
BEYOND TVLA: ANDERSON–DARLING LEAKAGE ASSESSMENT FOR NEURAL NETWORK SIDE-CHANNEL LEAKAGE DETECTION

Ján Mikulec

Faculty of Informatics and Information technologies
Slovak University of Technology
Bratislava, Slovakia
jan.mikulec@stuba.sk

Jakub Breier

TTControl GmbH
Vienna, Austria
jbreier@jbreier.com

Xiaolu Hou

Faculty of Informatics and Information technologies
Slovak University of Technology
Bratislava, Slovakia
xiaolu.hou@stuba.sk

March 20, 2026

ABSTRACT

Test Vector Leakage Assessment (TVLA) based on Welch’s t -test has become a standard tool for detecting side-channel leakage. However, its mean-based nature can limit sensitivity when leakage manifests primarily through higher-order distributional differences. As our experiments show, this property becomes especially crucial when it comes to evaluating neural network implementations. In this work, we propose Anderson–Darling Leakage Assessment (ADLA), a leakage detection framework that applies the two-sample Anderson–Darling test for leakage detection. Unlike TVLA, ADLA tests equality of the full cumulative distribution functions and does not rely on a purely mean-shift model.

We evaluate ADLA on a multilayer perceptron (MLP) trained on MNIST and implemented on a ChipWhisperer-Husky evaluation platform. We consider protected implementations employing shuffling and random jitter countermeasures. Our results show that ADLA can provide improved leakage-detection sensitivity in protected implementations for a low number of traces compared to TVLA.

1 Introduction

The deployment of neural networks on embedded and edge platforms has accelerated rapidly, driven by applications ranging from vision and biometrics to industrial monitoring and automotive control. While these deployments enable low-latency inference close to the data source, they also expose implementations to physical attacks [1]. In particular, side-channel analysis (SCA) [2] can exploit data-dependent variations in power consumption or electromagnetic emanations to infer sensitive information about intermediate computations, model parameters, or user inputs. As a result, practical countermeasures such as masking [3], shuffling [4], and jitter-based techniques [5] are increasingly considered when implementing machine-learning workloads on constrained devices.

A widely adopted first step in evaluating side-channel resistance is *leakage assessment*, which aims to determine whether data-dependent leakage is present without committing to a specific attack strategy [6]. The de facto standard methodology is Test Vector Leakage Assessment (TVLA) [7]. TVLA is attractive due to its simplicity and its well-established thresholding practice, however, it is fundamentally a mean-based test. When countermeasures reduce or hide mean shifts (e.g., through shuffling or desynchronization), leakage may persist in the form of higher-order distributional

differences that are less visible to a purely mean-sensitive statistic. This motivates the development of complementary leakage assessment techniques that can detect discrepancies beyond the first moment.

In this work, we propose *Anderson–Darling Leakage Assessment (ADLA)*, a leakage detection framework that leverages the two-sample Anderson–Darling test to compare leakage distributions arising from two controlled input conditions. In contrast to TVLA, which tests the equality of means, ADLA evaluates whether the two distributions share the same cumulative distribution function (CDF), thereby providing sensitivity to a broader class of leakage effects. We further derive an explicit decision threshold for ADLA to enable practical use in evaluation workflows.

We validate ADLA on a neural-network inference implementation measured on a ChipWhisperer-based side-channel acquisition setup. We evaluate protected implementations employing shuffling and jitter. Our experiments show that ADLA detects leakage with substantially fewer traces than TVLA in this setting, including cases where TVLA remains below its detection threshold. These results indicate that distribution-sensitive assessment can be particularly valuable when countermeasures reduce mean-based leakage signatures.

Contributions. This paper makes the following contributions:

- We introduce ADLA, a leakage assessment method for neural-network implementations based on the two-sample Anderson–Darling test.
- We derive a detection threshold for ADLA, enabling decision-making with a practical significance level to observe leakage.
- We experimentally demonstrate that ADLA provides higher leakage-detection sensitivity than TVLA at low trace counts on protected implementations, improving time efficiency in practical evaluation campaigns.

Practical relevance. From an evaluation perspective, improved sensitivity at low trace counts directly translates into shorter acquisition campaigns and reduced measurement cost. This is particularly beneficial for certification and testing laboratories, where throughput and time-to-result are critical and collecting very large trace sets may be impractical.

2 Related Work

In this section, we review the background relevant to our work. We first introduce neural networks (Subsection 2.1), followed by an overview of side-channel analysis attacks on neural network implementations (Subsection 2.2). We then discuss existing countermeasures (Subsection 2.3) and conclude with a review of leakage assessment methodologies (Subsection 2.4).

2.1 Neural Networks

Neural networks [8] are computational models composed of layers of interconnected neurons, whose behavior is governed by trainable parameters, typically weights and biases. During inference, these parameters, together with the chosen activation functions, determine the sequence of linear transformations and nonlinear mappings that produce the network output.

