A SLR of Modern AI-Driven SCA on NIST's PQC Standards

Edgard Nicéas Arcoverde Gusmão Lima[1,2][0000−0002−5241−8092], Fábio Wlademir Monteiro Maia[1,2], Tiago Alessandro Espínola Ferreira[3], and Danilo Monteiro Ribeiro[2]

[1] CISSA, CESAR, Recife, Brazil
enagl@cesar.org.br, fwmm@cesar.org.br
[2] CESAR School, Recife, Brazil
enagl@cesar.edu.br, fwmm@cesar.edu.br, dmr@cesar.school
[3] PPGIA, Universidade Federal Rural de Pernambuco (UFRPE), Recife, Brazil
tiago.espinola@ufrpe.br

Abstract. The imminent threat of quantum computing has catalyzed a global migration to Post-Quantum Cryptography (PQC), guided by the U.S. National Institute of Standards and Technology (NIST) standardization process. While these new cryptographic algorithms are designed for mathematical resilience against quantum adversaries, their physical implementations expose a critical vulnerability to Side-Channel Attacks (SCAs). This threat is fundamentally amplified by Artificial Intelligence (AI), which has transformed the adversary model from one requiring statistical analysis of numerous device interactions to one capable of potent, often single-trace, key extraction. This paper presents a systematic literature review (SLR) mapping the collision of AI-driven SCAs with the first cohort of NIST-selected PQC standards: CRYSTALS-Kyber, CRYSTALS-Dilithium, Falcon, SPHINCS+, and the prominent fourth-round candidate HQC. From the synthesized literature, we develop a comprehensive taxonomy that classifies attacks by linking specific AI methodologies to the unique vulnerabilities in each cryptographic family, such as the repetitive arithmetic in lattice-based schemes. Our analysis reveals three prevailing trends: the ascendancy of single-trace attacks that make in-field exploitation practical, the systematic neutralization of established countermeasures like masking by deep learning models, and the efficacy of hybrid attacks combining machine learning with algebraic cryptanalysis.

Keywords: Post-Quantum Cryptography · PQC · Side-Channel Analysis · SCA · Artificial Intelligence · AI · Machine Learning · Deep Learning · NIST PQC Standardization · ML-KEM · CRYSTALS-Kyber · ML-DSA · CRYSTALS-Dilithium · FN-DSA · Falcon · SLH-DSA · SPHINCS+ · HQC.

1 Introduction: The Collision of Post-Quantum Cryptography and AI-Powered Cryptanalysis

1.1 The Quantum Imperative and the NIST PQC Standardization

Modern digital infrastructure relies on public-key cryptography (PKC) schemes such as RSA and Elliptic Curve Cryptography (ECC), whose security stems from problems that are classically hard to solve [88]. Large-scale quantum computers running Shor's algorithm threaten this foundation by solving those problems in polynomial time [89]. To anticipate that "quantum day," the U.S. National Institute of Standards and Technology (NIST) launched an open competition to standardize quantum-resistant algorithms [64]. In 2022, NIST announced CRYSTALS-Kyber for public-key encryption and CRYSTALS-Dilithium for signatures, adding FALCON and SPHINCS+ as alternatives, while the code-based HQC remains under evaluation in the fourth round [3,30,4]. Table 22 summarizes the algorithms targeted in this review.

1.2 The Achilles' Heel: Implementation Security and the Side-Channel Threat

Mathematical soundness alone does not guarantee security. PQC schemes must run on physical devices that leak information via side channels such as timing, power consumption, and electromagnetic emanations [53,17]. These leakages let adversaries bypass theoretical guarantees by recovering keys from real deployments [80]. NIST therefore required side-channel resilience throughout the PQC evaluation process [1], effectively crowdsourcing a global red-team exercise. Discoveries such as the FALCON-targeting attacks [46] were treated as valuable feedback that shaped countermeasures, while several proposals in Table 23 were eliminated after catastrophic implementation-level breaks. PQC's novel arithmetic widens the attack surface, making practical defenses a primary concern.

1.3 The Paradigm Shift: Artificial Intelligence as a Force Multiplier

Side-channel analysis itself is being transformed by Artificial Intelligence (AI). Profiled attacks can be cast as supervised learning where traces are features and secret-dependent intermediates act as labels [20]. Deep learning—especially MLP- and CNN-based classifiers—learns leakage patterns directly from raw traces, eliminating much of the manual feature engineering required by classical statistics [57]. These models regularly defeat masking, shuffling, and other countermeasures by exploiting residual higher-order dependencies [73]. The integration of AI therefore lowers the barrier to executing high-efficacy attacks and must be assumed in any realistic threat model [80,39].

1.4 Problem Statement, Objectives, and Research Questions

PQC standardization, physical implementation risks, and AI-accelerated cryptanalysis intersect to form an urgent security gap. Although NIST is publishing

quantum-resistant algorithms, AI-equipped adversaries may compromise their implementations on day one. This systematic literature review (SLR) maps AI-driven side-channel attacks against CRYSTALS-Kyber, CRYSTALS-Dilithium, FALCON, SPHINCS+, and HQC. We focus on peer-reviewed or IACR ePrint-style work describing practical attacks that leverage Machine Learning, Deep Learning, or related modern AI techniques. Our objectives are to (1) catalogue the state of the art, (2) build a taxonomy that links AI vectors to cryptographic vulnerabilities, and (3) highlight trends, gaps, and open challenges to guide future defenses.

- **RQ1:** What peer-reviewed or relevant pre-print literature documents practical AI-based side-channel attacks on the NIST PQC standards and HQC?
- **RQ2:** How can those attacks be categorized to map AI techniques to the targeted cryptographic operations and leakage sources?
- **RQ3:** Which attack trends, research gaps, and challenges emerge from the synthesized data?

1.5 Structure of the Review

Section 2 provides background. Sections 3-4 follow the Kitchenham protocol [50,51] to detail the methodology and study selection. Sections 5 review related surveys. Section 6 report and synthesize the results with our taxonomy and analysis. Section 7 concludes and outlines future work.

2 Theoretical Background: The Convergence of AI, Side-Channel Analysis, and Post-Quantum Cryptography

2.1 The Quantum Threat to Modern Cryptography

Large-scale quantum computers pose a direct threat to modern public-key cryptography because Shor's algorithm can factor RSA moduli and solve elliptic-curve discrete logarithms in polynomial time [88,89,76,32], while Grover's algorithm quadratically accelerates brute-force search on symmetric primitives [33]. In contrast to symmetric schemes—which can be repaired by doubling key sizes—public-key systems such as RSA, Diffie-Hellman and ECC offer no viable parameter scale-up once a cryptanalytically relevant quantum computer becomes available. This creates an urgent need for post-quantum cryptography (PQC) [64], particularly for protecting long-term confidentiality against "harvest now, decrypt later" adversaries who can store encrypted data today and decrypt it once quantum capabilities mature [21,97,59]. More broadly, quantum algorithms fundamentally challenge assumptions underlying classical complexity theory [24,74,14,62], reinforcing the necessity of transitioning to cryptographic schemes whose security does not collapse under quantum computation.

