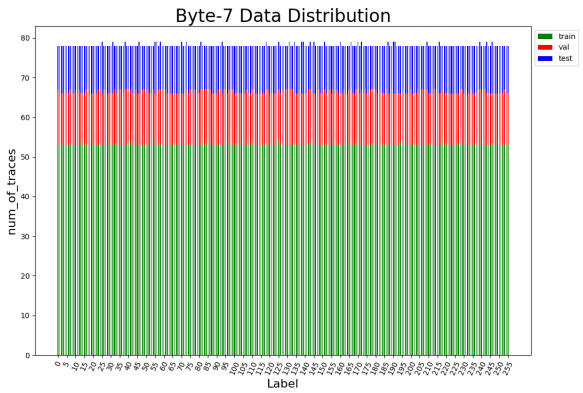


FIGURE 16. Label distribution of AES S-box Byte-7 output. Train/Val/Test = 5:1:1.

C. POWER CONSUMPTION MAP

Figure 25 is an example of a power consumption map. The image size is 201x201, with each pixel representing the power consumption (in watts) of a 100 square micron area. The power consumption values range from 0 to 0.0000402. In the image, many areas are yellow, indicating that the power consumption in those areas is 0, meaning no current flows through them and no power is consumed. On the other hand,

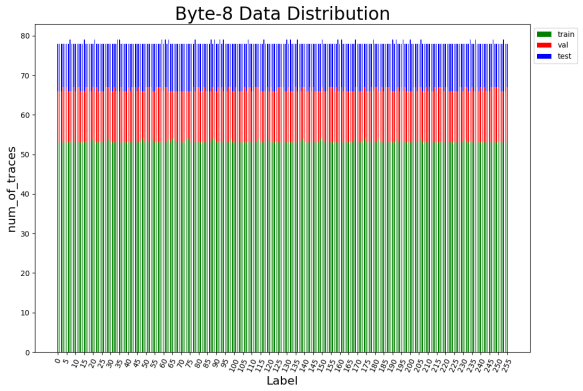


FIGURE 17. Label distribution of AES S-box Byte-8 output. Train/Val/Test = 5:1:1.

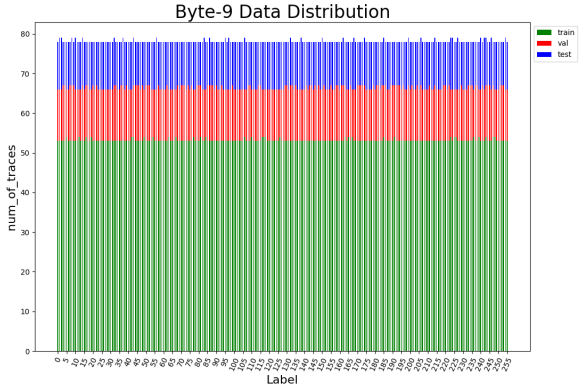


FIGURE 18. Label distribution of AES S-box Byte-9 output. Train/Val/Test = 5:1:1.

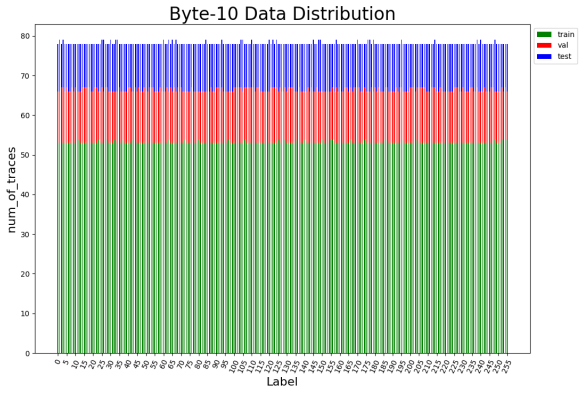


FIGURE 19. Label distribution of AES S-box Byte-10 output. Train/Val/Test = 5:1:1.

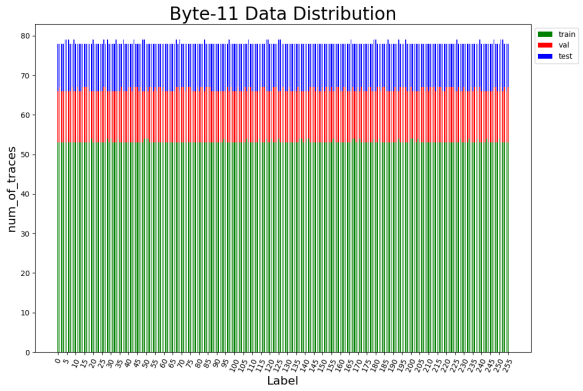


FIGURE 20. Label distribution of AES S-box Byte-11 output. Train/Val/Test = 5:1:1.

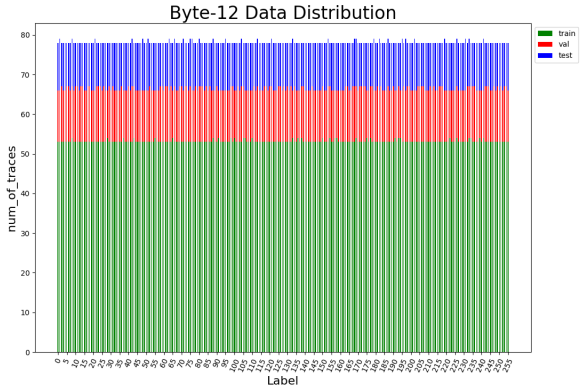


FIGURE 21. Label distribution of AES S-box Byte-12 output. Train/Val/Test = 5:1:1.

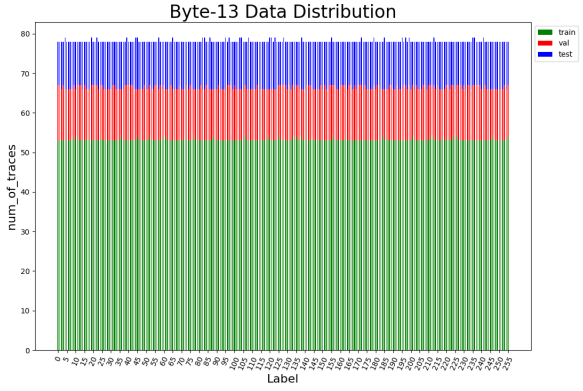


FIGURE 22. Label distribution of AES S-box Byte-13 output. Train/Val/Test = 5:1:1.

areas with darker colors represent regions with higher power consumption. The power consumption map allows for the rapid identification of important features (such as regions with power consumption greater than 0). In general, using power consumption map data for side-channel attacks is easier compared to using temperature datasets.

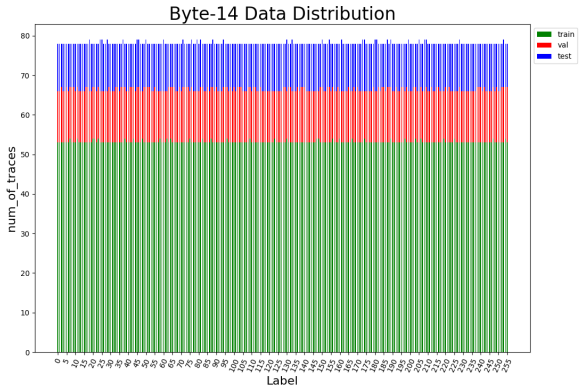


FIGURE 23. Label distribution of AES S-box Byte-14 output. Train/Val/Test = 5:1:1.

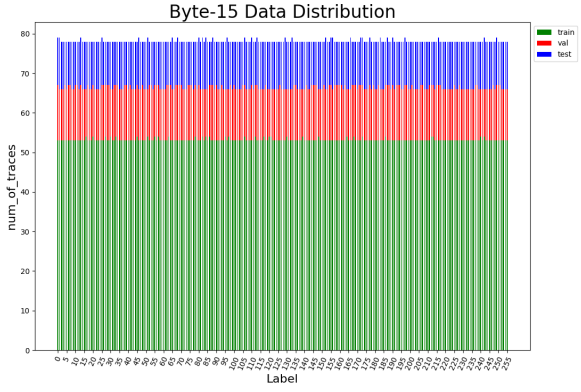


FIGURE 24. Label distribution of AES S-box Byte-15 output. Train/Val/Test = 5:1:1.

FIGURE 25. Sample Chip Power Consumption Map During AES Encryption (201*201 Pixels, Watts per 100 square micron Region)

D. TEMPERATURE MAP

Figure 26 is an example of a temperature map. The image size is also 201x201, with each pixel representing the average temperature (in degrees Celsius) of a 100 square micron area during AES encryption. Similar to the power map, the temperature range is from 27.15 degrees Celsius to 27.26 degrees Celsius. When compared to the power map, a significant difference in color distribution can be observed. The temperature map exhibits a coupling effect, where thermal energy diffuses outward through the medium, and regions with zero power consumption still show relatively high average temperatures. The temperature of each pixel is actually influenced by its neighboring areas.

