MT-MO: Efficient and Robust Non-Profiled Side-Channel Analysis Using Multitask Learning

Huy-Thanh Le[1], Van-Phuc Hoang[2*], Ngoc-Tuan Do[1], Xuan-Nam Tran[2]

[1]Telecommunications University, Khanh Hoa, Vietnam
[2]Le Quy Don Technical University, Hanoi, Vietnam
[*]Correspondence: phuchv@lqdtu.edu.vn

***Abstract*—Multi-Output Deep Learning based Side Channel Analysis (MO-DLSCA) with its variants have demonstrated their efficacy in non-profiled scenarios. However, current MO approaches remain limited to sequential single-byte recovery, failing to exploit shared leakage characteristics across the cryptographic key and resulting in redundant training overhead. This letter introduces a Multitask Multi-Output (MT-MO) deep learning architecture to mitigate this issue. By leveraging a shared feature extraction backbone with vectorized output heads, our model simultaneously recovers multiple key bytes in a single training session. Both Multitask Classification (MT-MOC) and Regression (MT-MOR) variants are introduced to target masking, noise injection, and desynchronization countermeasures. Experimental results obtained from three randomly selected target bytes indicate that the proposed models reduce the execution time by approximately 11 times compared to the sequential single-task model, while maintaining success rates of at least 73.33%.**

***Index Terms*—side channel attack, multi output, multitask**

I. Introduction

The increasing prevalence of embedded and IoT systems that process sensitive information has intensified the demand for strong hardware-assisted security. Although modern cryptographic algorithms are mathematically robust, their physical execution inevitably leaks unintended information—e.g., power consumption, electromagnetic emanations, or timing variations—that can be exploited through Side-Channel Analysis (SCA). These physical leakages have thus become a critical attack vector capable of revealing secret keys even when the underlying cryptographic design remains secure. In our ongoing project "AIPOSH" funded by the ASEAN IVO program, we are developing a comprehensive cybersecurity platform that integrates artificial intelligence powered hardware and software solutions for IoT based smart healthcare systems.

SCA techniques are broadly classified into profiled and non-profiled attacks. Profiled attacks rely on access to a reference device identical to the target, enabling the construction of highly accurate leakage models via extensive data collection and statistical [1] or machine-learning-based profiling [2]. While extremely powerful, their practicality is limited by the availability of such reference devices and the computational cost of the profiling phase, especially for closed commercial systems. In contrast, non-profiled attacks eliminate the profiling requirement and directly operate on traces from the target device. Classical methods, such as Differential Power Analysis (DPA) and Correlation Power Analysis (CPA), evaluate statistical relationships between hypothesized intermediate computations and observed leakages. More recently, deep learning has emerged as a promising direction in non-profiled SCA, with approaches such as Differential Deep Learning Analysis (DDLA) [3] demonstrating improved key-discrimination capabilities. However, DDLA imposes a computational burden by requiring the training process to be performed repeatedly to generate the metrics used for key recovery.

To address the computational bottleneck of DDLA, recent advancements have introduced Multi-Output Classification (MOC) [4] and Multi-Output Regression (MOR) [5]. These architectures integrate all 256 key hypotheses for a target byte into a single neural network with 256 corresponding output branches. MOC enables the simultaneous evaluation of all key candidates using categorical cross-entropy, achieving speedups of up to 30 times compared to DDLA. Similarly, MOR further improves this by using regression with Identity Labeling, allowing the model to directly estimate leakage values. This approach has exhibited speedups of 40 times faster than DDLA_{CNN} in de-synchronized scenarios. Despite their efficiency, current MOC and MOR implementations operate on a single-byte basis. In a realistic full-key recovery scenario (e.g., attacking all 16 bytes of AES-128), an attacker must replicate the attack procedure 16 times sequentially. This sequential approach presents two major limitations:

- Operational inefficiency: Training 16 independent models incurs significant overhead in terms of framework initialization, data management, and optimization steps.