2.2 Post-Quantum Cryptography and NIST's Standardization Effort

Post-quantum cryptography (PQC) aims to provide cryptographic primitives secure against both classical and quantum adversaries [14,105]. Unlike quantum cryptography, PQC algorithms are classical and designed to run efficiently on existing processors [62,24]. Their security is based on mathematical problems for which no quantum algorithm is known to provide exponential speedups, in contrast to the integer factorization and discrete logarithm problems broken by Shor's algorithm. The main families of PQC include lattice-based schemes grounded in problems such as SVP, LWE and Module-LWE [9,10,3,67,80]; code-based ([72,71]) cryptography originating with McEliece's 1978 system [61]; hash-based signatures built from minimal and well-understood assumptions [8,14]; as well as multivariate and isogeny-based constructions [42,84]. These families offer complementary trade-offs in security reductions, key sizes and implementation characteristics, and collectively form the foundation of quantum-resistant cryptography.

To orchestrate a global and transparent transition to PQC, NIST launched its public standardization process in 2016 through an open call for proposals [64,63,92]. From 69 complete submissions entering Round 1 [2], candidates were evaluated under criteria encompassing security (including resistance to classical, quantum and side-channel attacks) [66,1], cost and performance across platforms [65], and practical considerations such as design simplicity, IP status and implementability. After multiple competitive rounds, NIST selected a diverse portfolio for standardization: ML-KEM/CRYSTALS-Kyber for key establishment [68,69,9]; ML-DSA/CRYSTALS-Dilithium as the primary signature scheme [67,69,10]; FN-DSA/Falcon for compact signatures based on NTRU lattices [3,72,29,46]; and SLH-DSA/SPHINCS+ as a conservative, stateless hash-based alternative [70,69,8]. In March 2025, NIST further expanded the portfolio by selecting HQC as a code-based KEM [72,71,75,30],adding a non-lattice standard whose security relies on the hardness of quasi-cyclic syndrome decoding and whose design offers well-analyzed decryption-failure behavior and long-studied code-based security. These five algorithms—Kyber, Dilithium, Falcon, SPHINCS+, and HQC—constitute the standardized suite that will secure next-generation communication systems in the presence of quantum-capable adversaries.

2.3 Fundamentals of Physical Side-Channel Attacks (SCA)

While the theoretical security of a cryptographic primitive rests on its mathematical hardness, its practical security depends on the behavior of its physical implementation. Executing an algorithm on real hardware inevitably produces side effects—power consumption, electromagnetic (EM) radiation, and timing variations—that correlate with the secret-dependent internal state [106,53]. Side-channel attacks (SCA) exploit these leakages to recover long-term keys, often completely bypassing the underlying cryptographic assumptions.

Non-profiled attacks operate directly on the target device and typically require many traces to amplify weak leakage. Simple Power Analysis (SPA) inspects individual traces to reveal coarse-grained patterns such as square-and-multiply sequences in classical RSA implementations [52,53]. Differential and Correlation Power Analysis (DPA/CPA) use statistical techniques: DPA partitions traces according to a key hypothesis and observes a differential spike when the hypothesis is correct [53], whereas CPA correlates hypothetical power models (e.g., Hamming weight or Hamming distance) with measured leakage to identify the correct key [17,54].

Profiled attacks assume a worst-case adversary with access to an identical device. The attack proceeds in two phases [20]: during profiling, the attacker characterizes the device's leakage by collecting traces under known keys and constructing a statistical model; during the attack phase, only a few traces from the victim are required to identify the secret key. Template Attacks, which model leakage using multivariate Gaussian distributions, exemplify this paradigm and can succeed with a single trace when profiling is accurate [20,109].

Profiled AI-based attacks extends classical template ones, modern profiled attacks increasingly rely on machine learning (ML) and deep learning (DL) techniques to learn complex, non-linear leakage distributions. Early works showed that classical ML models such as Random Forests and SVMs can outperform Gaussian templates when leakage deviates from simple statistical assumptions [57]. More recently, deep neural networks—including MLPs and especially CNNs—have demonstrated remarkable robustness to noise, desynchronization and higher-order effects [12,60]. These AI-based profiled attacks follow the same two-phase structure as template attacks but replace parametric Gaussian models with learned discriminative models, enabling key recovery with significantly fewer traces and widening the applicability of profiled SCA to protected implementations.

The AI models applied to SCA form a broad taxonomy (as detailed in Table 1). This includes classical baselines like RF [55,37,12]; the dominant deep-/neural models, such as MLPs for automatic representation learning [12,57], state-of-the-art CNNs [18,60,49], and RNNs for temporal dependencies [39,44]; and reinforcement/decision models (e.g., DQNs), which are typically used to optimize attack parameters rather than for classification itself [39].

2.4 Side-Channel Analysis Countermeasures

Countermeasures against SCA operate at either the software or hardware level and aim to suppress exploitable leakage by (a) removing secret-dependent behavior or (b) randomizing it so that leakage becomes statistically unusable. However, recent ML-based profiled attacks have demonstrated the ability to recover higher-order information, undo shuffling, and exploit microarchitectural artefacts that classical protections fail to eliminate.

Software-Level Countermeasures are widely adopted in PQC libraries due to their portability and low deployment cost. Masking (or sharing) removes

Table 1. Modern AI-based models applied in SCAs against PQC algorithms, grouped by family.

Classical	**Random Forest** Ensemble of decision trees; robust, handles nonlinearities and mixed features.
Reinforcement / Decision	**Deep Q-Network (DQN)** Value-based RL: approximates $Q(s, a)$ with a deep net and experience replay.
Deep / Neural	**Convolutional Neural Network (CNN)** Learns local patterns via convolutions; strong on spatial/structured signals.
	Recurrent Neural Network (RNN) Captures temporal dependencies with hidden state across timesteps.
	Multi-Layer Perceptron (MLP) Feed-forward dense layers; general nonlinear baseline.
	Graph Neural Network (GNN) Message-passing on graphs; aggregates neighborhood information.
	Large Language Model (LLM) Transformer-based sequence model; self-attention for long-range context.
	Neural Network (NN) Generic neural architecture (unspecified); usually feed-forward baseline.
	Adaptive Slimmed Pyramid Network (ASPN) Lightweight *CNN variant*; pyramid design with slimmed channels for multi-scale feature extraction.

first-order statistical dependencies by splitting a sensitive variable into random shares [73,83], while shuffling randomizes the execution order of independent operations to induce temporal desynchronization [73,99]. Constant-time implementations suppress timing and cache-based leakages by eliminating secret-dependent branches or memory accesses [52,13,5], and blinding randomizes intermediate values to decorrelate successive executions [11,47].

Hardware Countermeasures aim to decorrelate or suppress physical leakage at the circuit, FPGA, or microarchitectural level and often provide stronger protection than their software counterparts. Modern gate-level techniques focus on balanced or leakage-equalizing logic styles and bit-level netlist augmentation to reduce data-dependent switching, with recent evaluations outlining their security-efficiency trade-offs [94,6]. Another line of work introduces on-chip hardware noise, such as low-cost correlated-power noise generators (CPNGs), which inject controlled noise into the power distribution network to reduce SNR and hinder CPA-style attacks [95]. Desynchronization-based approaches remain relevant, where randomized clocking and jitter disrupt trace alignment and mitigate both classical and DL-based SCAs on FPGA and ASIC platforms [85]. For PQC, hardware-friendly shuffling architectures—such as optimized Fisher–Yates variants for Kyber—provide practical execution-order randomization with minimal area and performance cost [107]. Finally, adaptive mitigation techniques leverage Dynamic Partial Reconfiguration (DPR) on FPGAs, enabling runtime reconfiguration combined with DL-based monitoring to disrupt the attacker's profiling assumptions and provide proactive protection [16].