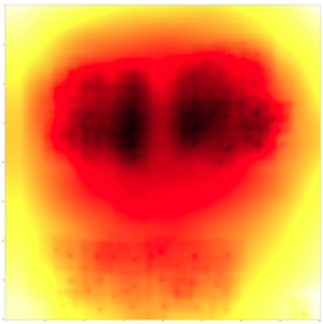


FIGURE 26. Sample Chip Temperature Map During AES Encryption (201*201 Pixels, degrees per 100 square micron Region)

TABLE 1. Coordinate Positions of the 16 Byte-Level Operating Points of Interest (POIs) on the 201*201 Temperature Map During AES-128 Encryption

Byte	0	1	2	3
Coordinate	(94,161)	(65,171)	(70,179)	(93,166)
Byte	4	5	6	7
Coordinate	(92,173)	(81,175)	(65,166)	(65,174)
Byte	8	9	10	11
Coordinate	(70,175)	(88,171)	(81,170)	(59,166)
Byte	12	13	14	15
Coordinate	(66,160)	(64,182)	(88,178)	(89,177)

E. OPERATING POIS

The above mentioned side-channel attack involves 16 labels, corresponding to byte-0 to byte-15. In the chip design, it was found that each byte has one most important position, called the operating POI. If, during machine learning training, the operating POI has a large weight or is regarded as an important feature, it is often possible to successfully break that byte. Table 1 shows the positions of the 16 operating POIs in this study. Figure 27 shows the corresponding positions of the 16 operating POIs on the temperature map.

F. DISCUSSION ON DATA REALISM AND SIMULATION FIDELITY

Although this study utilizes simulation-generated power and thermal maps for side-channel analysis, it is important to discuss the fidelity of these simulated datasets compared to real-world measurements. Simulations, such as those performed by ANSYS RedHawk-SC, allow for controlled data generation and rapid experimentation, which are essential for developing and testing new methodologies. However, simulated traces may not fully capture all sources of noise, process variation, or unexpected leakages observed in practical hardware environments.

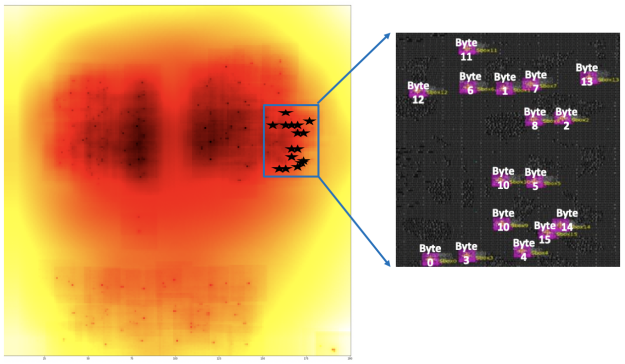


FIGURE 27. Spatial Distribution of 16 Operating Points of Interest (POIs) on the Temperature Map During AES-128 Encryption

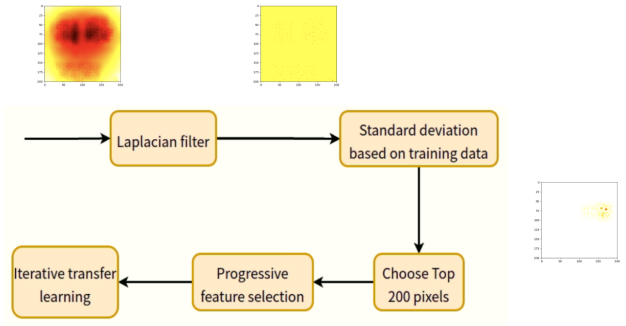


FIGURE 28. Flowchart of Side-Channel Attack

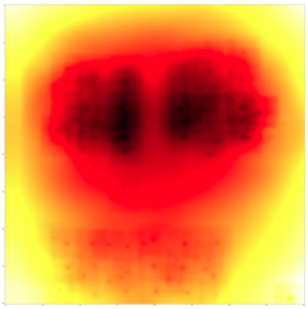


FIGURE 29. Original Temperature Map

As a result, models trained solely on simulated data may achieve higher performance during evaluation but could be less robust when applied to real measurement data. Therefore, while simulation fidelity has been improved by considering both near-field and far-field thermal effects, and by using high-resolution power estimations, validating new techniques on real hardware measurements remains an important direction for future research. This also underlines the necessity for the development of public real-world side-channel datasets to further bridge the gap between simulation and practical deployment.

IV. RESEARCH METHODOLOGY

This section will introduce the iterative transfer learning and progressive feature selection proposed in this study, and provide a comprehensive overview of the preprocessing used in the side-channel attack experiments.

A. OVERVIEW

This study proposes an iterative transfer learning method applied to deep learning models, convolutional neural networks, and multilayer perceptrons for side-channel attacks. This method effectively reduces the amount of data required for training and accelerates the convergence time. Figure 28 shows the flowchart for performing a side-channel attack in this research. First, the Laplacian filter and standard deviation are used to quickly identify the important features in the initial phase. Then, feature selection is applied to reduce the number of features. Finally, iterative transfer learning is used to perform the side-channel attack.

B. DATA PREPROCESSING

In section III, the dataset used in this research is introduced. The dataset consists of 201*201 pixel images. If all pixels are used as features for classifiers like MLP, SVM, random forest, or logistic regression, the training time becomes very long. Having too many features also makes it difficult for the model to classify effectively. Based on the chip design, the features that leak encryption (i.e., high-importance features) are concentrated in a small number of pixels. Therefore, preprocessing is necessary before using the aforementioned classifiers to reduce the number of features. [8] used a Laplacian filter and standard deviation for data preprocessing. The temperature variation data, compared to power consumption data, exhibits thermal junction coupling effects, meaning heat energy diffuses through the medium. As a result, for the temperature variation data, the Laplacian filter is applied to perform edge detection, which eliminates the influence of thermal junction coupling effects and preserves important features. Figure 30 demonstrates that after applying the Laplacian filter, the thermal junction coupling effect is removed, leaving behind the critical features. In [8], different filters were compared, including Sobel filter, Prewitt filter, and Laplacian filter, with the Laplacian filter performing the best.

The second step of data preprocessing is to calculate the standard deviation of each pixel in the training data. In statistical terms, the standard deviation represents the degree of dispersion of the values. A higher degree of dispersion indicates that the operations performed on that pixel are more diverse, making it more likely to be an important feature point. In this study, the top 200 pixels with the highest standard deviations are selected as the initial features. As shown in Figure 31, most of the top 200 pixels are concentrated in the upper right part of the chip, and the blue box in the figure overlaps with the operating POIs, which contain the most leakage information.

FIGURE 30. Laplacian Filter Applied Processing Effect

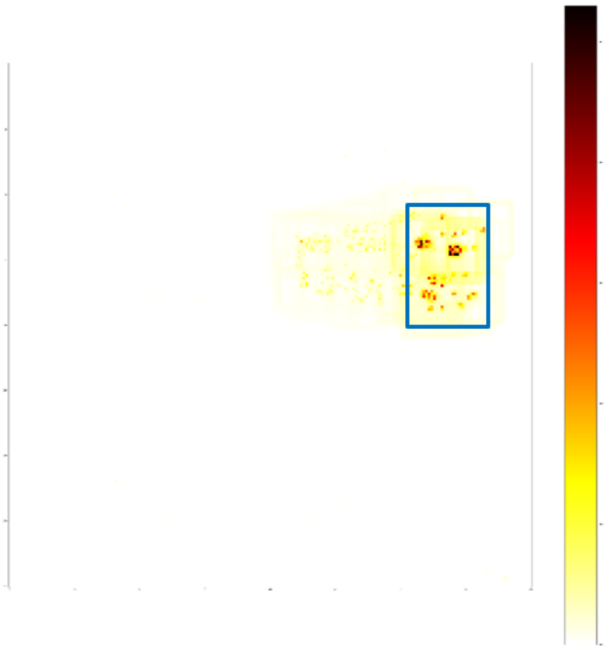


FIGURE 31. Standard Deviation Mapping Diagram [8]

Through the use of the Laplacian filter and standard deviation, the input features can be quickly reduced from 400,000 to 200, making it possible for subsequent feature selection to proceed within a reasonable range. The data preprocessing described above applies only to the temperature variation dataset and classifiers like multilayer perceptron, support vector machine, random forest, and logistic regression. The power consumption dataset does not apply the Laplacian filter because the purpose of the filter is to eliminate the coupling effect, which is not present in the power consumption data. For convolutional neural networks (CNNs), no data preprocessing is applied, and the original data is used as input.

This design choice reflects the differing strengths of traditional machine learning models and CNNs. Models like MLP and SVM are not well suited for capturing spatial dependencies in high-dimensional image data; directly using raw pixel inputs can lead to high computational cost, overfitting, and reduced accuracy. Preprocessing steps such as

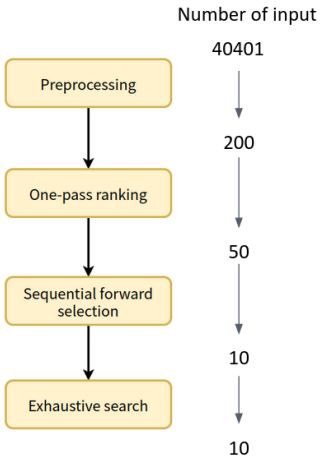


FIGURE 32. Progressive feature selection schematic diagram

Laplacian filtering and feature ranking are therefore essential to reduce dimensionality and retain only the most informative attributes.