- Lack of shared knowledge: Although the input traces for different bytes correspond to different time windows, they originate from the same hardware device performing the same cryptographic operation (e.g., SubBytes). The physical leakage characteristics, such as the power profile of an S-Box substitution, are fundamentally similar across all bytes. Treating each byte as an isolated learning task ignores this structural similarity, forcing the network to independently learn the same leakage patterns multiple times.

To overcome these limitations, this paper proposes a Multitask Multi-Output (MT-MO) architecture designed to attack multiple key bytes simultaneously in a single training session. Our approach leverages Hard Parameter Sharing [6], where a single CNN backbone is shared across multiple inputs.

The shared features are then fed into byte-specific output heads (MOC or MOR) to recover the respective sub-keys. By unifying the attack into a multitask framework, we not only eliminate the overhead of sequential training but also improve the model's generalization by exposing the shared backbone to leakage variations across multiple bytes. Experimental results will demonstrate that this architecture significantly reduces the total attack time while maintaining high success rates against countermeasures like masking and hiding.

II. Data Preparation

A. Experimental Platforms

To evaluate the robustness of the proposed Multitask architecture against various side-channel countermeasures, three distinct datasets derived from the ChipWhisperer (CW) and ASCAD databases were utilized.

1) Desynchronized CW Dataset: A collection of 5,000 power traces from an ATMEL XMEGA microcontroller executing AES-128 using the ChipWhisperer platform. Each trace consists of 500 samples. To simulate a hiding countermeasure, a random temporal shift within the range of $[0, 20]$ samples is applied to each trace relative to the trigger.

2) Masked ASCAD Dataset: To evaluate performance against first-order Boolean masking, we utilize the standard ASCAD database [7] This dataset contains traces captured from an 8-bit ATMega8515 executing a masked AES implementation. We utilized a subset of 20,000 traces corresponding to the processing of the third S-Box.

3) Noisy ASCAD Dataset: To investigate the impact of noise-based hiding, a modified version of the ASCAD dataset was generated. In this configuration, the mask protection is computationally removed using the known mask values, and the traces are augmented with synthetic Additive White Gaussian Noise (AWGN) with a mean $\mu = 0$ and a standard deviation $\sigma = 1.0$.

B. Dataset Reconstruction

To optimize the data for Multitask Learning, specific reconstruction and labeling strategies were applied. For the ASCAD datasets (both masked and noisy), Point of Interest (POI) selection based on Signal-to-Noise Ratio (SNR) was performed, reducing each trace to 700 relevant samples [7]. For the CW dataset, the trace length was maintained at 500 samples to preserve the desynchronization effects.

The labeling strategy is adapted to the specific objective of each model variant. For the MOC model, the Least Significant Bit (LSB) of the S-Box output is used as the label. Conversely, for the MOR model, the Hamming Weight (HW) of the S-Box output is employed, as it linearly correlates with power consumption in software implementations. Consequently, the final datasets consist of input traces (500 samples for CW, 700 for ASCAD) paired with a label vector corresponding to the 256 key hypotheses for each target byte. The overall structure of the datasets utilized in our experiments is visually presented in Figure 1.

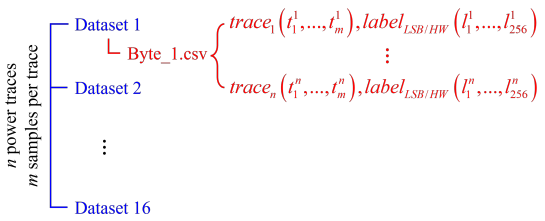

Fig. 1. Structure of the new datasets: There are 16 folders (blue) corresponding to 16 bytes of the secret key, each folder contains a csv file (red) which stores values of power traces and labels corresponding to 256 hypothesis keys.

III. Proposed Multitask Multi-output Deep Learning architecture

This section details the proposed Multitask Multi-Output (MT-MO) architecture, designed to overcome the sequential training bottleneck of previous non-profiled SCA methods. By leveraging *Hard Parameter Sharing* and *Vectorized Output Heads*, the proposed model simultaneously recovers multiple key bytes in a single training session while maintaining the efficiency of the Multi-Output approach.