2.5 Common SCA metrics

The Guess Entropy, Success Rate
TODO TODO

3 Review Methodology

We followed the Kitchenham and Charters SLR protocol [50,51]. We queried seven databases (IEEE Xplore, ACM DL, Scopus, SpringerLink, Engineering Village, ScienceDirect, IACR ePrint) up to August 2025 using a PICOC-derived search string targeting AI-based SCAs on NIST PQC algorithms. Full methodology details, including search strings, inclusion/exclusion criteria, quality checklist, and data extraction form, are provided in Appendix A.

4 Study Selection Overview

We identified 1,495 records across seven databases, reduced to 1,338 unique entries after de-duplication, and selected 27 primary studies following our inclusion and exclusion criteria. A high-level summary of quality assessment outcomes and full search statistics are provided in Appendix B. The synthesized findings from these studies are presented in Section 6.

5 Related Works

Other surveys and overview papers have been published in recent years that discuss side-channel and fault-injection threats to post-quantum cryptography (PQC). These related works provide useful background by classifying attack methodologies, highlighting vulnerable components of lattice-based schemes, and outlining known countermeasures. In particular, they help situate our systematic review within the broader research landscape by showing how the community has so far organized knowledge about PQC vulnerabilities, before the rapid expansion of AI-driven side-channel analysis in the last two years.

Hernández-Álvarez et al. [35] (2023) conducted a survey-style review of recent AI-based side-channel attacks targeting the NIST-selected PQC standards CRYSTALS-Kyber and CRYSTALS-Dilithium. Their work categorized attacks by the algorithmic function exploited (e.g., encoding, NTT, re-encryption) and analyzed the types of machine learning models employed, with a strong emphasis on multilayer perceptrons (MLPs) and clustering techniques such as k-means. In addition to cataloging attack scenarios and success rates, they discussed hyperparameters, regularization strategies, and limitations of current approaches, arguing that research should extend beyond MLPs toward CNNs and RNNs for better efficiency and generalization. As such, this publication provides a broad overview of AI-driven SCA methodologies but lacks the systematic methodology of an SLR, complementing our study by situating our results within the broader landscape of PQC-focused AI-based SCA research.

Li [56] (2023) provided an overview of attacks against CRYSTALS-Kyber, covering both classical cryptanalytic strategies and physical side-channel threats. The survey structured its discussion around four categories: common attacks (including module-LWE parameter considerations and decryption failures), side-channel attacks (timing, SASCA, message encoding, deep learning, and LDPC-based frameworks), side-channel assisted chosen-ciphertext attacks (e.g., EM-based plaintext-checking and CPA), and fault injection techniques (Roulette and error-tolerant recovery). Although written as an overview rather than a systematic review, the work offers a broad categorization of known vulnerabilities and highlights the practicality of advanced side-channel and fault-based approaches even against masked or optimized implementations. In contrast to our SLR, which systematically synthesizes AI-driven SCA results across PQC algorithms, Li's review provides a Kyber-focused narrative that helps contextualize attack feasibility and evolving defense requirements.

Ravi et al. [80] (2024) presented a comprehensive survey of side-channel and fault-injection attacks on lattice-based PQC schemes, focusing primarily on Kyber and Dilithium. Unlike prior overviews, this work systematically categorized known attacks by target procedure (key generation, encapsulation, decapsulation, signing) and attack vector (power, EM, glitches), while also proposing a unified characterization framework based on attacker capabilities, profiling requirements, number of traces, and SNR assumptions. Beyond surveying, they contributed new countermeasures, including customized protections that combine shuffling, masking, and novel defenses against chosen-ciphertext-based SCA/-FIA, benchmarked on embedded platforms such as ARM Cortex-M4 (pqm4) and Cortex-A53 (liboqs). However, their review does not emphasize modern AI-based techniques and is limited to Kyber and Dilithium, whereas our SLR systematically synthesizes AI-driven SCA results across a broader set of PQC algorithms. This distinction highlights how our work complements theirs by extending the perspective from general SCA/FIA to the specific and growing role of machine learning in PQC side-channel analysis.

While these surveys provide valuable insights, they also exhibit limitations when compared to a systematic literature review. Some, such as Hernández-Álvarez et al. [35], adopt a survey-style approach without a reproducible SLR methodology; others, like Li [56], restrict their focus to CRYSTALS-Kyber; and Ravi et al. [80] broaden the scope to Kyber and Dilithium but do not emphasize modern AI-based attack techniques. Moreover, most of these reviews were published in 2023 or early 2024, which makes them rapidly outdated: as shown in Figure 1, the number of AI-based SCA publications on PQC has nearly doubled each year from 2023 to 2025. This underscores both the fast pace of research in this field and the need for our systematic review, which synthesizes the latest AI-driven SCA advances across multiple PQC algorithms and protected implementations.

6 Synthesis of Key Findings

6.1 Attack Surface and Leakage Vectors in Kyber

The attack surface of Kyber, particularly concerning leakage vectors exploited by Side-Channel Attacks (SCAs), is broad, encompassing physical measurement channels, specific cryptographic operations, and residual vulnerabilities within countermeasures. See Tables 2 and 3.

Table 2. Channels, Leakage Models and Countermeasures Weakness in **Kyber**

Physical Channels and Measured Leakages
Power Consumption (PA): Dynamic power consumption is a primary source of leakage, widely exploited, particularly in implementations on ARM Cortex-M4 microcontrollers [22,31,44,81,82,100,111,38].
Electromagnetic (EM) Emanation: EM radiation is a prominent leakage vector [22,44]. EM attacks based on Deep Learning (DL) remain viable even when the microcontroller is protected by an **opaque anti-tampering cover**, despite signal attenuation [22].
Timing: Execution time leakage can be exploited, notably in relation to the integrity check in the Fujisaki-Okamoto (FO) transformation [44,98].
Leakage Models
Hamming Weight (HW): A common model used, for instance, in deep learning classifiers to recover the secret key coefficients during pointwise multiplication [36,43,81,82,103,38].
Hamming Distance (HD): This model quantifies the change between successive operations or shares. In hardware implementations, the HD between adjacent bits of polynomial shares ($m1/m2$) exhibits **strong leakage** due to the high fanout of internal registers (e.g., $m1\_reg/m2\_reg$) when bit values change [44].
Joint Distribution: Used in advanced blind SCAs, where the classifier exploits the joint distribution of the Hamming Weights of relevant intermediate values [81,82]. Second-order attacks specifically analyze joint information leakage in masked hash functions [103].
Vulnerability of Countermeasures
Masking Defeat: Masked implementations, including those that are first-order protected, have been successfully broken via DL-based SCAs [31,44,100,38]. Furthermore, advanced DL techniques, such as the recursive learning/copy-paste method, have demonstrated message recovery success against Kyber implementations masked up to **fifth-order** ($\omega \leq 5$) [25,22].
Hiding Defeat: The shuffling countermeasure can be bypassed. Voltage fault injection has been demonstrated as a method to **disable shuffling** in a masked implementation, enabling a single-trace message recovery attack by eliminating the temporal randomness introduced by shuffling [43]. Similarly, **standalone temporal hiding countermeasures** like Random Delays and Clock Jitter show limited efficacy against DL-based attacks [38]. However, combining shuffling with masking effectively resists recently proposed second-order attacks targeting the masked hash function [103].