In contrast, CNNs are inherently capable of learning local and hierarchical features from raw data through convolutional layers. Preprocessing techniques like edge detection or pixel selection may disrupt important spatial patterns, undermining CNN performance. Using raw inputs allows CNNs to fully exploit spatial structure, albeit with increased computational demands and potential sensitivity to noise. Each approach is thus aligned with the model's architecture and the characteristics of the input data.

C. PROGRESSIVE FEATURE SELECTION

This paper explores a novel approach that combines three feature selection methods, referred to as progressive feature selection, as illustrated in Figure 32. Initially, STD and the Laplacian filter are applied for preprocessing, enabling rapid feature selection. Subsequently, one-pass ranking, Sequential Forward Selection (SFS), and exhaustive search are used in sequence to gradually reduce the number of features from 40,401 to 10. Since SFS and exhaustive search typically yield better feature selection results but require longer computation time, this method first employs fast feature selection when dealing with a large number of features, followed by the more effective SFS and exhaustive search, achieving a balance between computational efficiency and selection quality.

D. ITERATIVE TRANSFER LEARNING

Currently, most studies on machine learning-based AES-128 attacks typically model the S-box output. Specifically, each byte is trained as an independent model without considering correlations between different bytes. The main issue with this approach is that, although the training process for each byte is similar, the critical features for different bytes are not entirely the same. As a result, training separate models for each byte independently may lead to inefficient resource utilization and

fail to exploit the potential inter-byte relationships.

To enhance training efficiency and reduce the computational cost of model training, this study introduces Transfer Learning. Transfer learning is a widely used technique in machine learning, based on the idea that when a new task shares a similar data distribution with a pre-trained model, the learned model weights can be leveraged to achieve faster convergence on the new task. Typically, transfer learning involves pre-training a model on a large dataset and then fine-tuning it on a smaller dataset. However, in the context of AES-128 attacks, relying solely on traditional transfer learning with limited pre-training data may still fail to successfully break the encryption.

To address this issue, this study adopts a Progressive Training approach to enhance the attack performance. Specifically, during training, the models for different bytes are not trained independently. Instead, the trained model from the previous byte is used as the pre-trained model for the next byte. This approach follows a recursive training flow, where each byte benefits from the learning experience of the preceding byte, further optimizing the training process. By iterating this process, the training follows a cyclic pattern, as illustrated in Figure 33.

One key advantage of this training method is that even with a limited dataset, deep learning models can still converge and successfully break AES-128. According to the experimental results, after two rounds of progressive training, the Minimum Trace Delay (MTD) for each byte stabilizes, indicating that the model has reached a steady state. Furthermore, although the order in which bytes are trained may have a slight impact on the final results, this progressive training strategy effectively improves model performance and reduces reliance on large-scale training datasets.

In this study, the byte-wise training sequence was fixed from Byte-0 to Byte-15. This order follows the natural layout of AES S-box processing and assumes relative independence among byte operations. While we did not conduct experiments on alternative orders, the observed convergence stability and consistent MTD results across different data sizes suggest that the proposed ITL method is not highly sensitive to training sequence.

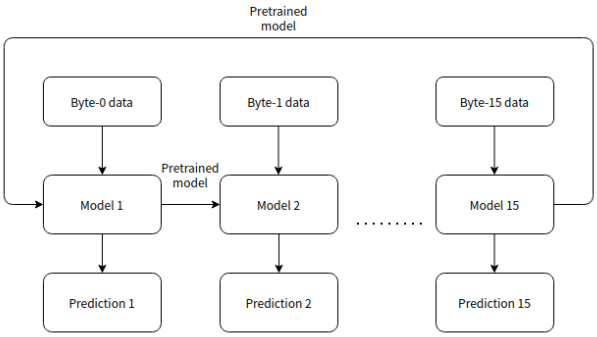


FIGURE 33. Iterative transfer learning illustration

V. EXPERIMENTAL DESIGN AND RESULTS DISCUSSION

A. EXPERIMENTAL ENVIRONMENT

The training and testing environment for this experiment is Ubuntu 18.04, equipped with one GPU (NVIDIA GeForce RTX 2070 SUPER) and one CPU (Intel Core i5-9400F). The program is written in Python 3.8, and the machine learning and deep learning models are implemented using PyTorch.

B. EXPERIMENTAL PROCESS

The experiments in this study are divided into three parts. The first part explores the impact of different models, loss functions, and preprocessing methods on AES-128 attack results under the condition of sufficient training data. The second part investigates the performance of models trained with different amounts of training data using the same settings. The third part examines the performance of models trained with different types of data. For the first two parts, temperature variation data are used as the training data. The training process is shown in Figure 28 from the previous Section, and the experimental design is as follows.

- Experiment 1: Comparison of AES-128 attack results with different model settings. This experiment includes the following details:
  - Experiment 1.1: Comparison of AES-128 attack results with different classification models.
  - Experiment 1.2: Impact of different loss functions on Multi-Layer Perceptron (MLP) and Convolutional Neural Networks (CNNs).
  - Experiment 1.3: Comparison of AES-128 attack results with different feature selection and feature extraction methods in classification models.
- Experiment 2: Investigating the performance of different models trained with various methods in attacking AES-128 under reduced training data conditions, comparing results using 2,000, 5,000, 8,000, and 13,600 training samples.
- Experiment 3: Compare the performance of various models trained on different datasets using different methods, including the temperature variation dataset and the power consumption dataset.

C. CLASSIFIER PARAMETER SETTINGS

1) Random Forest

In the Random Forest study, the experiment is conducted using sklearn [21]. A total of 100 decision trees are used, with entropy as the criterion for evaluating the splits. There is no limitation on the depth of the trees, and the branching continues until the number of samples in each node is less than 2.

2) Logistic Regression

In the logistic regression study, the experiment was assisted by sklearn [21]. To prevent overfitting during training, an L2 penalty was used, and early stopping was em-

TABLE 2. Multilayer Perceptron Architecture

Type	Output Shape	Param #
Linear	bs * 20	4020
BatchNorm1d	bs * 20	40
Mish	bs * 20	0
Dropout	bs * 20	0
Linear	bs * 20	420
BatchNorm1d	bs * 20	40
Mish	bs * 20	0
Dropout	bs * 20	0
Linear	bs * 256	5376

ployed. The solver parameter was set to limited-memory Broyden–Fletcher–Goldfarb–Shanno (lbfgs).

3) Support Vector Machine

In the support vector machine (SVM) study, the experiment was assisted by sklearn [21]. The regularization parameter was set to 1, and a linear kernel function was used to project the plane.

4) Multi-Layer Perceptron (MLP)

The architecture of the Multi-Layer Perceptron (MLP) is shown in Table 2, consisting of two hidden layers, each with 20 neurons. The activation function used is Mish, and dropout is applied to prevent overfitting, with a dropout rate of 0.5, meaning half of the neuron weights are randomly discarded during training. The batch size during training is 256, and the model is trained for 500 epochs using the Ranger optimizer. The parameters β_1 = 0.95, β_2 = 0.999 are used, and weight decay is set to 0.1. The loss function is set to cross-entropy. The training will stop based on changes in the loss function. If the loss function does not decrease after 200 epochs, the training will be stopped early, even if the maximum number of epochs has not been reached.

5) Convolutional Neural Network (CNN)

The architecture of the Convolutional Neural Network (CNN) is shown in Table 3. It consists of three convolutional layers with a kernel size of 3 and a stride of 2, using the Mish activation function. The pooling layer uses average pooling (AvgPooling). Dropout is also used to prevent overfitting, but only in the fully connected layers, with a dropout rate of 0.5. During training, the batch size is set to 128, and training is carried out for 500 epochs. The optimizer used is Ranger, with $\beta_1 = 0.95$ and $\beta_2 = 0.999$, and weight decay is set to 0.1. The loss function is cross-entropy. The criteria for stopping training are the same as for the multilayer perceptron.

D. EVALUATION METHOD

1) Ranking Function

In machine learning, accuracy is commonly used to evaluate classification problems. However, in side-channel attacks, due to the large number of classification categories, the predicted probability values are not very prominent, and the accuracy is usually low, making it difficult to assess the

TABLE 3. Convolutional Neural Network Architecture

Type	Output Shape	Param #
Conv2d	bs * 64 * 101 * 101	640
BatchNorm2d	bs * 64 * 101 * 101	128
Mish	bs * 64 * 101 * 101	0
Avgpool	bs * 64 * 50 * 50	0
Conv2d	bs * 32 * 25 * 25	18464
BatchNorm2d	bs * 32 * 25 * 25	64
Mish	bs * 32 * 25 * 25	0
Avgpool	bs * 32 * 12 * 12	0
Conv2d	bs * 16 * 6 * 6	4624
BatchNorm2d	bs * 16 * 6 * 6	32
Mish	bs * 16 * 6 * 6	0
Avgpool	bs * 16 * 3 * 3	0
Linear	bs * 20	9280
BatchNorm1d	bs * 20	128
Mish	bs * 64	0
Dropout	bs * 64	0
Linear	bs * 64	4160
BatchNorm1d	bs * 64	128
Mish	bs * 64	0
Dropout	bs * 20	0
Linear	bs * 256	16640

model's performance with a single data point. Therefore, the evaluation is done by aggregating the probabilities.