A Multilayer Perceptron (MLP) is a fundamental class of feedforward neural networks consisting of an input layer, one or more hidden layers, and an output layer. Each neuron computes a weighted sum of its inputs, adds a bias term, and applies a nonlinear activation function. The network parameters are typically optimized via backpropagation [9] combined with gradient-based learning algorithms to minimize a task-specific loss function. In a standard fully connected MLP architecture, every neuron in a given layer is connected to all neurons in the subsequent layer. This dense connectivity enables the network to approximate nonlinear functions and makes MLPs suitable for a wide range of classification and regression tasks.

2.2 Side-channel Analysis Attacks on Neural Networks

Side-channel analysis (SCA) attacks on neural network implementations typically assume a black-box setting in which the network architecture and model parameters are secret. The adversary observes physical side-channel information, so called side-channel *leakages*, such as timing behavior, power consumption, or electromagnetic (EM) emissions, during inference or training to gain information about the neural network. For example, differences in activation-function execution time may reveal the type of activation function used [2], power/EM side-channel information can leak sensitive input information [10], or expose internal architectural features such as layer types [11].

Of particular relevance to our work is the correlation power analysis (CPA) attack [2, 4, 12], in which statistical correlation is computed between hypothetical intermediate values (derived from candidate secret parameters) and

measured side-channel leakages. By identifying the parameter hypothesis that maximizes the statistical dependence with the observed leakage, an attacker can recover secret model parameters. Such attacks fundamentally rely on the data-dependency of physical leakages and the statistical distinguishability of the resulting distributions.

2.3 Countermeasures Against Side-Channel Analysis Attacks

Several countermeasures have been proposed to mitigate SCA attacks on neural network implementations. Desynchronization-based techniques [5] introduce jitters to the computations to randomize execution time and reduce the effectiveness of timing-based attacks. Masking approaches [3] randomize intermediate computations to decrease the statistical dependence between the measured leakage and sensitive variables, and have been demonstrated for neural networks with integer weights to hinder correlation power analysis (CPA). However, applying masking across an entire network typically incurs substantial computational and implementation overhead.

In this work, we adopt two countermeasures against CPA attacks: shuffling and random jitter. Shuffling randomizes the order of multiplications within a layer to disrupt CPA [13], with subsequent work [4] protecting the shuffled index generation mechanism itself against SCA [14]. Random jitter introduces random delays into the computation to desynchronize side-channel measurements [15]. While well-studied in cryptographic implementations, its application to neural networks has mainly addressed timing-based attacks [5], and its impact on power-based CPA remains less explored.

2.4 Leakage Assessment

Leakage assessment was originally developed for evaluating the side-channel security of cryptographic implementations. From a developer’s perspective, it provides a systematic methodology to determine whether an implementation exhibits detectable data-dependent leakage, without requiring knowledge of a specific attack strategy. As new side-channel attacks continue to emerge, it is generally impractical to validate resistance against each attack individually. Leakage assessment addresses this challenge by analyzing measured side-channel leakages and determining whether statistically significant data-dependent information is present [16].

Among the proposed methodologies, Test Vector Leakage Assessment (TVLA) [7] has become the de facto standard for evaluating cryptographic implementations. TVLA employs statistical hypothesis testing to detect leakage. More recently, TVLA has also been adopted to evaluate the side-channel security of neural network implementations. To the best of our knowledge, TVLA remains the only established leakage assessment methodology currently applied to neural network implementations. Further details on TVLA and its statistical foundations are provided in Subsections 3.2 and 4.1.

Beyond mean-based TVLA, side-channel leakage can also be evaluated using χ^2 -type tests [17]. In our setting, however, these tests were shown to be highly sensitive to the discretization of the measurement space, particularly the choice of binning, thus making them hard to interpret.

3 Statistical Hypothesis Testing

This section introduces the notation and basic concepts of statistical hypothesis testing (Subsection 3.1), reviews Welch’s t -test underlying TVLA (Subsection 3.2), and describes the two-sample Anderson–Darling test on which our ADLA framework is based (Subsection 3.3).

3.1 Notation and Preliminaries

A *statistical hypothesis* [18] is a formal statement concerning one or more unknown parameters of the underlying probability distribution(s) governing the data. It is termed a hypothesis because its validity is not known *a priori* and must be assessed based on observed data.

Statistical hypothesis testing is a methodological framework that uses sample data to evaluate such statements. More precisely, it provides a decision rule for determining whether the observed sample is consistent with a specified hypothesis about the underlying data-generating distribution(s). Based on the outcome of the test, the hypothesis is either rejected or not rejected. Importantly, failing to reject a hypothesis does not imply that it is true; rather, it indicates that the observed data do not provide sufficient evidence against it.