6.2 Attack Surface and Leakage Vectors in Dilithium

The attack surface of Dilithium is extensive, focusing primarily on operations involving the secret key components and intermediate random values. Attacks target both the signature generation and key generation procedures. See Tables 4 and 5.

Table 3. Key Algorithmic Attack Points in **Kyber**

Plaintext Checking (PC) Oracles via FO Transformation
PRF/Hash Function Execution: The leakage arises during the execution of the Pseudo-Random Function (PRF) or Hash function (e.g., **SHAKE** or **SHA3/Keccak**) when the decrypted message (m') is re-encrypted (ct') to check equality with the received ciphertext (ct) [93,98,103,38].
Masked Hash Leakage: In first-order masked Kyber implementations, second-order attacks specifically exploit the joint information leakage in the masked hash function (SHA3-512 based Keccak) [103]. This leakage can be used to distinguish decrypted messages differing by **1 bit** or, through specialized ciphertext construction, by **32 bits** [103,93].
Multiple-Valued Oracles: The leakage can be utilized to create a Multiple-Valued PC (MV-PC) oracle, generalizing the binary oracle to extract more secret key information per trace, significantly reducing the total required traces [93].
Message Decoding and Encoding
Masked Decoding (`poly_to_msg`): This function, which converts the shared key from the polynomial domain back to the message domain, is a demonstrated vulnerability in masked implementations [31,44,38]. The attack exploits leakage found in the **masked message decoding function** during the decapsulation decryption step [44,38].
Message Encoding (`masked_poly_frommsg`): Attacks also target the masked message encoding procedure carried out during the re-encryption step [43,100,38]. This leakage, often referred to as "direct-copy leakage," allows message recovery [43].
Core Arithmetic Operations
Pointwise Multiplication: The computation of $w^* = \text{NTT}(u') \circ \text{NTT}(\hat{s})$ (the multiplication of the ciphertext coefficients with the secret key coefficients) is a critical leakage target [36,81]. Attacks target the individual coefficient multiplications (e.g., five $fqmul$ operations per $basemul$) where register loading and storing result in leakage of input and output Hamming Weights [36,81].
Modular Reduction: The modular reduction function (e.g., Barrett reduction) is vulnerable to chosen-ciphertext clustering attacks [31]. It is possible to profile the leakage of reductions performed on controllable data (R1) and transfer this model to successfully attack reductions involving secret key material (R2/R3) without requiring a separate reference device [108].

6.3 Attack Surface and Leakage Vectors in Falcon

Falcon, an NTRU-based digital signature, and its enhanced variants (such as Mitaka, Antrag, and SOLMAE) employ **Cumulative Distribution Table (CDT) sampling** to quickly generate arrays of random values from a discrete Gaussian distribution during the signature generation phase [23]. The attack surface centers on the leakage of these extracted random values, which, when used with secret key information, contributes to the **indirect leakage of secret information** [23]. **Single Trace Analysis (STA)** is successfully employed to recover the sampled values by exploiting comparison operations in various environments [23]. See Tables 6 and 7.

6.4 Attack Surfaces in Hash-based SPHINCS+

Although SPHINCS+ is stateless and hash-based, it is not inherently immune to side-channel leakage. However, at the time of this review, we found no published work that systematically studies SPHINCS+ under modern AI-assisted side-channel analysis. Consequently, no concrete AI-driven attack surfaces or leakage

Table 4. Channels, Leakage Models, and Countermeasures Weakness in **Dilithium**

Physical Channels and Measured Leakages
Power Consumption (PA): Power consumption is the primary channel used in profiling attacks, with experiments conducted successfully on ARM Cortex-M4 microcontrollers (e.g., ChipWhisperer STM32F3 or STM32F4 boards) [34,48,58,78,102,79]. The single-trace attack aims to recover the entire key from one power trace, regardless of optimization level (-O0, -O3, -Os) [34,48].
Timing: Execution time is noted as a side channel, although constant-time implementation aims to mitigate this risk [34,58].
Leakage Models and Profiles
Hamming Weight (HW): Used as the assumed power consumption model in software implementations [34,102]. The Hamming Weight difference of intermediate values (e.g., in mont_reduce) determines the efficiency of distinguishing secret coefficients [34]. Also used in correlation analysis for the temporary values r and y in polyz_unpack [78].
LSB Leakage: The Least Significant Bit (LSB) of the t0 coefficients is targeted specifically using t-test analysis, showing distinct leakage patterns [101].
Zero/Non-Zero Distinction: Crucial for the bit-unpacking attack on the y vector, where classifiers predict whether a coefficient is zero or non-zero, exploiting leakage visible via t-test [58].
Bit Leakage (Linear Regression Model): Used in profiled attacks on the y vector, where linear regression is employed to account for varying leakage weights of individual bits [78].
Vulnerability of Countermeasures
Masking Defeat: Masked Dilithium implementations have been shown vulnerable to single-trace attacks targeting the multiplication operation, requiring a minimal number of traces for key exposure [48]. Applying masking to the signing procedure eliminates the advantage of NTT attacks, as input coefficients become uniformly distributed [34]. However, masking introduces a high performance overhead (estimated at around five times slower) [58].
Determinism/Randomization: The attack targeting the y generation via bit-unpacking applies to both the deterministic and non-deterministic (probabilistic) signing variants, as the target step (bit-fiddling) is common to both [58]. Single-trace attacks are effective even against randomized implementations because they extract information (y) per signature execution [58].
Key Compression (t0): The public key compression, which omits the low-order bits t0 of vector t, is intended for performance optimization, not security [102]. However, knowledge of t0 significantly enhances the efficiency of post-processing techniques used to recover s1 and s2 [102].

vectors specific to SPHINCS+ can be reported, highlighting a gap that warrants further investigation.

6.5 Attack Surface and Leakage Vectors in HQC

HQC, a code-based KEM, relies on the Fujisaki-Okamoto (FO) transformation to achieve CCA security. This construction exposes a critical attack surface during the re-encryption and validity-checking phase of decapsulation, where the implementation invokes a Pseudo-Random Function (PRF) or Pseudo-Random Number Generator (PRG). Ueno *et al.* demonstrate that these PRF/PRG computations leak sufficient information to build a plaintext-checking oracle via power/EM side-channel analysis, marking the first reported physical analysis of the FO transformation in code-based KEMs. Their framework applies generically to FO-based schemes, and HQC is included in the key-recovery cost evaluation.

However, it is important to note that the authors do *not* present HQC-specific power or EM leakage measurements; rather, they analytically include HQC in their evaluation of re-encryption-based side-channel vulnerabilities. See Tables 8 and 9.

6.6 Single-trace is now a reality

The effectiveness of advanced side-channel analysis (SCA) techniques, particularly those leveraging Deep Learning (DL), has transformed the feasibility of single-trace attacks (STA) from a theoretical threat into a practical reality across multiple Post-Quantum Cryptography (PQC) schemes. Choi et. al. [23] analytically analysis single trace attacks against Falcon. Jendral et. al. [43] shows against Kyber. And Kim et. al. [48,78,102,34] demonstrates against Dilithium. These STA methods demonstrate the capability to recover sensitive information, including full secret keys or shared keys, from implementations protected by common countermeasures, often requiring minimal computational resources. See Table 10 for more details.