First, the S-box predicted label (0-255) is reverse-engineered using the plaintext of this data to predict the key. The probability of the predicted key corresponds to the probability of the S-box prediction. The key prediction probabilities are then log-sum and ranked. The rank of the real key is the output of the ranking function. The range of the ranking function is from 0 to 255, where 0 is the best and indicates the successful decryption of the AES-128 byte. If, after aggregating all the data, the ranking function is not 0, it indicates that the byte cannot be successfully decrypted.

2) Measurement-to-disclosure (MTD)

MTD (Minimum Trace Delay) represents the number of data points required during the attack to stabilize the predicted key such that the ranking function equals 0. During the attack, the probability of each data point is log-summed sequentially, and the ranking function is calculated after each step, until all the attack data (in this experiment, 3,000 data points) have been processed. If the ranking function remains at 0 after 500 data points, then the MTD is 500. If the ranking function remains at 0 after 2,000 data points, the MTD is 2,000. Figure 34 shows the changes in the ranking function when using Multi-Layer Perceptron (MLP) and Support Vector Machine (SVM) on Byte-12. It is found that after 58 data points, the MLP's ranking function remains at 0, indicating an MTD of 58. For the SVM, the ranking function becomes 0 after 2,306 data points but then fluctuates back to 1, stabilizing at 0 only after 2,523 data points, so the MTD for SVM is 2,523.

In side-channel attacks, there are 16 bytes that need to be decrypted. AES-128 is considered successfully decrypted only when the MTD of all 16 bytes is smaller than the number of data points in the attack phase. Therefore, when evaluating the model's performance, the worst MTD is typically used for

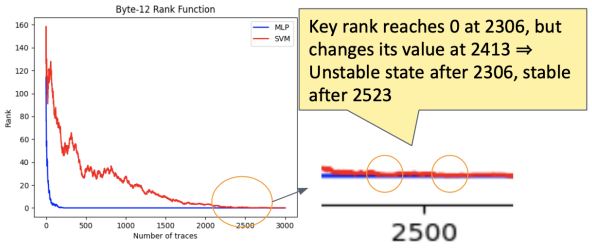


FIGURE 34. Byte-12 Ranking Function Changes

description. In this experiment, both the maximum MTD and the average MTD are used to assess the attack performance. If AES-128 cannot be successfully cracked, the average MTD will not be displayed.

E. EXPERIMENT 1: AES-128 ATTACK RESULTS WITH DIFFERENT MODEL SETTINGS

The purpose of this experiment is to compare the results of various models attacking AES-128, including Correlation Energy Analysis, Random Forest, Support Vector Machine, Logistic Regression, Multi-Layer Perceptron, and Convolutional Neural Networks. Additionally, the experiment observes the impact of different loss functions on the deep learning models, Multi-Layer Perceptron and Convolutional Neural Networks, as well as the effect of different feature selection methods on these models.

1) Experiment 1.1: Comparison of Different Classification Models Attacking AES-128

a: Experiment Setup

The training data used in this experiment consists of 13,600 temperature change images, which provides sufficient training data. Classifiers such as multilayer perceptrons (MLP), logistic regression, random forests, and support vector machines only undergo preprocessing and do not perform feature selection, so the number of features is 200. The proposed MLP and proposed CNN use the iterative transfer learning training method. In the calculation of MTD, since the order of the probability data can also affect the MTD results, this experiment randomly shuffles the order 100 times, and the average MTD is used as the result.

The parameter settings for the various models used in this experiment have been detailed in Section V-C.

b: Experimental Results and Analysis

Table 4 presents the experimental results. Most of the attack models successfully broke AES-128, with the exception of the random forest, which failed to break the encryption. A failure to break is defined as having at least one byte that could not be successfully cracked. The reason for this failure is that the diffusion of temperature causes the feature values

TABLE 4. Results of Attacking AES-128 with Different Classification Models

Attack Model	Average MTD ↓	Worst MTD ↓	Average Rank ↓
CPA	1402	2750	0
CNN	125	233	0
MLP	55	111	0
Logistic	334	937	0
Random Forest	x	>3000	46
SVM	647	2990	0
Proposed MLP	**54**	**88**	0
Proposed CNN	83	164	0

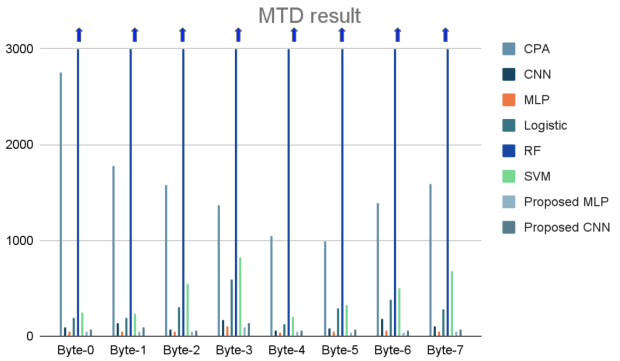


FIGURE 35. The MTD of different classifiers for byte-0 to byte-7, with arrows indicating that the MTD did not converge within 3,000

to have insignificant differences, making it more difficult to judge based on branching alone.

In comparison, deep learning models, such as multilayer perceptrons (MLP) and convolutional neural networks (CNN), showed significant improvement over traditional machine learning models. Despite their complex architectures and weights, these deep learning models were still able to learn important features from the temperature dataset, thereby improving the model performance.

Figures 35 and 36 show the MTD for each classifier across every byte. The arrows at the top of the figures represent an MTD exceeding 3,000, indicating that the byte could not be successfully cracked. In the case of the random forest, MTD was only successful for byte-14, while the rest of the bytes could not be cracked.

The proposed iterative transfer learning applied to MLP and CNN showed improvements across the results. As seen in the table and figures, the iterative transfer learning approach achieved better performance in all classifiers. The MLP model showed a smaller progression in average MTD, possibly because, under the experimental conditions, the MLP was already performing sufficiently well, and most bytes showed limited improvement. However, there was still a noticeable improvement in the worst-case MTD.

2) Experiment 1.2: Investigating the Impact of Different Loss Functions on Multi-Layer Perceptrons and Convolutional Neural Networks

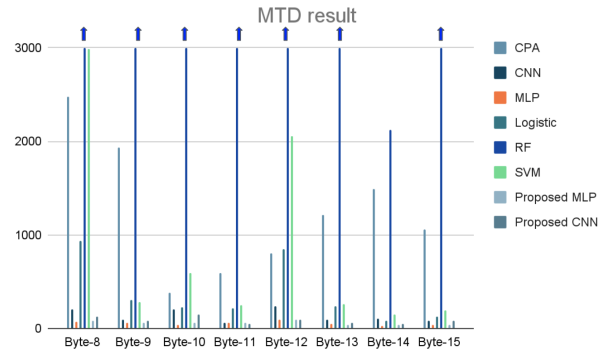


FIGURE 36. The MTD of different classifiers for byte-8 to byte-15, with arrows indicating that the MTD did not converge within 3,000

a: Experiment Setup

The loss functions and their settings for this experiment are described in detail in Section II-D. The model and other parameter settings for this experiment are the same as those in Experiment 1.1.

b: Experimental Results and Analysis

Based on Tables 5 and 6, it can be observed that the cross entropy ratio fails to successfully break AES-128. However, from Figures 37–40, by examining the MTD for each byte, it is found that when combined with the multilayer perceptron, about half of the bytes can still be successfully cracked, such as byte-0, byte-1, and byte-2, indicating that the cross entropy ratio is capable of breaking AES-128.

As mentioned in [22], the denominator when training with cross entropy ratio presents a challenge. This denominator is the cross entropy between the predicted probability and the negative labels, which distinguishes the distribution of negative labels. In classification problems, there might be data that is easy to classify or difficult to classify. Data that is easy to classify will significantly improve the calculation of the denominator, thereby reducing the overall cross entropy ratio loss.

Furthermore, in [13], it is stated that the cross entropy ratio shows notable improvement with imbalanced training data. However, in this experiment, where uniformly labeled training data is used, the MTD for cross entropy ratio does not perform exceptionally well.

Ranking loss, which is calculated differently from cross entropy and cross entropy ratio, takes into account MTD and calculates the loss based on ranking. This is a loss function adjusted for side-channel attacks (SCA). However, in this experiment's dataset, it did not outperform cross entropy. Based on the current experimental results, no significant difference between cross entropy and ranking loss can be observed.

In conclusion, for this experiment, comparing different loss functions combined with multilayer perceptron and convolutional neural networks, cross entropy remains the best loss function.