The hypothesis under investigation is referred to as the *null hypothesis*, denoted by H_0 . It is tested against a competing statement called the *alternative hypothesis*, denoted by H_1 . The performance of the test is characterized by its *significance level*, denoted by α , which is defined as an upper bound on the probability of rejecting H_0 when H_0 is true

(Type I error). For a given choice of α , a critical region (or equivalently, a decision threshold) is determined according to the distribution of the test statistic under H_0 .

In this paper, we focus on the comparison of two probability distributions. Let X and Y denote the corresponding random variables. Let $\{X_1, X_2, \dots, X_n\}$ and $\{Y_1, Y_2, \dots, Y_n\}$ be independent samples drawn from the distributions of X and Y , respectively¹.

A test statistic is computed from the observed samples. If the value of this statistic falls within the critical region (equivalently, exceeds the predefined threshold), the null hypothesis H_0 is rejected; otherwise, it is not rejected.

3.2 Welch's t -test

Welch's t -test [19] is a parametric test designed to compare the means of two normal distributions.

Let $X \sim \mathcal{N}(\mu_x, \sigma_x^2)$ and $Y \sim \mathcal{N}(\mu_y, \sigma_y^2)$ denote two independent random variables corresponding to the distributions under consideration. The null and alternative hypotheses are defined as

$$H_0 : \mu_x = \mu_y, \quad H_1 : \mu_x \neq \mu_y.$$

The test statistic is given by

$$t := \frac{\bar{X} - \bar{Y}}{\sqrt{\frac{S_x^2}{n} + \frac{S_y^2}{n}}}, \quad (1)$$

where \bar{X} and \bar{Y} denote the sample means, and S_x^2 and S_y^2 denote the unbiased sample variances of the respective samples.

Under the null hypothesis, the statistic t approximately follows a t -distribution. For sufficiently large sample sizes, the distribution of t converges to the standard normal distribution by the central limit theorem. In this asymptotic regime, the threshold corresponding to a significance level α is $z_{\alpha/2}$, defined by

$$\Phi(z_{\alpha/2}) = 1 - \frac{\alpha}{2},$$

where Φ denotes the cumulative distribution function of the standard normal distribution. Equivalently,

$$\frac{\alpha}{2} = 1 - \Phi(z_{\alpha/2}). \quad (2)$$

The null hypothesis is rejected if $|t| > z_{\alpha/2}$. In this case, at significance level α , the observed data provide sufficient statistical evidence to reject H_0 in favor of H_1 , indicating that the population means differ.

3.3 Two-sample Anderson-Darling Test

The two-sample Anderson–Darling test [20] is a nonparametric procedure for testing whether two independent samples originate from the same (continuous) distribution. In contrast to mean-based tests, it compares the entire distributions via their cumulative distribution functions.

Let F_x and F_y denote the cumulative distribution functions (CDFs) of the random variables X and Y , respectively. The null and alternative hypotheses are

$$H_0 : F_x = F_y, \quad H_1 : F_x \neq F_y.$$

Let $\{X_1, \dots, X_n\}$ and $\{Y_1, \dots, Y_n\}$ be two independent samples of equal size n . Consider the pooled sample of size $2n$, arranged in increasing order,

$$Z_{(1)} \leq Z_{(2)} \leq \dots \leq Z_{(2n)}.$$

The pooled sample consists of all observations from both samples, ordered increasingly while retaining information about their sample of origin. For each $i \in \{1, \dots, 2n - 1\}$, let M_i denote the number of observations among $\{X_1, \dots, X_n\}$ that are less than or equal to $Z_{(i)}$.

The Anderson–Darling test statistic is defined as²

$$A^2 := \frac{1}{n^2} \sum_{i=1}^{2n-1} \frac{(2nM_i - ni)^2}{i(2n - i)}. \quad (3)$$

¹For simplicity, we assume that both samples have the same size n . This assumption is justified in the context of side-channel measurements, where it is typically feasible to collect an equal number of traces under different experimental conditions.

²The square in the notation is historical and reflects the fact that the statistic is a quadratic functional of the empirical process.

Under the null hypothesis and assuming continuity of the common distribution, the statistic A^2 converges in distribution, as $n \rightarrow \infty$, to a nondegenerate limiting distribution. In the two-sample case this limiting distribution can be expressed as [21]

$$A_\infty^2 := \sum_{j=1}^{\infty} \frac{1}{j(j+1)} W_j, \tag{4}$$

where $\{W_j\}_{j \geq 1}$ are independent chi-square random variables with one degree of freedom.

Since the limiting distribution does not admit a closed-form expression, thresholds corresponding to a prescribed significance level α are obtained from tabulated asymptotic percentiles or via numerical approximation. In particular, Scholz and Stephens [21] computed approximate percentiles by matching the first four moments of the limiting distribution and fitting a Pearson curve, following the methodology of Stephens [22] and Solomon and Stephens [23]. This approach has been shown to provide accurate approximations.