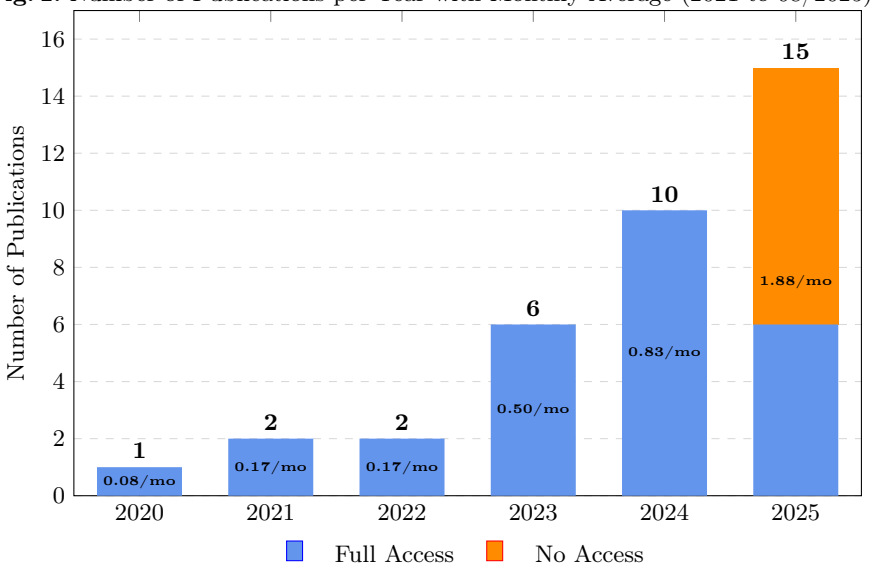
6.7 Bibliometric Analysis of Extracted Data

The following analysis is a synthesis of the information collected from the 27 primary studies using the Data Extraction Form (defined in Table 19). This step builds upon the study selection process (Table 20) and the quality assessment outcomes (Table 21), providing a comprehensive picture of how the literature on AI-based side-channel attacks (SCAs) against PQC has evolved.

Temporal Distribution The analysis of the publication years reveals a clear and accelerating trend as shown in Figure 1. A small number of foundational papers were published between 2020 and 2022. However, a marked increase in publication volume is observed from 2023 onwards, with a significant peak in 2025. This surge not only directly correlates with the conclusion of the third round of the NIST PQC standardization process, but also aligns with the evidence that AI-driven SCAs are highly effective, achieving results that surpass traditional techniques. Such approaches have been demonstrated to break hardware implementations [58], to overcome protected implementations with countermeasures such as masking and constant-time design [25,102], and, in some cases, to succeed with as little as a single trace [102]. These results indicate that AI-based methods can outperform classical template and differential/correlation power analysis (DPA/CPA) techniques in both efficiency and robustness [82], underscoring their significance in the evaluation of post-quantum cryptographic implementations.

Adopted AI Models in SCA Against PQC Figure 2 illustrates the distribution of the most prevailing AI models employed in SCAs targeting PQC algorithms, based on the dataset extracted from the selected studies (Section 4). The analysis highlights that multilayer perceptrons (MLPs) and convolutional

Fig. 1. Number of Publications per Year with Monthly Average (2021 to 08/2025)



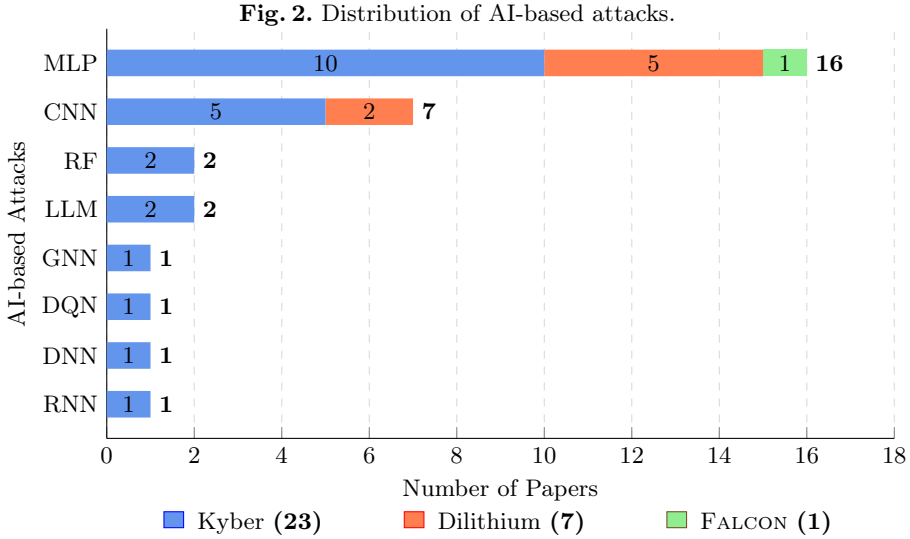
neural networks (CNNs) dominate the landscape, reflecting their effectiveness in capturing and exploiting complex leakage patterns. Other approaches, such as recurrent neural networks (RNNs), Random Forests, and more recent explorations with graph neural networks (GNNs), deep reinforcement learning (DQN), adaptive sparse models (ASPN), and even large language models (LLMs), appear less frequently but signal an emerging diversification of techniques. This distribution indicates that while the community is converging on MLP- and CNN-based profiling as the de facto standard, there is also growing interest in testing novel architectures that may further advance AI-based side-channel analysis.

Targeted PQC Algorithms As shown in Figure 2, the landscape of AI-driven SCA research is heavily skewed toward *Kyber* and *Dilithium*, the primary NIST standards. Conversely, the alternative algorithms have received negligible attention. *Falcon* features in only one publication (restricted to leakage assessment), *HQC* in one analytical paper, and *SPHINCS+* has not been addressed in this context. This concentration of effort leaves the standardized fallback options under-evaluated. To guarantee the resilience of the full PQC ecosystem, future research must expand beyond the primary candidates to include these secondary schemes.

7 Conclusion

The only study applying **fault-injection** attack is [43]. It uses voltage-glitch an instruction (bne) in the Fisher-Yates shuffle, against Kyber. Definitly there is

Fig. 2. Distribution of AI-based attacks.



open research oportunities to combine AI-based attacks with FI, not only using several other FI techniques but also targeting other points.

There are just two papers defeating shuffling countermeasure. None defeats shuffling with higher order masking applied together. None defeats masked Threshold Implementation. So, there is still room for research to defeat stronger countermeasures and combination of them.

This SLR set out to (i) map the state of peer-reviewed, AI-driven side-channel attacks (SCAs) on the NIST PQC standards and the prominent HQC candidate, (ii) classify attacks by linking learning techniques to specific implementation vulnerabilities, and (iii) identify trends, gaps, and open challenges. Our synthesis shows that *AI has reshaped the side-channel threat model*. First, single-trace (or single-signature) feasibility has moved from exception to expectation on embedded targets: Dilithium and Falcon admit high-accuracy single-trace recovery under realistic lab conditions, and Kyber—even with masking and shuffling—has been compromised in one-trace, fault-assisted settings. Second, protections once considered table-stakes (first-order masking, shuffling, constant-time coding, even anti-tamper covers) no longer guarantee safety against modern profiled analysis: MLPs/CNNs learn residual higher-order structure, fault injection erases desynchronization benefits, and tailored CNN variants sustain accuracy through physical shielding. Third, *hybrid pipelines*—learning-based leakage extraction plus algebraic post-processing (e.g., leveraging compressed public-key bits in Dilithium)—consistently outperform either approach alone, tightening attack budgets from thousands to handfuls of traces.