TABLE 5. Results of Different Loss Functions in Multi-Layer Perceptron

Loss Function	Average MTD ↓	Worst MTD ↓	Average Rank ↓
Cross entropy	54	88	0
Cross entropy ratio	x	>3000	4
Ranking loss	61	123	0

TABLE 6. Results of Different Loss Functions in Convolutional Neural Network

Loss Function	Average MTD ↓	Worst MTD ↓	Average Rank ↓
Cross entropy	125	233	0
Cross entropy ratio	x	>3000	89
Ranking loss	195	610	0

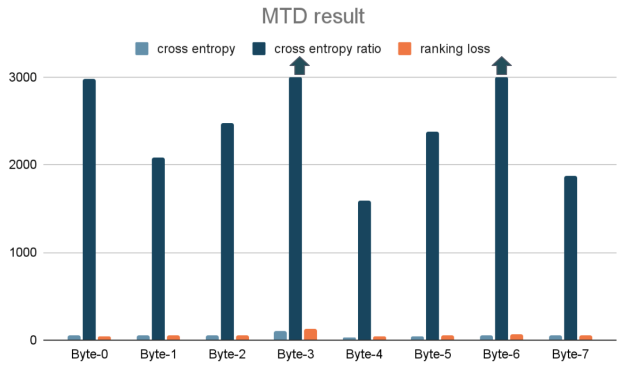


FIGURE 37. MTD for Byte-0 to Byte-7 with MLP combined with different loss functions

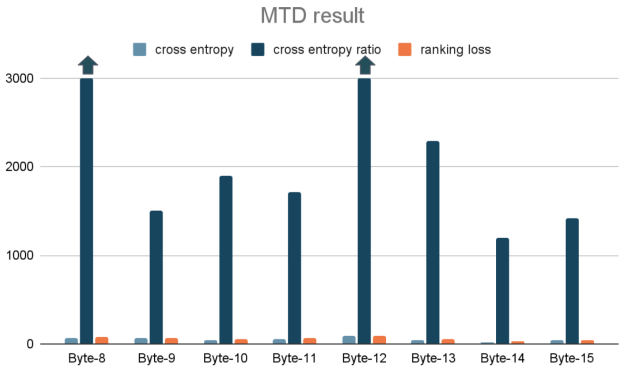


FIGURE 38. MTD for Byte-8 to Byte-15 with MLP combined with different loss functions

3) Experiment 1.3: Exploring the Impact of Different Feature Selection and Feature Extraction on Different Classifiers

a: Experiment Setup

Table 7 presents the parameter settings for feature selection and feature extraction. The feature selection process evaluates features using logistic regression, which is chosen due to its lower computational cost. All feature selection and feature extraction methods undergo preprocessing using standard deviation normalization, resulting in an initial feature count of 200.

Progressive feature selection employs a series of feature

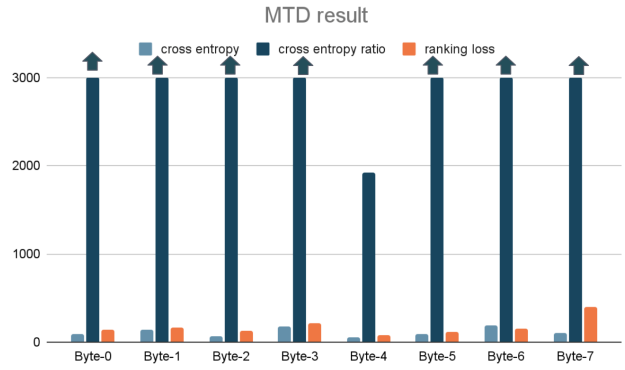


FIGURE 39. MTD for Byte-0 to Byte-7 with CNN combined with different loss functions

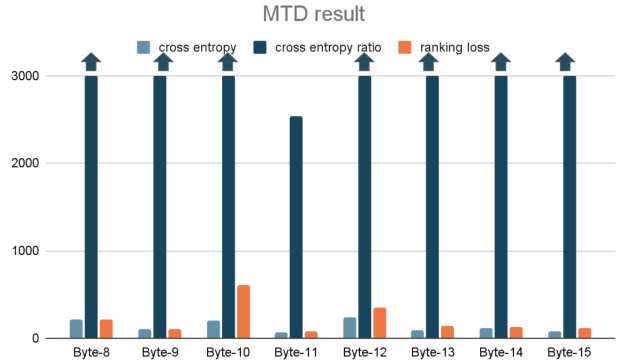


FIGURE 40. MTD for Byte-8 to Byte-15 with CNN combined with different loss functions

TABLE 7. Parameters of Feature Selection and Feature Extraction

Method	Feature Selection Model	Num of Input	Num of Output
One-pass ranking	Logistic regression	200	10, 50
Sequential feature selection (SFS)	Logistic regression	200	10
Progressive feature selection	Logistic regression	40401	10
Principal component analysis (PCA)	–	200	10
Linear discriminant analysis (LDA)	–	200	4,5,6

selection steps, with the feature count in the table representing the total initial feature count of the input image. The dimensionality reduction for LDA is determined based on the MTD of the validation dataset. Specifically, the dimensionality is set to 4 for logistic regression and support vector machines, 5 for multilayer perceptron (MLP) and the proposed MLP, and 6 for random forest.

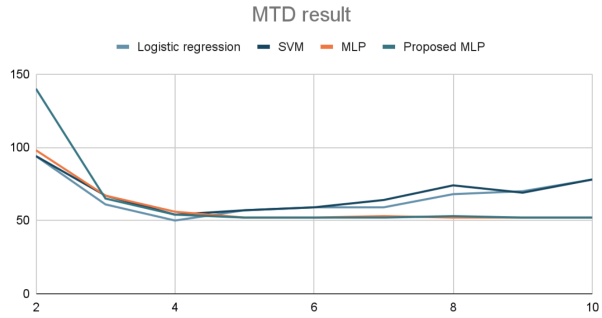


FIGURE 41. Comparative Analysis of Classification Performance: Logistic Regression vs. SVM vs. MLP vs. Proposed MLP Across LDA Dimensions

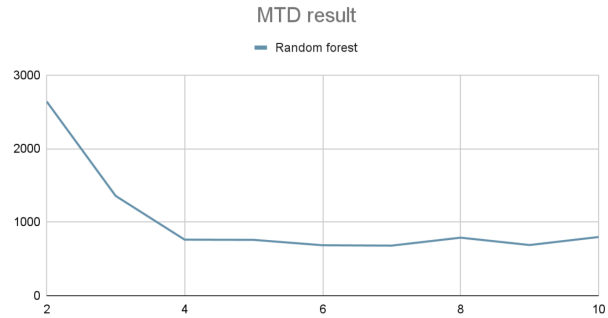


FIGURE 42. Dimensionality Impact Analysis: Random Forest MTD Variation on Validation Data with LDA Projection

b: Experimental Results and Analysis

Figure 41 and 42 shows the average MTD results of different classifiers under various LDA dimensionality reductions on the validation dataset. It can be observed that the lowest MTD for Logistic Regression and Support Vector Machine (SVM) occurs at a dimensionality of 4, while Multi-Layer Perceptron (MLP) and the proposed MLP achieve the lowest MTD at a dimensionality of 5. For Random Forest, the lowest MTD is observed at a dimensionality of 6. Based on these results, the dimensionality reduction for LDA will be adjusted accordingly for each classifier during the attack.

From the experimental results, it is evident that Logistic Regression and Support Vector Machine (SVM) achieve a significant reduction in MTD when using one-pass ranking, SFS, and progressive feature selection. As shown in Table 8, Random Forest does not successfully break the encryption through feature selection. Since Multi-Layer Perceptron (MLP) and the proposed MLP already have relatively low MTD, the improvement is limited, and no significant reduction in MTD is observed after feature selection. However, feature selection effectively reduces the feature space, which in turn reduces the time spent on training.

Furthermore, the results of the proposed progressive feature selection in this paper are consistent with those of SFS. Since the feature space is reduced to 10 after SFS, the benefit of exhaustive search is limited. A possible solution could be to increase the number of features searched by exhaustive

TABLE 8. Performance Evaluation of Feature Engineering Methods in Random Forest Classification

Feature Engineering Method	MTD_{avg} (ms)	MTD_{max} (ms)	$Rank_{avg}$
Raw Features (200D)	N/C	>3,000	46
Univariate Filter (200D→10D)	N/C	>3,000	7
Sequential Forward Selection (200D→10D)	N/C	>3,000	6
Progressive Feature Selection (200D→50D→10D)	N/C	>3,000	6
PCA Projection (200D→10D)	N/C	>3,000	156
LDA Projection (200D→6D)	644	1,600	0

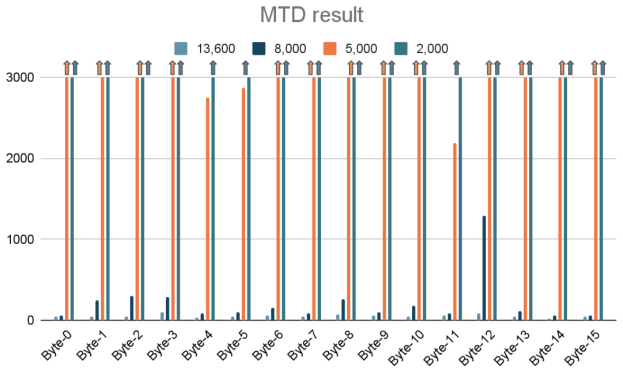


FIGURE 43. MTD of MLP for Individual Bytes with Varying Training Data Sizes, Arrows Indicate MTD Failing to Converge Within 3,000

search, but this would also increase the computational time.