4 Anderson-Darling Leakage Assessment

This section introduces the ADLA framework for detecting secret-dependent leakage in neural network implementations. Subsection 4.1 formulates leakage detection as a statistical hypothesis test and connects it to TVLA based on Welch’s t -test (Subsection 3.2). Subsection 3.3 motivates the two-sample Anderson–Darling test as a distribution-sensitive alternative to mean-based methods, while Subsection 4.2 derives the ADLA threshold.

4.1 Leakage Detection as a Statistical Hypothesis Test

As discussed in Section 2, SCAs targeting neural networks aim to recover secret parameters by exploiting statistical dependencies between side-channel observations and the secret values. A leakage assessment method in this setting aims to determine whether such secret-dependent leakage is present.

To formalize leakage detection within the framework of statistical hypothesis testing, we model side-channel measurements as realizations of random variables. Under the null hypothesis of no secret-dependent leakage, the distribution of the measured leakage should be independent of the secret value. Consequently, the leakage distributions corresponding to different secret-dependent conditions should be identical.

In practice, leakage detection is typically conducted by collecting two sets of measurements: one obtained under a fixed input and another obtained under a different fixed input.

In the context of neural networks, we consider a specific input neuron corresponding to the secret parameter under investigation. For example, when evaluating potential leakage associated with the first weight in the first hidden layer, the value of the corresponding input neuron is varied while all other inputs are kept constant. This setting reflects a realistic attack scenario in which an attacker controls a chosen input neuron in order to induce variations in the intermediate computation involving the secret weight, thereby potentially amplifying secret-dependent leakage.

Under the null hypothesis of no data-dependent leakage, a necessary condition is that the distributions of side-channel leakages collected under the two different fixed input configurations are identical. If a statistically significant difference between these distributions is observed, the null hypothesis is rejected, indicating the presence of data-dependent leakage.

The TVLA methodology evaluates leakage by testing equality of means using Welch’s t -test under an approximate normality assumption [16]. A significant difference in sample means implies a difference in the underlying distributions and thus indicates data-dependent leakage. However, failure to reject the null hypothesis does not imply the absence of leakage – it merely indicates that no statistically significant mean difference was detected.

The motivation for employing the two-sample Anderson–Darling test (AD test) in our setting is that it compares the entire distributions rather than only their means. This allows detection of more general forms of distributional differences, including variance or tail discrepancies, which may not be captured by mean-based tests.

4.2 Derivation of the ADLA Threshold

In standard TVLA practice, the detection threshold is set to $\tau_t := 4.5$ [7, 24]. That is, after computing the test statistic t (cf. Eq. 1), leakage is declared if

$$|t| > \tau_t = 4.5.$$

Under the asymptotic normal approximation (cf. Eq. 2), this threshold corresponds to a two-sided significance level of approximately

$$\alpha \approx 3.4 \times 10^{-6},$$

providing a highly conservative criterion for leakage detection [16].

To ensure direct comparability with the standard TVLA methodology, we adopt the same significance level α in our proposed *Anderson–Darling Leakage Assessment (ADLA)* framework.

As discussed in Subsection 3.3, the limiting distribution of the two-sample Anderson–Darling statistic does not admit a closed-form expression. Consequently, the corresponding critical value must be determined numerically. We denote the threshold for ADLA by τ_A , defined as the upper $(1 - \alpha)$ -quantile of the limiting distribution:

$$\Pr(A_\infty^2 > \tau_A) = \alpha \approx 3.4 \times 10^{-6}.$$

To approximate τ_A , we employ the Pearson curve fitting method [21]. Using the additivity and scaling properties of cumulants for independent random variables [25] and Eq. 4, the r th cumulant of A_∞^2 is given by

$$\kappa_r = 2^{r-1}(r-1)! \sum_{j=1}^{\infty} \frac{1}{(j(j+1))^r}.$$

The infinite series can be evaluated in closed form via partial fraction decomposition [26]. For $r = 1, 2, 3, 4$, we obtain

$$\sum_{j=1}^{\infty} \frac{1}{(j(j+1))^r} = \begin{cases} 1, & r = 1, \\ \frac{\pi^2}{3} - 3, & r = 2, \\ 10 - \pi^2, & r = 3, \\ \frac{\pi^4}{45} + \frac{10\pi^2}{3} - 35, & r = 4. \end{cases}$$

Hence, the first four cumulants of A_∞^2 are

$$\kappa_1 = 1, \quad \kappa_2 = \frac{2\pi^2}{3} - 6, \quad \kappa_3 = 80 - 8\pi^2, \quad \kappa_4 = \frac{16\pi^4}{15} + 160\pi^2 - 1680.$$

Using the standard relations between cumulants and central moments [27], the first four moments satisfy

$$\mu_1 = \kappa_1, \quad \mu_2 = \kappa_2, \quad \mu_3 = \kappa_3, \quad \mu_4 = \kappa_4 + 3\kappa_2^2.$$