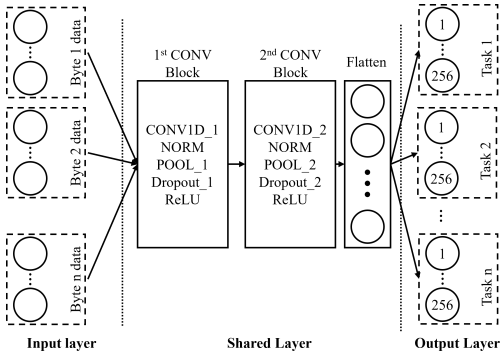

Fig. 2. Detailed structure of the proposed MT-MO model.

As illustrated in Fig. 2, the proposed architecture is composed of three main stages: the Multitask Input Layer, the Shared Feature Extraction Backbone, and the Task-Specific Output Heads.

Multitask Input Layer: The left side of the diagram depicts the input stage, which accepts n (where $1 \leq n \leq 16$) distinct trace segments simultaneously. Each input branch (labeled Byte 1 to Byte n) corresponds to the specific time window of a target byte's leakage in the AES-128 algorithm.

Shared Layer (Backbone): The core of the architecture is the Shared Layer, which utilizes a *Hard Parameter Sharing* mechanism. All input branches are processed by the same set of weights to extract generalized leakage features. This backbone consists of two sequential Convolutional Blocks. Each block commences with a 1D Convolutional layer (`CONV1D`) to capture temporal dependencies. This is followed by Batch Normalization (`NORM`) to stabilize learning and Average Pooling (`POOL`) to reduce dimensionality. Crucially, a `Dropout` layer is inserted after pooling to prevent overfitting, ensuring the model learns robust features rather than memorizing noise.

Finally, a Rectified Linear Unit (`ReLU`) activation function is applied.

Output Layer: The output from the shared backbone is flattened into a single feature vector. This vector is then bifurcated into n independent task-specific heads (Task 1 to Task n). Consistent with the vectorized output strategy, each head consists of a fully connected layer with exactly 256 nodes, corresponding to the 256 possible sub-key values.

A. Multitask Multi-Output Classification (MT-MOC)

The MT-MOC architecture is designed to perform simultaneous Non-Profiled attacks on multiple target bytes by classifying the Least Significant Bit (LSB) of the S-Box output. Unlike the original MOC, which employs a single-input single-task structure, our proposed architecture accepts a set of inputs corresponding to different target bytes and processes them through a unified pipeline.

1) Shared Feature Extraction Backbone: The core of the architecture is a shared Convolutional Neural Network (CNN) backbone, denoted as $\mathcal{F}_{\theta_{share}}$. Let $\mathbf{T} = \{\mathbf{t}^{(1)}, \mathbf{t}^{(2)}, \ldots, \mathbf{t}^{(B)}\}$ represent the input set, where $\mathbf{t}^{(b)}$ is the trace segment associated with the b-th target byte and B is the number of simultaneous tasks. These inputs are processed by the same set of convolutional filters, batch normalization, and pooling layers. This *Hard Parameter Sharing* strategy forces the network to learn a generalized, translation-invariant representation of the power leakage that is consistent across different byte operations. The shared feature vector $\mathbf{f}^{(b)}$ for the b-th byte is computed as $\mathbf{f}^{(b)} = \mathcal{F}_{\theta_{share}}(\mathbf{t}^{(b)})$.

2) Vectorized Output Heads: A key contribution in this work is the vectorization of the output heads to optimize the training pipeline. In traditional implementations, the 256 key hypotheses might be handled by separate branches or sequential evaluations, which fragments the computational graph and underutilizes GPU parallelism. Conversely, our approach fuses the 256 hypotheses into a single dense layer, allowing the gradients for all candidates to be computed via optimized matrix multiplication.