Regarding feature extraction, applying PCA causes all classifiers to fail in breaking the encryption. As an unsupervised method, PCA does not utilize label information during dimensionality reduction, which may result in the loss of key-dependent features critical for successful decryption. In contrast, supervised methods like LDA leverage class labels to optimize class separability, preserving important distinctions in the data. By projecting the data into a low-dimensional space while maintaining discriminative structure, LDA enables successful decryption—most notably improving the performance of Random Forest, which initially failed to break the encryption.

In summary, LDA is the best method for both feature selection and feature extraction. It effectively reduces the feature space, provides faster computation times, and allows the model to successfully break the encryption.

F. EXPERIMENT 2: EVALUATING THE PERFORMANCE OF DIFFERENT MODELS IN ITERATIVE TRANSFER LEARNING ATTACKS ON AES-128 USING LIMITED TRAINING DATA AND VARIOUS METHODS.

1) Experiment Setup

This experiment employs Multi-Layer Perceptron (MLP) and Convolutional Neural Network (CNN) deep learning models to investigate the effectiveness of iterative transfer learning in attacking AES-128 under limited data conditions and compare it with non-iterative transfer learning methods. The training data sizes are 2,000, 5,000, 8,000, and 13,600, with validation data set at one-fourth of the training data (500, 1,250, 2,000, and 3,400, respectively), and the attack data fixed at 3,000.

The loss function used is cross-entropy, with no feature selection or extraction applied. If an attack requires more than 3,000 data points, it is considered unsuccessful, denoted as ">3000" in the table.

2) Experimental Results and Analysis

Figures 43 to 46 illustrate the trends of different classifiers under varying amounts of training data. It is evident that as the amount of training data decreases, the required MTD continuously increases. Deep learning models without iterative transfer learning fail to successfully break AES-128 when trained with only 5,000 samples. Notably, CNN is unable to crack byte-2 even with 8,000 training samples, highlighting the critical importance of training data volume.

In contrast, deep learning models utilizing iterative transfer learning can successfully break AES-128 with only 5,000 training samples, with both MLP and CNN achieving successful attacks. Specifically, the proposed MLP maintains an MTD below 2,000 even with only 2,000 training samples. Although the proposed CNN fails to crack AES-128 with 2,000 training samples, it succeeds when provided with 5,000 samples. This demonstrates that iterative transfer learning is beneficial when training data is insufficient.

When training data is sufficient, the performance difference between MLP and the proposed MLP is minimal, with MLP even outperforming the proposed MLP in some cases. However, as the amount of data decreases, the proposed MLP significantly outperforms the standard MLP.

These results indicate that when the original deep learning model has a low MTD, the benefits of iterative transfer learning are limited. However, when the MTD is high or when certain bytes cannot be cracked, iterative transfer learning can significantly improve MTD performance. This experiment validates that the proposed iterative transfer learning method effectively breaks AES-128 even with reduced training data, leading to notable improvements in MTD.

Figure 47 illustrates the separately trained MLP, which corresponds to the standard MLP used in this experiment. It can be observed that the initial values of the loss function for byte-0, byte-1, and byte-2 are similar, all starting at approximately 5.66 before decreasing.

Figure 48 shows the loss variation when applying iterative transfer learning to train the MLP. It is evident that the initial loss values for byte-1 and byte-2 are lower than in the standard MLP. This is because the model pre-trained on byte-0 serves as the initialization for subsequent bytes, providing better initial weights that enhance performance and accelerate convergence.

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2025.3570763

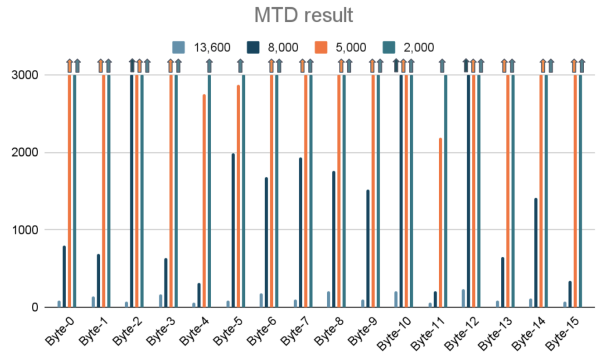


FIGURE 44. MTD of CNN for Individual Bytes with Varying Training Data Sizes, Arrows Indicate MTD Failing to Converge Within 3,000

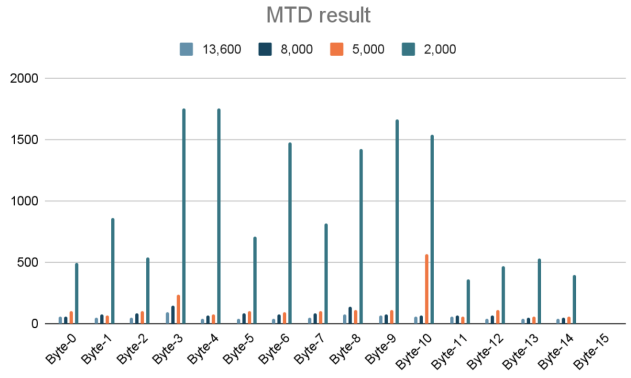


FIGURE 45. MTD of Proposed MLP for Individual Bytes with Varying Training Data Sizes, Arrows Indicate MTD Failing to Converge Within 3,000

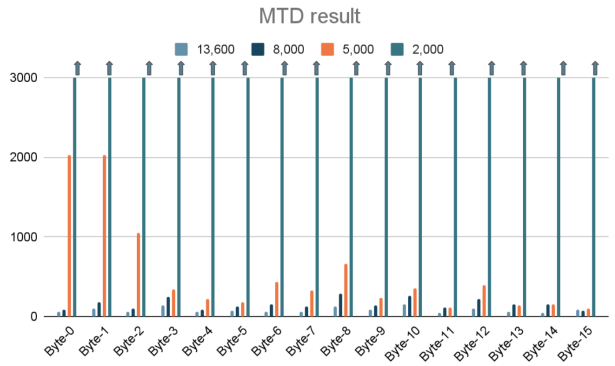


FIGURE 46. MTD of Proposed CNN for Individual Bytes with Varying Training Data Sizes, Arrows Indicate MTD Failing to Converge Within 3,000

Figure 49 presents the changes in MTD across different rounds. Each round consists of training from byte-0 to byte-15 once. In this figure, training is conducted using 2,000 samples over four rounds, meaning each byte is trained four times. The results indicate that as the number of rounds increases, the MTD gradually converges.

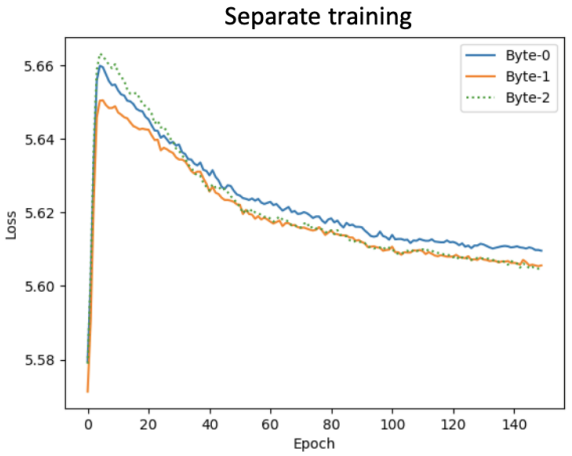


FIGURE 47. Loss Function Variation of MLP Under Separate Training Strategy

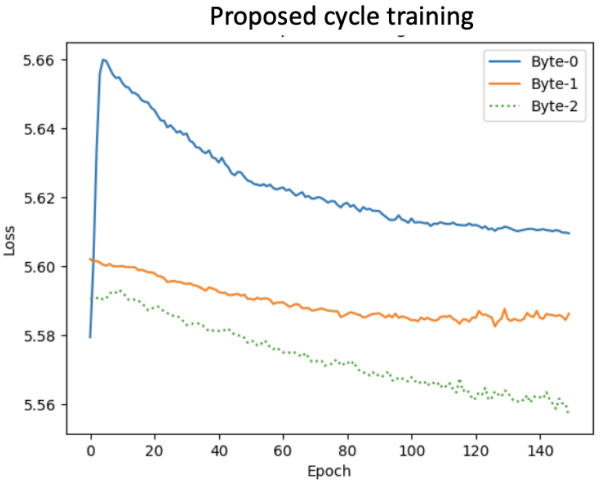


FIGURE 48. Loss Function Variation of MLP Under Iterative Transfer Learning Strategy

G. EXPERIMENT 3: COMPARE THE PERFORMANCE OF VARIOUS MODELS TRAINED WITH DIFFERENT METHODS ON DIFFERENT DATASETS.

1) Experiment Setup

In this experiment, we compared the performance of the models mentioned earlier across different datasets. The datasets used were the power dataset and the temperature dataset. Detailed information about the datasets is provided in section III. All other settings remained the same as in the previous experiments.