The skewness and kurtosis are therefore given by [28]

$$\gamma_1 = \frac{\mu_3}{\mu_2^{3/2}}, \quad \gamma_2 = \frac{\mu_4}{\mu_2^2}.$$

μ_1, μ_2, γ_1 , and γ_2 uniquely determine a member of the Pearson system, which we use to approximate the limiting distribution underlying ADLA. The Pearson curve fitting is performed using the `PearsonDS` package in \mathbb{R} [29]. For $\alpha = 3.4 \times 10^{-6}$, the resulting ADLA threshold value is

$$\tau_A \approx 11.99.$$

5 Experimental Evaluation

In this section, we first describe the experimental setup in Subsection 5.1 and subsequently present and discuss the evaluation results in Subsection 5.2.

5.1 Experimental Setup

To evaluate the proposed ADLA framework, we trained a multilayer perceptron (MLP) on the MNIST handwritten digit dataset [30] and collected side-channel power measurements using the ChipWhisperer-Husky platform [31].

MNIST comprises grayscale images of handwritten digits (0–9), where each sample is represented as a 28×28 pixel array. The dataset contains 60,000 training samples and 10,000 test samples and is widely used as a benchmark for image classification. Prior to training and evaluation, pixel intensities were normalized to the range $[0, 1]$.

The evaluated MLP consists of an input layer, three fully connected hidden layers, and an output layer with 784, 256, 128, 64, and 10 neurons, respectively. The hidden layers employ rectified linear unit (ReLU) activations, and the output layer uses a softmax activation. All computations were performed in 32-bit floating-point arithmetic. The trained network achieves 97.03% classification accuracy on the MNIST test set.

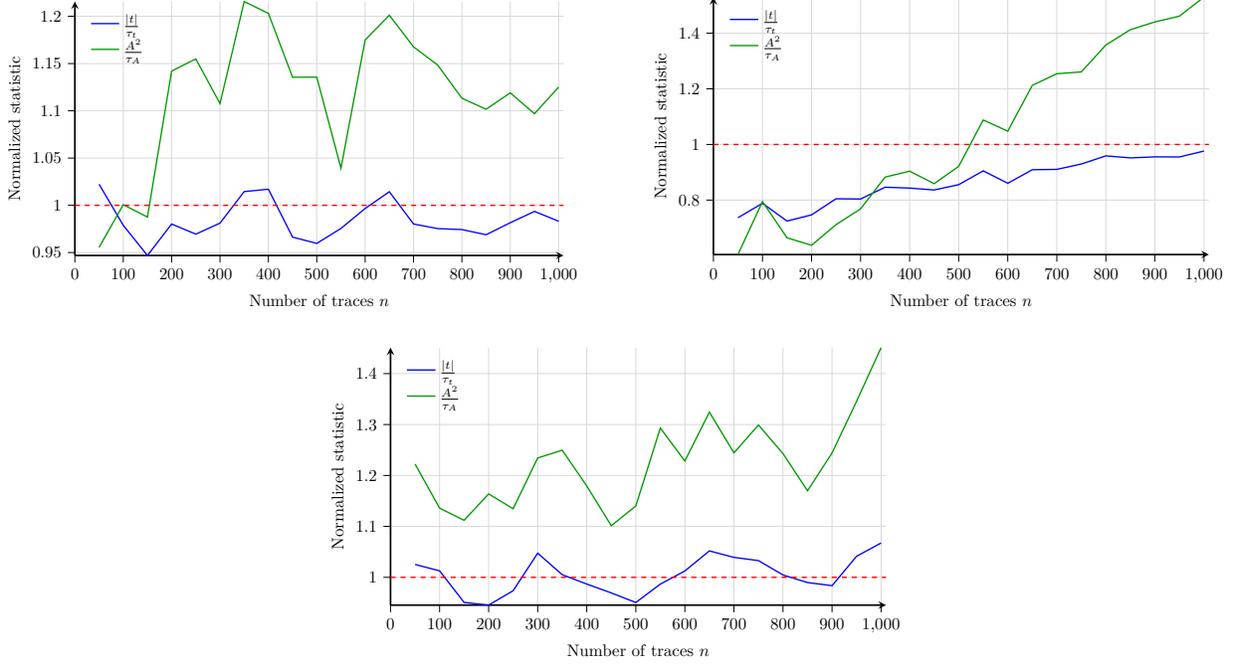


Figure 1: Normalized TVLA and ADLA statistics versus the number of traces n for three fixed input-value pairs in the protected implementation.