For the MT-MOC variant, the output head $H^{(b)}_{MOC}$ for the b-th byte projects the feature vector $\mathbf{f}^{(b)}$ into a probability space for 256 key candidates:

$$\hat{\mathbf{y}}^{(b)} = \sigma(\mathbf{W}^{(b)}\mathbf{f}^{(b)} + \mathbf{c}^{(b)}) \tag{1}$$

where $\mathbf{W}^{(b)} \in \mathbb{R}^{256\times d}$ is the weight matrix, $\mathbf{c}^{(b)}$ is the bias, and $\sigma(\cdot)$ is the **Sigmoid** activation function. The use of Sigmoid is imperative because the LSB classification for each key hypothesis is treated as an independent Bernoulli trial. Unlike Softmax, which enforces a probability distribution summing to one, Sigmoid allows the network to independently estimate the probability $P(\text{LSB} = 1|\mathbf{t}, k)$ for every key candidate k, without mutual suppression.

3) Loss Function: The model minimizes the Binary Cross-Entropy (BCE) loss, averaged across all 256 key hypotheses. The loss for the b-th task is defined as:

$$\mathcal{L}^{(b)}_{MOC} = -\frac{1}{256}\sum_{k=0}^{255}\left[y_k^{(b)}\log(\hat{y}_k^{(b)}) + (1-y_k^{(b)})\log(1-\hat{y}_k^{(b)})\right] \tag{2}$$

where $y_k^{(b)}$ is the ground truth LSB label derived from the k-th key guess. This vectorized loss calculation allows the network to converge by identifying the key hypothesis that consistently minimizes the entropy between the predicted probability and the actual leakage behavior.

To enable the simultaneous recovery of multiple key bytes, the global objective function $\mathcal{L}_{Total}$ is computed as the weighted summation of losses from all B target bytes:

$$\mathcal{L}_{Total} = \sum_{b=1}^{B} w_b \cdot \mathcal{L}^{(b)}_{MOC} \tag{3}$$

where B is the number of target bytes and w_b represents the weight assigned to the b-th task. This joint optimization strategy forces the shared backbone to learn generalized features that are robust across all simultaneous tasks.

B. Multitask Multi-Output Regression (MT-MOR)

The MT-MOR architecture shares the same feature extraction backbone $\mathcal{F}_{\theta_{share}}$ as the MT-MOC model but fundamentally differs in its prediction objective and labeling strategy. While the MOC approach classifies discrete bits, the MOR approach directly estimates the scalar leakage value associated with the cryptographic operation.

1) Hamming Weight Labeling vs. Identity: A significant deviation from the original MOR proposal is the adoption of *Hamming Weight (HW)* labeling instead of Identity (ID) labeling. The original MOR attempts to regress the S-Box output value (0-255) directly. However, the physical power consumption of CMOS devices is linearly correlated with the HW of the data being processed, not its integer value. The relationship between the ID and power consumption is highly non-linear and complex for a regression model to capture accurately. By mapping the labels to the HW domain ($y \in [0, 8]$), we align the learning objective with the physical leakage model $L \approx \alpha \cdot HW(S(x \oplus k)) + \beta$, thereby simplifying the regression task and accelerating convergence.

2) Linear Activation and MSE Loss: To support this regression task, the task-specific heads in MT-MOR employ a **Linear** activation function instead of Sigmoid. The output is computed as:

$$\hat{\mathbf{y}}^{(b)} = \mathbf{W}^{(b)}\mathbf{f}^{(b)} + \mathbf{c}^{(b)} \tag{4}$$

This allows the network to predict continuous values representing the estimated leakage. Consequently, the optimization objective is to minimize the Euclidean distance between the predicted leakage and the actual Hamming Weight. The Mean Squared Error (MSE) loss function is utilized::

$$\mathcal{L}^{(b)}_{MOR} = \frac{1}{256}\sum_{k=0}^{255}\left(y_k^{(b)} - \hat{y}_k^{(b)}\right)^2 \tag{5}$$