Answering **RQ1**, we catalogued 27 primary studies (cutoff: 16 Aug 2025) that span Kyber, Dilithium, Falcon, and HQC, with no practical AI-driven SCA yet on SLH-DSA (SPHINCS+). **RQ2** yielded a taxonomy aligning model families

to targets and countermeasures: classical ensembles (e.g., Random Forest) enable blind or weakly labeled attacks; deep models (MLP/CNN and lightweight variants like ASPN) break masked/shuffled code and tolerate misalignment; reinforcement learning automates segmentation and other "last-mile" steps; and LLM-assisted workflows lower the expertise barrier for SPA. For **RQ3**, three trends emerged: (1) the ascendancy of single-trace and few-trace practicality; (2) systematic neutralization of standard countermeasures by data-driven models; and (3) growing realism in assumptions (no reference device, blind inputs, physical covers), all of which compress the gap between lab and field.

Gaps & risks. The conspicuous absence of convincing, practical AI-based SCAs on SPHINCS+ suggests a research and evaluation deficit—not necessarily inherent resilience. Leakage around public-key compression (e.g., Dilithium's t_0) has outsized impact on post-processing and should be treated as a first-class attack surface. Finally, reproducibility remains uneven (limited code/datasets), complicating cross-paper comparisons and engineering takeaways. The same applies to HQC with just one paper doing just analytical analysis, and Falcon with just one paper doing leakge assessment (not an key/message recovery attack).

Implications for defenders. Defense must assume an *AI-equipped adversary*. We recommend: (i) layering countermeasures (higher-order masking *with* shuffling and hardened randomness paths), (ii) auditing compression/-packing/unpacking code paths as aggressively as core arithmetic, (iii) adopting design-time leakage discovery and patching (e.g., RTL-level analyses) before FPGA/ASIC sign-off, (iv) validating physical mitigations (e.g., covers) against modern CNNs/RL pipelines, and (v) standardizing evaluation: shared datasets, alignment-robust metrics (GE, SR), and mandatory implementation disclosures to improve reproducibility.

Future work. Priority directions include: scalable, composable leakage proofs that reflect ML adversaries; standardized, open corpora for PQC SCA benchmarking across devices and countermeasures; principled integration of fault + learning under realistic noise/fault budgets; and dedicated studies on SLH-DSA implementations. As our bibliometrics indicate, output in this area is doubling year-over-year, underscoring that the threat landscape—and the corresponding best practices—are rapidly evolving. This review provides a current map; keeping systems secure will require treating AI-aware SCA evaluation as a continuous process throughout the PQC deployment lifecycle.

Table 5. Key Algorithmic Attack Points in **Dilithium**

Key Generation and Unpacking Procedures
Secret Key Unpacking (`skDecode`): This procedure, executed at the start of the signing algorithm, is attacked to recover secret key coefficients s1 and s2 in the range $[-\eta, \eta]$ [102]. The underlying function small_polyeta_unpack processes coefficients twice, resulting in unique Hamming Weight pairs that allow for distinguishability [102].
t0 Unpacking (polyt0_unpack): Targeted for recovering the LSB of t0 coefficients to reduce the number of signatures required to cryptographically reconstruct the full t0 vector [101].
Key Generation Subroutines: When `NTT` is protected, attacks must target sampling, addition, rounding, and packing operations to find the secret vector s2 [34]. The addition operation showed the best recovery results (SR $>$ 98%) among these subroutines in experiments [34].
Core Arithmetic and Polynomial Operations
Number Theoretic Transform (NTT): A critical target in both the key generation (to recover s1) and the signing procedure (to recover s1 and s2) [34,48]. The attack specifically exploits leakage in the Montgomery reduction (Mont_r or montgomery_reduce) operation within the NTT, where the intermediate value t exhibits significant power consumption differences based on the input coefficient p[j+len] [34,48].
Inverse NTT (INTT)/Montgomery Reduction: The INTT operation, particularly the Montgomery reduction involved in computing cs1, cs2, and ct0, is a significant leakage source [78,79]. The operations are identical for cs1, cs2 and ct0, enabling profiling on publicly available ct0 leakage to recover the private terms cs1 and cs2 [79]. Leakage arises from the shifting and storage operations in the final stages of the reduction [78].
Sparse Multiplication: Targeted in single-trace attacks against masked Dilithium implementations, where leakage occurs in the load, save, and multiplication operations involving the Boolean masked secret key shares [48].
Polynomial Multiplication ($c \cdot s1$): The process of polynomial multiplication is susceptible to Correlation Power Analysis (CPA), regardless of whether schoolbook multiplication, NTT, or sparse multiplication is used [58]. The narrow range of the resulting product cs1 favors side-channel attacks [78].
Signature Calculation and Randomness Leakage
Randomness Generation (y): The bit-unpacking function used in ExpandMask to generate the random vector y leaks information about whether coefficients are zero or non-zero, making it a viable target for profiling attacks [58].
Polynomial Addition (`z = y + cs1`): This critical step is exploited in attacks that aim to recover `cs1` or `y` [58,78]. The attack exploits the inherent relationship between the known signature z and the addition of the large random number y (range `q`) and the smaller value $cs1$ (range β) to infer partial information about y (high bits) [78]. The results from individual attacks on y and $cs1$ can be integrated to significantly enhance the overall success rate of cs1 recovery [78].

Table 6. Channels, Leakage Models, and Countermeasures Weakness in **Falcon**

Physical Channels and Measured Leakages
Power Consumption (PA): Power consumption traces are measured as the primary channel using the ChipWhisperer Lite (CW-Lite) platform for security evaluation [23]. – **8-bit AVR MCU (XMEGA128D4)**: Used to study the **visible vulnerability** (non-constant-time leakage) [23]. – **32-bit Arm Cortex-M4 MCU (STM32F303)**: Used to investigate and demonstrate the **invisible vulnerability** (Hamming Weight leakage) in a constant-time execution environment [23]. **Timing**: The variation in the number of clock cycles creates a **visible vulnerability** on 8-bit AVR MCUs [23].
Leakage Models and Vulnerabilities
Visible Leakage (Non-Constant Time): This vulnerability occurs on 8-bit AVR MCUs because the compiler splits large numbers (e.g., 64-bit data used in Mitaka) into smaller units for comparison [23]. This operation is subject to **premature termination (early stop)**, causing variable execution times and rendering the implementation non-constant-time [23]. **Invisible Leakage (Hamming Weight Difference)**: This vulnerability is found in the **constant-time** environment of the 32-bit Arm Cortex-M4 [23]. The leakage is caused by **Hamming Weight (HW) differences** that occur when the results of comparison operations (0 or 1) are accumulated using an add instruction in internal registers [23]. **Deep Learning Profiling**: A profiling analysis utilizing a Deep Learning (MLP) model is successfully employed to predict the CDT sampling output value from a **single power trace** on the Arm Cortex-M4, achieving a recovery accuracy of 99.97% and an F1 score of 1.0 [23].
Vulnerability of Countermeasures
Defeat of Constant Time: A comparison-operation-based CDT sampling algorithm was proposed as a countermeasure to eliminate branching instructions (using 'adc') and satisfy constant-time execution on AVR (defeating visible leakage) [23]. However, this approach does not mitigate the **invisible HW leakage** present on the Arm Cortex-M4 [23]. **Traditional Countermeasures**: Masking and shuffling are considered computationally slow and memory heavy, indicating that further research is required to design secure algorithms against DL-based SCA in the context of CDT sampling [23].