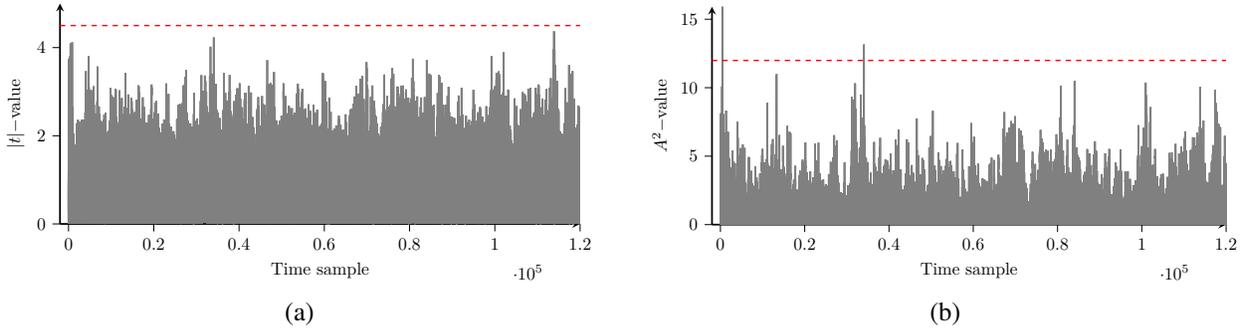


Figure 2: TVLA and ADLA statistics, (a) $|t|$ and (b) A^2 , evaluated at each time sample for $n = 850$ traces in the protected implementation.

Side-channel measurements were acquired using a ChipWhisperer-Husky capture device connected to a CW313 target board equipped with an Atmel SAM4S (ARM Cortex-M4) microcontroller as a device under test (DUT). The DUT was clocked at 7.3728 MHz, and the ADC sampling rate was set to $4\times$ that frequency. During inference, the device power consumption was recorded as time-series traces, each trace contains 120,000 samples.

The evaluated implementation combines the shuffling countermeasure proposed in [4] with random jitter to further desynchronize the measured traces. Shuffling randomizes the execution order of multiplications, while jitter inserts a pseudo-random delay immediately before each multiplication. The delay is implemented as a bounded busy-wait loop with a pseudo-random iteration count (e.g., `rand() & 127`) and is guarded against compiler optimization using a `volatile` sink and a memory barrier.

Following the leakage assessment methodology described in Subsection 4.1, we target the first network weight. Two sets of measurements were generated by applying two distinct values to the first input neuron, while keeping all remaining input neurons fixed across all captures. This procedure yields two sets of traces corresponding to two experimental conditions that induce different intermediate computations involving the targeted weight.

Leakage assessment was then performed independently at each time sample by comparing the empirical distributions of the two trace sets. Under the TVLA methodology, the null hypothesis assumes equal means for the leakage distributions associated with the two input conditions. In contrast, the proposed ADLA framework tests whether the two leakage distributions share the same cumulative distribution function (CDF).

Due to memory constraints on the target device, only the computation of the first hidden neuron was implemented and executed during the measurements. Since the targeted weight directly contributes to this computation through the associated multiply–accumulate operations, this restricted implementation remains representative for evaluating potential side-channel leakage. Leakage assessment was conducted on the recorded traces without additional preprocessing.

5.2 Evaluation Results

We performed leakage assessments for varying numbers of traces per experimental condition. Using the notation introduced in Subsections 3.2 and 3.3, let n denote the number of traces in each of the two trace sets. For each selected value of n , we collected two sets of n traces under two fixed values applied to the first input neuron, while keeping all remaining input neurons constant. Specifically, we evaluated three different pairs of fixed values for the first input neuron (all within $[0, 1]$), and fixed the remaining inputs to the pixel values of a randomly selected MNIST image.

For each time sample of the recorded traces, we compared the two trace sets by computing (i) the TVLA statistic, i.e., the absolute Welch t -statistic $|t|$ defined in Eq. (1), and (ii) the ADLA statistic, i.e., the two-sample Anderson–Darling statistic A^2 defined in Eq. (3). To facilitate a direct comparison between the two methodologies, we report threshold-normalized test statistics obtained by dividing each statistic by its corresponding detection threshold. Specifically, we plot $\frac{|t|}{\tau_t}$ and $\frac{A^2}{\tau_A}$, where $\tau_t = 4.5$ is used for TVLA and $\tau_A = 11.99$ is used for ADLA. With this normalization, values exceeding 1 indicate rejection of the null hypothesis (i.e., detectable leakage) at the significance level $\alpha = 3.4 \times 10^{-6}$ adopted throughout this work.

Figure 1 summarizes the normalized TVLA and ADLA results for the protected implementation. To further illustrate the difference between both tests, Figure 2 reports the test statistics for a representative input-pair experiment with $n = 850$ traces. In this instance, the TVLA statistic $|t|$ remains below the detection threshold across the trace, whereas the ADLA statistic exceeds its threshold at two time samples, indicating statistically significant leakage under ADLA but not under TVLA.