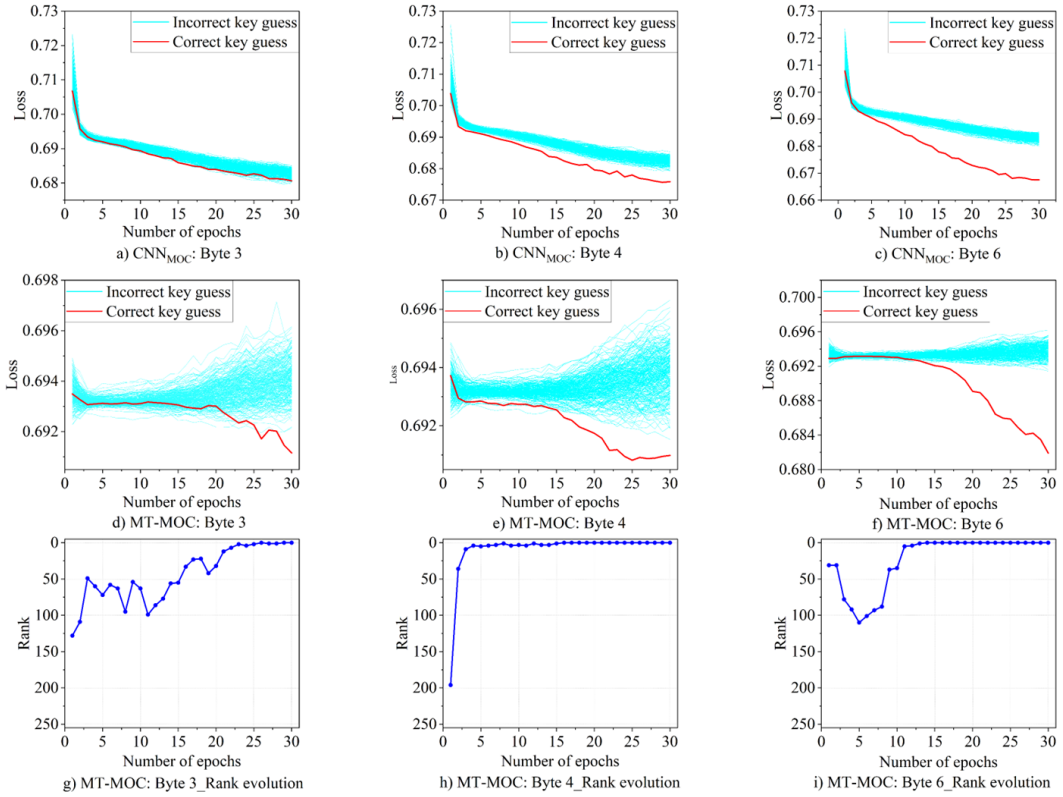

Fig. 3. Attack results on masking dataset. First row: CNN$_{MOC}$ model, second row: proposed MT-MOC model, third row: rank evolution of the correct key.

The use of MSE is appropriate here as it heavily penalizes large deviations, forcing the network to align its predictions with the linear leakage model of the correct key. Incorrect key hypotheses, which produce uncorrelated labels, will inherently result in a high irreducible error (high MSE), while the correct key will converge to a minimal MSE, allowing efficient key recovery.

3) Joint Optimization: To simultaneously recover B target bytes, the total loss $\mathcal{L}_{total}$ is computed as the summation of losses from all task-specific branches: $\mathcal{L}_{total} = \sum_{b=1}^{B} \mathcal{L}^{(b)}$. This joint optimization strategy ensures that the shared backbone learns features that are robust and universally applicable to the cryptographic algorithm's operation, leading to improved generalization and significantly reduced attack time compared to training B isolated models.

IV. Experimental Evaluation

The efficacy of the proposed architecture is demonstrated through a series of experiments on datasets employing different counter-measurements. All experimental procedures were implemented using the Keras framework on a laptop with an Apple M1 processor and 8GB of unified memory. The assessment metrics include the average Success Rate (SR) and total computational time (T_A). Notably, the multitask architecture increases memory consumption linearly with the number of output heads. Therefore, due to the hardware constraints, the simultaneous recovery analysis was restricted to a subset of three randomly selected bytes of the secret key. Given the independent nature of byte-wise operations in the AES algorithm (specifically the S-Box substitution), these findings can be extrapolated to the remaining key bytes. Consequently, this setup serves as a representative proof-of-concept for the proposed method's scalability.