2) Experimental Results and Analysis

Table 9 presents the average MTD (Minimum Test Data) of various models across different datasets. All models successfully break AES-128 when trained on the power dataset.

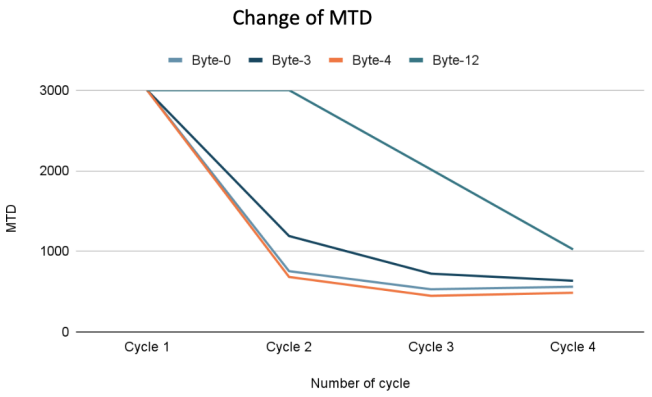


FIGURE 49. MTD Variations Across Training Cycles

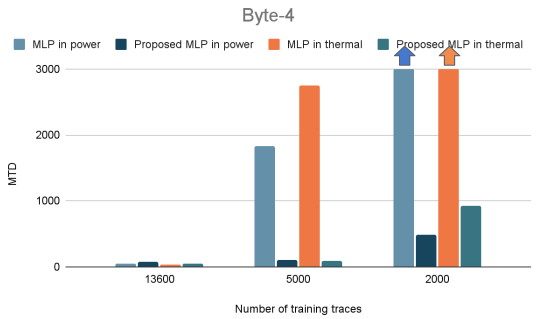


FIGURE 50. MTD Comparison Between MLP and Proposed MLP Across Multiple Datasets with Varying Training Data Sizes at Byte-4 (Arrows Indicate Non-Convergence Within 3,000)

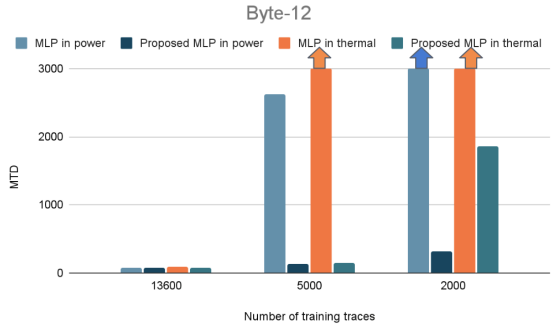


FIGURE 51. MTD Comparison Between MLP and Proposed MLP Across Multiple Datasets with Varying Training Data Sizes at Byte-12 (Arrows Indicate Non-Convergence Within 3,000)

Notably, there is a clear improvement in both the correlation energy analysis and machine learning models. The absence of the coupling effect in the power data facilitates the model's ability to learn the features more effectively.

In particular, the Random Forest model, which fails to break the temperature dataset, demonstrates a significant decrease in average MTD when trained on the power dataset, with the value dropping below 1037. This suggests that the coupling effect, which affects temperature data, can complicate the feature extraction process.

Moreover, the experiment reveals that the MTD of the Convolutional Neural Network (CNN) is higher when trained on the power dataset compared to the temperature dataset. This can be attributed to the lack of coupling effect in the power data, which results in the concentration of significant features within a single pixel. In contrast, temperature data involves the surrounding pixels of operating Points of Interest (POIs), which, influenced by the coupling effect, exhibit similar features. The CNN is designed to focus on specific regions, which makes it more challenging to concentrate on individual pixels, thereby leading to poorer performance when trained on power data.

The Multi-Layer Perceptron (MLP) and Iterative Transfer Learning (ITL) models have similar MTD values for both power and temperature datasets. However, convergence is faster with the power dataset. This means that using power data has advantages, especially when less training data is available.

Figures 50 and 51 display the MTD for Multi-Layer Perceptron (MLP) and the proposed MLP models on both power and temperature datasets for byte-4 and byte-12. When the data volume is reduced, training on the power dataset results in better MTD performance. However, when only 2,000 data points remain, the power dataset still fails to successfully break the byte, whereas Iterative Transfer Learning (ITL) enables successful cracking. Similar trends are observed for the other bytes.

Based on the experiment, Iterative Transfer Learning proves to be effective across various datasets. The power dataset generally yields better MTD results compared to the temperature dataset, but the gap between the datasets is reduced when ITL and MLP are applied.

TABLE 9. Average MTD Across Different Datasets

Attack Model	Power	Thermal
CPA	423	1368
CNN	204	125
MLP	54	59
Logistic Regression	313	334
Random Forest	1037	>3000
SVM	111	261
Proposed MLP	52	53
Proposed CNN	56	82

VI. CONCLUSION AND FUTURE WORK

A. CONCLUSION

This study presents an innovative deep learning model training method—Iterative Transfer Learning, aimed at enhancing the interrelationship between different byte models by gradually training models across multiple bytes, thus improving the overall model performance. In this method, each trained model serves as the pre-trained model for the next byte,

forming a continuous iterative training process. Experimental results demonstrate a clear advantage of traditional machine learning and deep learning models over correlation-based energy analysis, with deep learning models outperforming machine learning models. Furthermore, when sufficient data is available, combining progressive transfer learning with multi-layer perceptrons (MLP) further improves decryption performance.

We tested different loss functions and found that cross-entropy gave the best results in every case, both in average and maximum MTD. Feature selection and extraction also helped improve model performance when the right methods were chosen.

In the case of insufficient data, progressive transfer learning demonstrated its critical role in enhancing the performance of deep learning models. The experiments revealed that traditional deep learning models failed to successfully break the AES-128 encryption with insufficient data, but by introducing progressive transfer learning, the model's MTD significantly improved and successfully decrypted AES-128. This indicates that progressive transfer learning provides new initial weights for the bytes, allowing the model to learn effectively and solve complex problems with limited data.

Through experimental analysis of power consumption and temperature variation data, it was observed that the leakage of information from CPA and traditional machine learning models (such as logistic regression, random forests, and support vector machines) became more apparent. In deep learning models, both power consumption and temperature change data exhibited similar MTD results, further demonstrating the strong learning capability of deep learning models. Even when faced with complex data types, they can still extract valuable information.

The experimental results demonstrate that the implemented deep learning models and data processing methods perform exceptionally well across diverse data types. Progressive transfer learning effectively addresses the common challenge of data scarcity in deep learning, significantly expanding the applicability of these methods.

In addition to technical advancements, we acknowledge the ethical implications of developing more powerful side-channel attack methodologies. Such techniques, if misused, could threaten the security of critical infrastructures and personal data. Therefore, we urge researchers and practitioners to use these findings solely for defensive purposes, such as vulnerability assessment and the enhancement of hardware security. Collaborating closely with industry and regulatory bodies to ensure responsible application and prompt mitigation is crucial for minimizing potential ethical risks associated with this technology.

B. FUTURE WORK

In future work, we aim to extend our proposed iterative transfer learning (ITL) method to more complex datasets and diverse physical leakage sources beyond temperature and power consumption, such as electromagnetic (EM) emissions and battery radiation leakage. These modalities often involve stronger coupling effects and higher noise levels, presenting new challenges for generalization and robustness. We also plan to validate the methodology on real-world measurement data, as current simulations do not fully capture hardware variations or environmental unpredictability. This will help assess the practical effectiveness of ITL and identify challenges in transitioning from simulated to physical environments.

While this study did not include evaluation of cross-device transferability between thermal and electrical datasets or between different hardware, existing literature [15] [16] suggests that transfer learning approaches can potentially bridge some domain gaps. In the future, we will investigate the applicability and challenges of extending our models across diverse device types and signal domains.

Additionally, we recognize the increasing difficulty of side-channel attacks due to the growing use of masking and obfuscation techniques. To address these challenges and improve training efficiency, we plan to explore multitask learning approaches that can predict all 16 AES-128 bytes simultaneously using shared input features. This may reduce the need for training separate models per byte and shorten the overall training time.

Another promising direction involves the design of new loss functions tailored to the unique characteristics of side-channel analysis. Unlike conventional classification tasks, side-channel attacks typically involve a large number of classes, and success is measured using metrics like minimum traces to disclosure (MTD) rather than accuracy. Developing specialized loss functions and appropriate pre-/postprocessing techniques may lead to improved model performance and better alignment with side-channel attack objectives.

Finally, we acknowledge the ethical implications of advancing side-channel attack methodologies. These techniques should be applied responsibly, primarily for evaluating vulnerabilities and strengthening hardware security. We encourage collaboration with industry and regulatory bodies to ensure the safe and ethical use of such technologies.

ACKNOWLEDGMENT

Zhang Kai and Tun-Chieh Lou are co-first authors.

REFERENCES

[1] S. Chari, J. R. Rao, and P. Rohatgi, "Template attacks," in *Cryptographic Hardware and Embedded Systems - CHES 2002, 4th International Workshop, Redwood Shores, CA, USA, August 13-15, 2002, Revised Papers*, ser. Lecture Notes in Computer Science, vol. 2523. Springer, 2002, pp. 13–28.