Note: The authors did a leakage assessment only, but did not do a key/message recovery.

Table 7. Key Algorithmic Attack Points in **Falcon**

Gaussian Sampling Procedures
CDT Sampling (Cumulative Distribution Table): This is the fundamental target operation in Falcon and its variants (Mitaka, Antrag, SOLMAE) [23]. It implements discrete Gaussian sampling using repeated **comparison operations** between a random input (rnd) and stored table values to determine the output sampled value (S) [23].
Comparison Operation Leakage: The specific comparison function ('Compare(Ai, Bi)') within the CDT implementation is the exact point of side-channel leakage [23]. – **AVR (Visible)**: Leakage is caused by the variable clock count of branch instructions, resulting from the comparison operation's susceptibility to **premature termination** when processing multi-byte data [23]. – **Arm Cortex-M4 (Invisible)**: Leakage occurs during the **add instruction** that accumulates the results (0 or 1) of the comparison operations due to resulting **Hamming Weight (HW) differences** within the registers [23].
Target of Recovery: The primary objective of the single trace attack is to recover the **sampled value (S)** produced by the CDT sampling, as the leakage of this value can indirectly lead to the exposure of secret information [23].
Note: The authors did a leakage assessment only, but did not do a key/message recovery.

Table 8. Channels, Leakage Models, and Countermeasures Weakness in **HQC**

Physical Channels and Measured Leakages
Power/EM Analysis: The proposed generic attack is a power/Electromagnetic (EM) side-channel analysis methodology that exploits leakage during the re-encryption process [98]. This work constitutes the first reported power/EM analysis of the FO transformation in code-based KEMs such as HQC [98].
Timing: HQC is susceptible to timing attacks if the FO transformation is implemented in a non-constant-time manner, especially in the ciphertext-equality verification step ($c = c'$), which must be constant-time [98]. The proposed attack, however, focuses on power/EM leakage from the PRF execution, which allows it to bypass constant-time countermeasures applied to the equality check [98].
Leakage Models and Profiles
PRF/PRG Leakage: The fundamental leakage mechanism is the side-channel trace produced during the execution of the PRF or PRG in the re-encryption phase of KEM.Decaps [98]. This leakage is utilized to distinguish whether the decrypted plaintext (m') is equivalent to the known reference plaintext (m) or some other random plaintext ($\hat{m}$) [98].
Deep Learning Distinguisher: The attack uses a Deep Learning (DL) based two-classification neural network (NN) to realize the plaintext-checking oracle [98]. This approach does not require specific knowledge about the target implementation, allowing the attack to be mounted in a black-box manner [98].
Vulnerability of Countermeasures
PRF/Masking Defeat: The proposed SCA remains effective even when the PRF (e.g., SHAKE) is software-masked, as the NN distinguisher can still extract sufficient information for plaintext discrimination [98].
Constant-Time Defeat: The attack targets the PRF leakage, meaning it can achieve key recovery independently of the PKE.Dec implementation and remains effective even if the comparison operation ($c = c'$) is implemented securely using constant-time conditional move operations (cmov) [98].
Note: The authors do not perform HQC-specific power/EM measurements; HQC is included analytically in the re-encryption attack framework.

Table 9. Key Algorithmic Attack Points in **HQC**

Plaintext Checking (PC) Oracle via FO Transformation
PRF/Hash Function Execution: The leakage is generated during the execution of the PRF/PRG (denoted as G in the FO transformation) when the decrypted message (m') is re-encrypted (c') to check equality with the received ciphertext (c) [98].
HQC PRF Instantiation: HQC specifically employs SHAKE (the $\mathrm{SHAKE256}_{512}(\mathrm{m}, 0x03)$ function) in its KEM.Decaps re-encryption phase to obtain the decrypted plaintext [98]. The leakage from this SHAKE execution is used to enact the plaintext-checking oracle [98].
KR-PCA Methodology: The Key-Recovery Plaintext-Checking Attack (KR-PCA) can be performed on HQC following a strategy similar to lattice-based KEMs (like FrodoKEM), leveraging the fact that the decrypted plaintext (m') will be 0^{ℓ} or a vector with a single '1' bit ($0^{i-1}\|1\|0^{\ell-i-1}$) [98].
Attack Complexity: The attack requires a feasible number of oracle accesses to achieve full-key recovery, albeit generally more complex than lattice-based KEMs [98]. – hqc128 requires approximately $18,111$ oracle accesses [98]. – hqc256 requires approximately $58,536$ oracle accesses [98].
PKE Short Randomness: The underlying PKE scheme used by HQC internally utilizes the XOF function $\mathrm{SHAKE256}(r, 0x02)$ for short randomness generation [98].

Note: The authors do not perform HQC-specific power/EM measurements; HQC is included analytically in the re-encryption attack framework.

Table 10. Single-Trace Attacks

Kyber: Defeating Masking and Shuffling
Combined Attack: A single-trace message recovery attack was successfully demonstrated against a Kyber KEM software implementation that was protected by both masking and shuffling [43].
Methodology: The attack combines voltage **fault injection** (to bypass the shuffling) with **MLP** profiled power analysis based on the Hamming weight leakage model [43].
Efficiency Gain: This approach recovers the shared key from a single power trace, marking a significant improvement over previous pure SCA methods that required approximately 3K power traces to attack a masked and shuffled Kyber KEM [43]. The success of STA in this context suggests that encapsulation algorithms in Kyber, which are typically unprotected, also require security measures [43].
Dilithium: Single Signature Key Recovery
NTT Operation: Profiling attacks based on **MLP** targeting the Number-Theoretic Transform (NTT) operation during signature generation and key generation can expose the full secret key vectors (s1 and s2) [34,48]. It has been shown experimentally that a single-trace attack can find the full key with a 100% success rate regardless of the optimization level (-O0,-O3,-Os) by targeting the NTT encryption [48].
Defeating Masking: Even when Dilithium is protected by masking, STA remains possible by targeting sparse multiplication, achieving 100% full key exposure regardless of the optimization level [48].
Unpacking Leakage: A single-trace attack exploiting leakage during the secret key unpacking procedure of the signing algorithm can recover the full secret vector s1 with a non-negligible probability (9%) when the low-order bits of the public key vector t (t0) are known. Previous attacks generally required over 100 traces [102].
Single Signature Success: By targeting the polynomial addition z=y+cs1 during signing, practical attacks on Dilithium2 can achieve private key recovery (s1) with a 60% success probability within one hour using leakage from a single signature [78]. Techniques like Linear Regression (LR)-based profiling successfully recover the random vector y with a 40% success rate from one trace, and **CNN**-based methods targeting Montgomery-reduction to recover cs1 with a 74% success rate from a single trace [78]. In hybrid attack scenarios, the private key s1 was successfully recovered using as few as 2 signatures, or optimally, a single signature [78].
Falcon: High-Accuracy Leakage in CDT Sampling
Visible Vulnerability: In the 8-bit AVR environment (non-constant-time implementation), STA can recover sampled values using a single trace by exploiting a visible vulnerability (premature termination of comparison operations) [23].
Invisible Vulnerability (Defeating Constant Time): Even when operating on a modern Arm Cortex-M4 chip in a constant-time environment, STA successfully recovers the CDT sampling value [23]. This attack exploits an invisible vulnerability related to Hamming weight differences that occur during the addition of comparison results [23]. The proposed **MLP** model achieves a recovery success rate of 99.97% using just a single trace [23].