A. Masking

To evaluate the efficiency of the proposed model on masked data, we conducted experiments comparing the MOC model from [6] (denoted as CNN$_{MOC}$) and the proposed MT-MOC model using the ASCAD dataset with 15,000 power traces. The experiments were repeated 30 times to compute the Success Rate (SR) and average attack time (T_A). The graphical results are depicted in Fig. 3, and the quantitative metrics are summarized in Table I.

Observing the loss graphs in Fig. 3, a distinct difference in key discrimination capability is evident. For the baseline CNN$_{MOC}$ (a-c), this model failed to retrieve the correct key for Byte 3 but succeeded in recovering for Byte 4 and Byte 6. However, the separation between the loss curves of the correct key guess (red) and incorrect guesses (cyan) is relatively narrow. Conversely, for the MT-MOC model (d-f), this margin is significantly wider and manifests earlier during the training phase, indicating superior leakage extraction even under masking countermeasures. This rapid convergence is

TABLE I
ATTACK RESULTS (TIME AND SUCCESS RATE) OF BASELINE AND PROPOSED MODEL ON DIFFERENT DATASETS.

Model	**Metrics**	Dataset								
		ASCAD			Unmasked ASCAD + noise			CW desync		
		Byte 3	Byte 4	Byte 6	Byte 5	Byte 6	Byte 7	Byte 1	Byte 5	Byte 7
CNN_{MOC}	T_A (s)	179.2	179.5	179.9	-	-	-	-	-	-
	SR (%)	50	96.7	100	-	-	-	-	-	-
MT-MOC	T_A (s)	**116.1**			-	-	-	-	-	-
	SR (%)	**73.3**	**100**	**100**	-	-	-	-	-	-
CNN_{MOR}	T_A (s)	-	-	-	104.2	105.8	107.2	595.8	596.7	598.1
	SR (%)	-	-	-	100	100	100	86.7	56.7	53.3
MT-MOR	T_A (s)	-	-	-	**28.2**			**97.6**		
	SR (%)	-	-	-	**100**	**100**	**100**	**93.3**	**86.7**	**80**

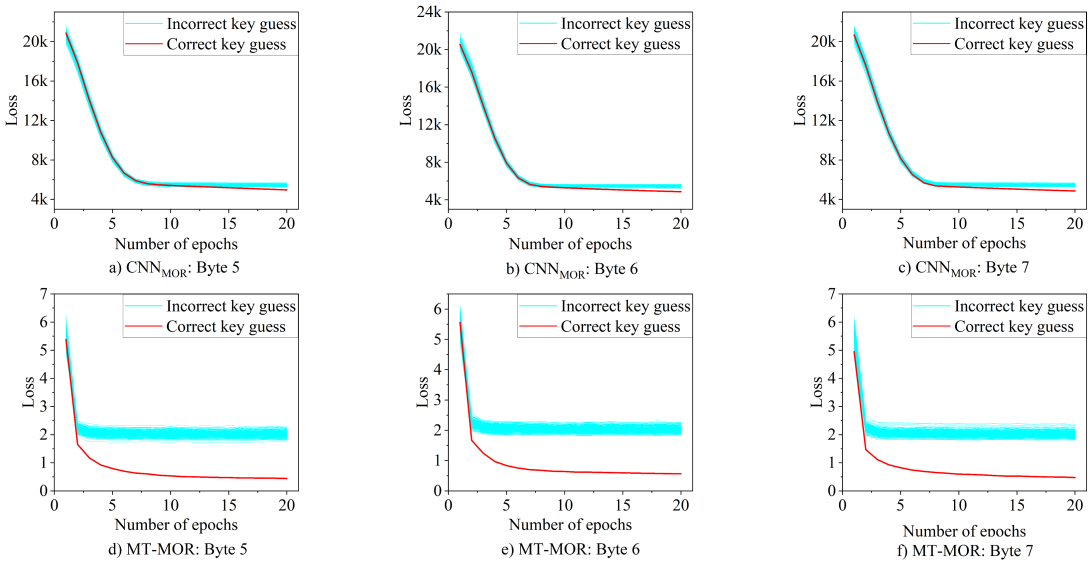

Fig. 4. Attack results on noisy dataset. First row: CNN_{MOR} model, second row: proposed MT-MOR model.

further corroborated by the rank evolution plots (g-i), where the rank of the correct key swiftly descends to 0.