Overall, the experimental results indicate that ADLA is more sensitive in this setting, enabling leakage detection with fewer traces than TVLA. This difference is reflected not only in the detection outcome but also in the margin above the decision threshold: whereas $\frac{|t|}{\tau_t}$ typically remains close to 1 and exhibits only modest threshold exceedances, $\frac{A^2}{\tau_A}$ surpasses 1 by a substantially larger factor across the tested input pairs. These observations suggest that the leakage is not primarily characterized by a shift in the mean, but rather by broader distributional differences, which are captured by ADLA and may not be fully reflected by the mean-based TVLA statistic.

To assess whether the leakage samples follow a Gaussian distribution – an assumption adopted when interpreting Welch’s t -test (Subsection 3.2) in the TVLA methodology, we further employ a quantile–quantile (Q–Q) plot [32] with respect to the normal distribution. For a fixed time sample, the measured leakage values are sorted to obtain empirical quantiles and plotted against the corresponding theoretical quantiles of a standard normal distribution. If the leakage were normally distributed (up to an affine transformation), the plotted points would lie approximately on a straight line (the normal reference line).

Figure 3 shows the Q–Q plot constructed from a dataset of $n = 1000$ traces at time sample $t = 1316$, using measurements collected under a single fixed input configuration. The time sample $t = 1316$ corresponds to the highest peak observed in Fig. 2 (b). The pronounced deviation of the empirical quantiles from the normal reference line confirms that the leakage distribution at this time sample is not Gaussian.

6 Conclusion and Future Work

We introduced and evaluated the ADLA framework as a distribution-sensitive alternative to TVLA. For the shuffling- and jitter-protected implementation considered in this work, ADLA consistently detected leakage at relatively low trace counts, including cases where TVLA remained below its detection threshold. These results suggest that, in this setting, leakage is not necessarily dominated by mean shifts, but can instead arise from broader distributional differences that are captured by ADLA.

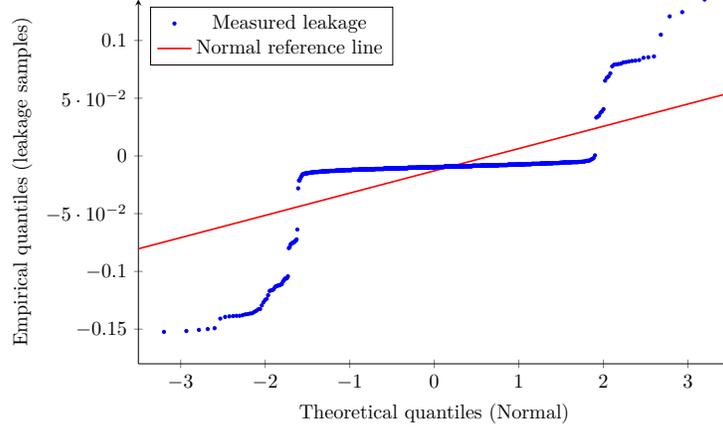


Figure 3: Q–Q plot of leakage samples at time sample $t = 1316$, obtained from a dataset of $n = 1000$ traces corresponding to a fixed input value.

χ^2 -based leakage tests. In addition to mean-based t -tests and distribution-based tests such as ADLA, side-channel leakage can also be assessed using χ^2 -type tests, as discussed in [17]. In our setting, however, we observed that χ^2 -based results are highly sensitive to the discretization of the measurement space, i.e., to the choice of the number of bins and bin boundaries used to form the contingency table. Designing a robust χ^2 -based test for this setting—including principled binning strategies and stability analyses across noise regimes—is therefore an interesting direction for future work.

Higher-order power analysis. As another future work, we will investigate whether the leakage points detected by ADLA can be exploited to reveal secret weights. While CPA is ineffective against shuffling [4], the leakage detected by ADLA motivates investigating stronger adversaries. Future work will therefore consider higher-order attacks and statistical analyses [33] to assess exploitability.

References

- [1] Lejla Batina, Shivam Bhasin, Jakub Breier, Xiaolu Hou, and Dirmanto Jap. On implementation-level security of edge-based machine learning models. In *Security and Artificial Intelligence: A Crossdisciplinary Approach*, pages 335–359. Springer, 2022.
- [2] Lejla Batina, Shivam Bhasin, Dirmanto Jap, and Stjepan Picek. {CSI}{NN}: Reverse engineering of neural network architectures through electromagnetic side channel. In *28th USENIX Security Symposium (USENIX Security 19)*, pages 515–532, 2019.
- [3] Anuj Dubey, Afzal Ahmad, Muhammad Adeel Pasha, Rosario Cammarota, and Aydin Aysu. Modulonet: Neural networks meet modular arithmetic for efficient hardware masking. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pages 506–556, 2022.
- [4] Leonard Puškáč, Marek Benovič, Jakub Breier, and Xiaolu Hou. Make shuffling great again: A side-channel-resistant fisher–yates algorithm for protecting neural networks. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2025.
- [5] Jakub Breier, Dirmanto Jap, Xiaolu Hou, and Shivam Bhasin. A desynchronization-based countermeasure against side-channel analysis of neural networks. In *International Symposium on Cyber Security, Cryptology, and Machine Learning*, pages 296–306. Springer, 2023.
- [6] Information technology – security techniques – testing methods for the mitigation of non-invasive attack classes against cryptographic modules, 2016.
- [7] Benjamin Jun Gilbert Goodwill, Josh Jaffe, Pankaj Rohatgi, et al. A testing methodology for side-channel resistance validation. In *NIST non-invasive attack testing workshop*, volume 7, pages 115–136, 2011.
- [8] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.