[2] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Advances in Cryptology — CRYPTO' 99*, M. Wiener, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, pp. 388–397.

[3] E. Brier, C. Clavier, and F. Olivier, "Correlation power analysis with a leakage model," in *Cryptographic Hardware and Embedded Systems - CHES 2004*, M. Joye and J.-J. Quisquater, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 16–29.

[4] P. C. Kocher, "Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems," in *Advances in Cryptology — CRYPTO '96*, N. Koblitz, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1996, pp. 104–113.

[5] S. Jin, S. Kim, H. Kim, and S. Hong, "Recent advances in deep learning-based side-channel analysis," *ETRI Journal*, vol. 42, 02 2020.
[6] S. Yang, Y. Zhou, J. Liu, and D. Chen, "Back propagation neural network based leakage characterization for practical security analysis of cryptographic implementations," in *Information Security and Cryptology-ICISC 2011: 14th International Conference, Seoul, Korea, November 30-December 2, 2011. Revised Selected Papers 14*. Springer, 2012, pp. 169–185.
[7] Z. Martinasek and V. Zeman, "Innovative method of the power analysis," *Radioengineering*, vol. 22, pp. 586–594, 06 2013.
[8] C. W. Chen, "Points of interest selection in location based thermal emission side-channel analysis and machine learning based attack model," Master's thesis, National Taiwan University, Jan 2021.
[9] S. Albelwi and A. Mahmood, "A framework for designing the architectures of deep convolutional neural networks," *Entropy*, vol. 19, no. 6, 2017. [Online]. Available: https://www.mdpi.com/1099-4300/19/6/242
[10] H. Maghrebi, T. Portigliatti, and E. Prouff, "Breaking cryptographic implementations using deep learning techniques," in *Security, Privacy, and Applied Cryptography Engineering: 6th International Conference, SPACE 2016, Hyderabad, India, December 14-18, 2016, Proceedings 6*. Springer, 2016, pp. 3–26.
[11] R. Benadjila, E. Prouff, R. Strullu, E. Cagli, and C. Dumas, "Deep learning for side-channel analysis and introduction to ascad database," *Journal of Cryptographic Engineering*, vol. 10, 06 2020.
[12] B. Hettwer, S. Gehrer, and T. Güneysu, "Profiled power analysis attacks using convolutional neural networks with domain knowledge," in *International Conference on Selected Areas in Cryptography*. Springer, 2018, pp. 479–498.
[13] J. Zhang, M. Zheng, J. Nan, H. Hu, and N. Yu, "A novel evaluation metric for deep learning-based side channel analysis and its extended application to imbalanced data," *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, vol. 2020, pp. 73–96, 2020.
[14] G. Zaid, L. Bossuet, F. Dassance, A. Habrard, and A. Venelli, "Ranking loss: Maximizing the success rate in deep learning side-channel analysis," Cryptology ePrint Archive, Paper 2020/872, 2020, https://eprint.iacr.org/2020/872. [Online]. Available: https://eprint.iacr.org/2020/872
[15] D. Thapar, M. Alam, and D. Mukhopadhyay, "Deep learning assisted cross-family profiled side-channel attacks using transfer learning," in *2021 22nd International Symposium on Quality Electronic Design (ISQED)*. IEEE, 2021, pp. 178–185.
[16] H. Yu, H. Shan, M. Panoff, and Y. Jin, "Cross-device profiled side-channel attacks using meta-transfer learning," in *2021 58th ACM/IEEE Design Automation Conference (DAC)*. IEEE, 2021, pp. 703–708.
[17] A. Garg and N. Karimian, "Leveraging deep cnn and transfer learning for side-channel attack," in *2021 22nd International Symposium on Quality Electronic Design (ISQED)*. IEEE, 2021, pp. 91–96.
[18] B. Hettwer, T. Horn, S. Gehrer, and T. Güneysu, "Encoding power traces as images for efficient side-channel analysis," in *2020 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*. IEEE, 2020, pp. 46–56.
[19] N. Chang, D. Zhu, L. Lin, D. Selvakumaran, J. Wen, S. Pan, W. Xia, H. Chen, C. Chow, and G. Chen, "Ml-augmented methodology for fast thermal side-channel emission analysis," in *Proceedings of the 26th Asia and South Pacific Design Automation Conference*, ser. ASPDAC '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 463–468. [Online]. Available: https://doi.org/10.1145/3394885.3431641
[20] J. Wen, N. Chang, L. Lin, D. Luo, J.-S. R. Jang, and H. Chen, "Security integrity analytics by thermal side-channel simulation: An ml-augmented auto-poi approach," in *Proceedings of DesignCon*, January 2022.
[21] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
[22] M. Kerkhof, L. Wu, G. Perin, and S. Picek, "Focus is key to success: A focal loss function for deep learning-based side-channel analysis," Cryptology ePrint Archive, Paper 2021/1408, 2021, https://eprint.iacr.org/2021/1408. [Online]. Available: https://eprint.iacr.org/2021/1408

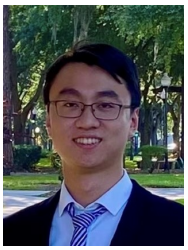
KAI ZHANG is currently pursuing a doctoral degree at National Taiwan University and is also an assistant professor at Yiwu Industrial and Commercial College. He obtained his master's degree from National Taiwan University in 2023 and his bachelor's degree from Providence University in 2021. His primary research areas include speech recognition and intent recognition, with research interests in large-scale language models and large-scale speech models.

PLACE PHOTO HERE

TUN-CHIEH LOU graduated from the Department of Computer Science and Information Engineering at National Taiwan University in 2022, earning his master's degree.

CHUNG-CHE WANG. received his Ph.D. from the CS Department at the National Tsing Hua University (Hsinchu, Taiwan) in 2017. His research interests include audio retrieval and audio processing

JYH-SHING ROGER JANG received the Ph.D. degreein electrical engineering and computer sciences fromthe Department of Electrical Engineering and ComputerSciences, University of California, Berkeley,Berkeley, CA, USA, in 1992. He studied fuzzy logicand artificial neural networks with Prof. Lotfi Zadeh,the father of fuzzy logic. After Ph.D., he joined theMathWorks to coauthor the Fuzzy Logic Toolbox(for MATLAB). He has since cultivated a keen interestin implementing industrial software for machinelearning. From 1995 to 2012, he was a Professorwith Computer Science Department, National Tsing Hua University, Hsinchu,Taiwan. Since August 2012, he has been a Professor with the Department ofComputer Science and Information Engineering, National Taiwan University(NTU), Taipei, Taiwan. He was the IT Director for NTU Hospital during2017–2019, and the Director for FinTech Center with NTU during 2018–2022.He is currently serving as CTO of E.Sun Financial Holding Company, Taipei.He has authored or coauthored one book titled Neuro-Fuzzy and Soft Computing(Prentice-Hall, 1997). He has also maintained toolboxes for machine learningand speech/audio processing. His research interests include machine learningin practice, with wide applications to speech recognition/assessment/synthesis,music analysis/retrieval, image classification, medical/healthcare data analytics,and FinTech. As of November 2022, Google Scholar shows more than 19,000citations for his seminal paper on adaptive neuro-fuzzy inference systems(ANFIS) published in 1993. Dr. Jang was the General Chair of the InternationalSociety forMusic Information Retrieval (ISMIR) Conference, Taipei, 2014 andwas a General Co-Chair of ISMIR Conference, Suzhou, 2017.



HENIAN LI is a Ph.D. candidate in the Department of Electrical and Computer Engineering, University of Florida, under the supervision of Prof. Mark Tehranipoor. His research interests include hardware security, fault-injection assessment, side-channel analysis, and secure scan.

LANG LIN is a product manager of Ansys ESOBU based in San Jose, California. He is dedicated into deploying power integrity, thermal integrity and IC security verification methodologies to the worldwide semiconductor customers to achieve IC product success. He holds a doctorate degree in electrical and computer engineering from University of Massachusetts with research expertise in low-power design, side-channel analysis, and hardware security. He has co-authored 30+ technical papers and patents, including the best paper award of IEEE HOST, iSES and the CEO innovation award of Ansys TechCon.

NORMAN CHANG co-founded Apache Design Solutions in February 2001 and currently serves as Ansys Fellow and Chief Technologist at Electronics, Semiconductor, and Optics BU, ANSYS, Inc. He is also currently leading AI/ML and security initiatives at ANSYS. Dr. Chang received his Ph.D. in Electrical Engineering and Computer Sciences from University of California, Berkeley. He holds 33 patents and has co-authored over 60 IEEE papers and a popular book on "Interconnect Analysis and Synthesis" by Wiley Interscience at 2000. Dr. Chang is an IEEE Fellow for his contribution on "Leadership and contributions to the physical-level sign-off of Electronic Design Automation for SoC/3DIC". He is also a recipient of 2024 "Distinguished Entrepreneur of the Year" Award from Chinese Institute of Engineers (CIE). He also actively engages in industry committees such as IEEE EDPS (Electronic Design Process Symposium) and SI2.