The quantitative data in Table I reinforces these visual observations. Regarding accuracy, while CNN_{MOC} achieved an SR of only 50% for Byte 3, MT-MOC significantly improved this to 73.3%. For Bytes 4 and 6, the proposed model attained absolute accuracy (100%), outperforming the 96.7% and 100% of the baseline, respectively. Notably, in terms of computational efficiency, MT-MOC demonstrated a substantial advantage by reducing the total attack time for all three bytes to 116 seconds, which is approximately 4.6 times faster than the 538 seconds required by CNN_{MOC}. These results substantiate that MT-MOC not only enhances the attack capability but also significantly optimizes computational costs.

B. Noise generation

To evaluate the robustness against noise-based hiding countermeasures, we employed the CNN_{MOR} and MT-MOR models to attack the unmasked ASCAD dataset augmented with Gaussian Noise ($\mu = 0, \sigma = 1.0$). This dataset comprises 5,000 power traces. The convergence behavior of the attacks is illustrated in Fig. 4. The experiments were also repeated 30 times to validate the reliability of the proposed models, and the performance metrics are summarized in Table I. As depicted in Fig. 4, the original CNN_{MOR} model exhibits a relatively slow convergence with a negligible margin between the loss of the correct key and incorrect hypotheses. Conversely, the proposed MT-MOR model demonstrates superior discrimination capabilities; the loss metric for the correct key drops significantly and maintains a distinct gap from the incorrect guesses, thereby facilitating key recovery.

Regarding computational efficiency, Table I highlights a substantial improvement. While both models achieved a 100% Success Rate (SR) across target bytes 5, 6, and 7, the execution time (T_A) differs significantly. The CNN_{MOR} required an average of roughly 105 seconds per byte (totaling $\approx$ 317 seconds for sequential execution). In contrast, the MT-MOR architecture recovered all three bytes simultaneously in just 28.2 seconds, representing a speedup factor of approximately $11\times$.

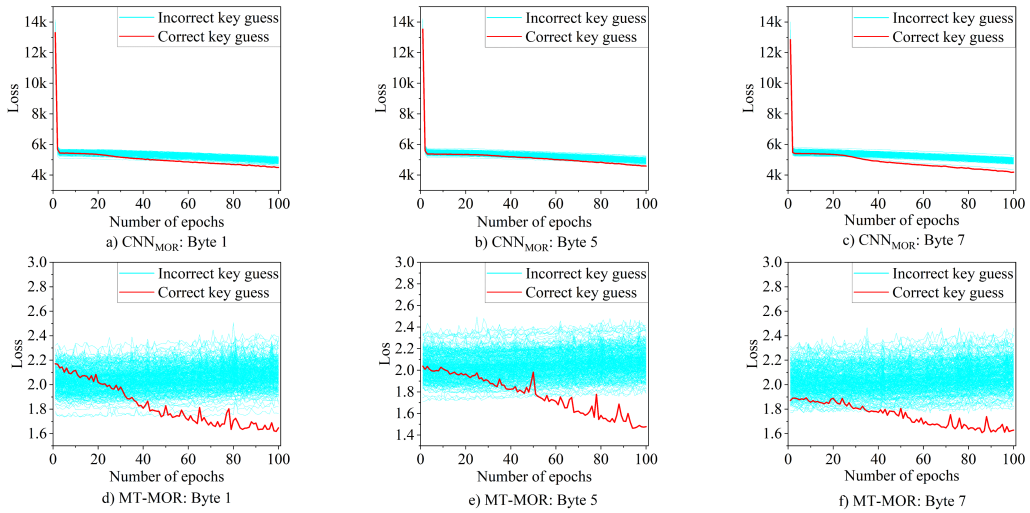

Fig. 5. Attack results on desynchronization dataset. First row: CNN_{MOR} model, second row: proposed MT-MOR model