- [9] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.
- [10] Lejla Batina, Shivam Bhasin, Dirmanto Jap, and Stjepan Picek. Poster: Recovering the input of neural networks via single shot side-channel attacks. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pages 2657–2659, 2019.
- [11] Xiaobei Yan, Xiaoxuan Lou, Guowen Xu, Han Qiu, Shangwei Guo, Chip Hong Chang, and Tianwei Zhang. Mercury: An automated remote side-channel attack to nvidia deep learning accelerator. *arXiv preprint arXiv:2308.01193*, 2023.
- [12] Zdenko Lehocký, Jakub Breier, Dirmanto Jap, Shivam Bhasin, and Xiaolu Hou. Side-channel analysis of openvino-based neural network models. In *International Conference on Availability, Reliability and Security*, pages 307–324. Springer, 2025.
- [13] Yusuke Nozaki and Masaya Yoshikawa. Shuffling countermeasure against power side-channel attack for mlp with software implementation. In *2021 IEEE 4th International Conference on Electronics and Communication Engineering (ICECE)*, pages 39–42. IEEE, 2021.
- [14] Manuel Brosch, Matthias Probst, and Georg Sigl. Counteract side-channel analysis of neural networks by shuffling. In *2022 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 1305–1310. IEEE, 2022.
- [15] Jean-Sébastien Coron and Ilya Kizhvatov. An efficient method for random delay generation in embedded software. In *Cryptographic Hardware and Embedded Systems-CHES 2009: 11th International Workshop Lausanne, Switzerland, September 6-9, 2009 Proceedings*, pages 156–170. Springer, 2009.
- [16] Xiaolu Hou and Jakub Breier. *Cryptography and Embedded Systems Security*. Springer, 2024.
- [17] Amir Moradi, Bastian Richter, Tobias Schneider, and François-Xavier Standaert. Leakage detection with the χ^2 -test. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2018(1):209–237, 2018.
- [18] Sheldon M Ross. *Introduction to probability and statistics for engineers and scientists*. Academic press, 2020.
- [19] Bernard L Welch. The generalization of ‘student’s’ problem when several different population variances are involved. *Biometrika*, 34(1-2):28–35, 1947.
- [20] AN Pettitt. A two-sample anderson–darling rank statistic. *Biometrika*, pages 161–168, 1976.
- [21] F Scholz and M Stephens. K-sample anderson-darling tests of fit, for continuous and discrete cases. *University of Washington, Technical Report*, (81), 1986.
- [22] Michael A Stephens. Asymptotic results for goodness-of-fit statistics with unknown parameters. *The annals of statistics*, pages 357–369, 1976.
- [23] Herbert Solomon and Michael A Stephens. Approximations to density functions using pearson curves. *Journal of the American Statistical Association*, 73(361):153–160, 1978.
- [24] A Adam Ding, Liwei Zhang, François Durvaux, François-Xavier Standaert, and Yunsi Fei. Towards sound and optimal leakage detection procedure. In *Smart Card Research and Advanced Applications: 16th International Conference, CARDIS 2017, Lugano, Switzerland, November 13–15, 2017, Revised Selected Papers*, pages 105–122. Springer, 2018.
- [25] John E Kolassa. *Series approximation methods in statistics*. Springer, 2006.
- [26] M. Hata. *Problems and Solutions in Real Analysis*. Series on number theory and its applications. World Scientific, 2007.
- [27] Ron C Mittelhammer and Ron C Mittelhammer. *Mathematical statistics for economics and business*. Springer, 2013.
- [28] Japuji K Sharma. *Business statistics*. Pearson Education India, 2012.
- [29] Martin Becker and Stefan Klöbner. *PearsonDS: Pearson Distribution System*, 2025. R package version 1.3.2.
- [30] Li Deng. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE signal processing magazine*, 29(6):141–142, 2012.
- [31] NewAE Technology Inc. *ChipWhisperer-Husky and HuskyPlus User Manual*. NewAE Technology Inc., 2025.
- [32] John M Chambers. *Graphical methods for data analysis*. Chapman and Hall/CRC, 2018.
- [33] Emmanuel Prouff, Matthieu Rivain, and Régis Bevan. Statistical analysis of second order differential power analysis. *IEEE Transactions on computers*, 58(6):799–811, 2009.