C. De-synchronization

In this experiment, we adopted the MOR-based models due to their superior performance observed in the previous ones. The dataset employed in this experiment is derived from the ChipWhisperer platform, comprising 5,000 power traces. To simulate desynchronization, samples were randomly shifted by a maximum of 20 samples relative to the original trigger position. The experimental procedure was conducted similarly to the previous benchmarks. The attack results are illustrated in Fig. 5 in which the temporal misalignment poses a significant challenge to the regression models. The CNN_{MOR} exhibits poor convergence characteristics, the loss trajectory of the correct key is barely distinguishable from the incorrect hypotheses, particularly for Bytes 1 and 5. This visual ambiguity is reflected in the sub-optimal Success Rates (SR) recorded in Table I, where the CNN_{MOR} achieves only 56.7% and 53.3% for Bytes 5 and 7, respectively. In contrast, the proposed MT-MOR demonstrates remarkable robustness against jitter. By leveraging the shared feature extraction backbone, the model effectively learns translation-invariant features. Although the loss curves exhibit high variance, a clear separation between the correct key and the incorrect guesses is maintained across all target bytes. This improved discrimination is quantified by a substantial increase in success rates, reaching up to 93.3%. Furthermore, the multitask architecture retains its computational efficiency, completing the attack on all three bytes in 97.6 seconds.

V. Conclusion

This article proposes a multitask multi-output deep learning framework designed to enhance the efficiency and robustness of non-profiled side-channel analysis. By implementing a hard parameter sharing strategy, the proposed architecture effectively mitigates the computational redundancy inherent in sequential single-task attacks. Experimental evaluations across diverse countermeasures yield distinct advantages for the multitask approach. In the context of masking, the MT-MOC model improved the Success Rate on difficult targets up to 73.3%, achieving a speedup factor of approximately $4.6\times$. Furthermore, against hiding countermeasures, the MT-MOR variant exhibited remarkable resilience. Under noise injection scenarios, the model reduced the attack time by approximately $11\times$ while maintaining a 100% success rate. These results collectively validate that utilizing shared feature extraction facilitates better generalization and operational efficiency.

Acknowledgment

This publication is the output of the ASEAN IVO project, "Artificial Intelligence Powered Comprehensive Cyber-Security for Smart Healthcare Systems (AIPOSH)", and financially supported by NICT, Japan.

References

[1] S. Chari, J. R. Rao, and P. Rohatgi, "Template attacks," in *Cryptographic Hardware and Embedded Systems - CHES 2002*, B. S. Kaliski, ç. K. Koç, and C. Paar, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 13–28.

[2] E. Cagli, C. Canovas, and E. Prouff, "Convolutional neural networks with data augmentation against jitter-based countermeasures - profiling attacks without pre-processing," in *Workshop on Cryptographic Hardware and Embedded Systems*, 2017. [Online]. Available: https://api.semanticscholar.org/CorpusID:54088207

[3] B. Timon, "Non-profiled deep learning-based side-channel attacks with sensitivity analysis," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, vol. 2019, no. 2, p. 107–131, Feb. 2019. [Online]. Available: https://tches.iacr.org/index.php/TCHES/article/view/7387

[4] V.-P. Hoang, N.-T. Do, and V. S. Doan, "Efficient nonprofiled side-channel attack using multi-output classification neural network," *IEEE Embedded Systems Letters*, vol. 15, no. 3, pp. 145–148, 2023.

[5] N.-T. Do, V.-P. Hoang, and V. S. Doan, "A novel non-profiled side channel attack based on multi-output regression neural network," *Journal of Cryptographic Engineering*, vol. 14, no. 3, pp. 427–439, Sep 2024. [Online]. Available: https://doi.org/10.1007/s13389-023-00314-4

[6] S. Ruder, "An overview of multi-task learning in deep neural networks," 2017. [Online]. Available: https://arxiv.org/abs/1706.05098

[7] E. Prouff, R. Strullu, R. Benadjila, E. Cagli, and C. Dumas, "Study of deep learning techniques for side-channel analysis and introduction to ASCAD database," Cryptology ePrint Archive, Paper 2018/053, 2018. [Online]. Available: https://eprint.iacr.org/2018/053