Table 11. Summary of MLP SCAs against Kyber

MLP (Multi-Layer Perceptron neural network)
Kyber-512 (M,HW): Decapsulation; HD(m[i-1],m[i]) labeling; 1000 profiling traces → 255k segments; MLP architecture with BN/Dense/ReLU layers; single-trace HD accuracy ≈ 0.615; full-message success: 97.7% with 299 traces, 99.7% with 399; cross-device profiling reduces success to 94.8%; training hyperparameters explicitly listed [44].
Kyber-768 (M,SW): Decapsulation; m[i] labeling; ω =1st-5th-order masking broken via recursive-training (weight transfer across ω). 30k profiling traces → 960k; cyclic rotations amplify determiners (0x0000/0xFFFF). Single-trace bit accuracy 0.999→0.979 ($\omega = 1 \rightarrow 5$); message success 78.9%→0.56%. For $\omega = 5$, majority voting with 5 power traces reaches 87%. Encapsulation attack, labeling HW(m[i]), failed even unmasked [25].
Kyber-768 (M,SW): Decapsulation; polytomsg(m)[i] labeling; first-order masked poly tomsg attacked. 50k traces; bitwise segmentation ($256 \times 3 \times 225$); MLP achieves 90% validation accuracy and 99% **key-coefficient recovery** in 65h; Attack requires chosen ciphertexts, preprocessing and ECT-based error correction [31]. See Table 14 for failing RNN attack of this paper.
Kyber-768 (M,S,SW): Decapsulation; HW(msg byte) labeling; countermeasures evaluated: masking, noise, random delay, clock jitter. classifier with several hidden layers; trimming Masked_Encode boosts accuracy 30%→88%; Decode DL accuracy drops to 0.37% under N0.1; Masked_Decode <1% accuracy with noise or AR200 jitter [38].
Kyber-768 (M,S,SW): Decapsulation; HD(m[i-1],m[i]) labeling; Single-trace DL-SCA + **Fault Injection** voltage-glitch bypass of Fisher-Yates shuffle; 10k profiling traces → 2.56M segments, one attack trace; MLP (320-dim input, Nadam, ≤ 250 epochs) predicts message bits; full-message success 0.122→0.887 (32-bit enumeration)→0.969 (64-bit enumeration). [43].
Kyber-768 (HW): Decapsulation; m[i] labeling; Profiling uses 200k traces → 6.4M samples; MLP (BN-Dense-BN-ReLU-Dense-Softmax, 21k params) trained with Nadam (100 epochs). Attack uses $256 \times 4 \times 5$ traces via 4-sliced multi-bit error injection, recovering all messages with byte-rank ≤ 3 (enumeration ≤ $4^{32} = 2^{64}$). Recovering messages from 7 chosen ciphertexts enables long-term **secret-key extraction** [45].
Kyber-768 (SW, Masked): Decapsulation; first-order masked message encoding. Profiling (BN-Dense-BN-Dense-BN-Dense, ReLU, Softmax, Dropout 0.5) on joined-share traces (input=380). 1000 traces → 32k samples; Adam, CE loss; val. acc. reaches 100%. Single-trace message recovery; 2-stage CCA-assisted attack recovers **full secret key** with 9 traces (≈ 100% success), or 18 traces with negligible brute-force. [100]
Kyber512 (SW): Decapsulation; labels=HW(R1 reduction, profiled)→applied to R2 (key-dependent). MLP 160-100-40-12, ReLU/Softmax, trained on 10^6 R1 sub-traces (SGD, lr=0.01). Validation accuracy 68.6%. **Key-recovery** via probabilistic accumulation: 76.9% (5 traces)→100% (11 traces). [108]
Kyber (SW): Decapsulation; Blind SCA via DL-BSCA using MLP classifier with MC-labeling (GMM on 100 PoIs). 80k profiling traces; 20k attack; label encoding $(B+1)h_m+h_y$ with ≈1.2% correct labels. Hyperparameter search over 100 models (dropout/no-dropout). Achieves GE=9.2 for s_0 recovery under blind conditions. Has been outperformed by its CNN version [82]..
Kyber (SW, Masked): Decapsulation; FO-based re-encryption leakage through masked software PRF. Attack uses a 5-layer MLP (FC1-FC5) with Sigmoid output for two-class (label) distinguishing of PRF inputs. Fully-connected design chosen to capture multivariate leakage of masked SW. Produces a reliable plaintext-checking oracle enabling adaptive re-encryption; key recovery succeeds within ≤60k oracle traces. [98]. **HQC** was included analytically, but it wasn't really attacked.

(SW)Firmware, (HW)FPGA, (R)RTL, (M)Masked, (S)Shuffled, (A)Anti-Tampering Cover

Table 12. Summary of MLP SCAs against Dilithium and Falcon

MLP (Multi-Layer Perceptron neural network)
Dilithium (SW): Signing + key-gen; secret-coefficient labeling (s1,s2). single-trace profiling attack on NTT (s1: 100%) and sampling/addition/rounding/packing (s2: $\leq$ 98%). 60k traces (45k train/5k val/10k attack). NN: BN-Dense-BN-ReLU-Dense-Softmax (Nadam). Full s1 recovery; s2 up to 98% per op. [34].
Dilithium (SW): Signing; $y_{i,j}$ zero/nonzero labeling; Four MLP classifiers (Dense/Dropout stacks; up to $\approx 3.7\times10^5$ params) tuned via Hyperband; 548-sample trace windows. Profiling on Device A; attack uses $\sim$100k signing traces; classifier accuracy $\approx$ 0.999. Full s_1 recovery and end-to-end **key extraction**; Level-2 break requires $\approx$10 min of signatures. [58]
Dilithium-2 (SW): Signing; coefficient-value labeling ($-2..2$) during `skDecode`. Eight MLP classifiers (440-sample input; 5 classes; Nadam, 100 epochs) trained on 3.2M segments ($<$10h profiling). Single-trace coefficient recovery $\approx$ 0.92; full s_1 recovery with single trace succeeds with 9% (linear-equation variant). 1000 traces $\rightarrow$ 100% **key recovery**; BKZ variant also feasible without t_0. [102]
Dilithium-2 (SW): Signing; LSB(t_0[i]) labeling during `polyt0_unpack`. 168 models (8 coeff.-types; 512-256-256 FC layers); trained on 2.56M segments (550-sample windows). Majority voting yields $\approx$57 wrong LSBs; unanimous voting reduces to $\approx$15 with $\approx$126 undecided. With 960 averaged traces, 80.5% of (s_1,s_2) coefficients recovered with 0.875 probability; combined with partial t_0 recovery gives $\approx$ 0.7 full s_1 **key recovery**. [101]
Dilithium-2 (M, SW): Signing; NTT-encryption (unmasked) and sparse multiplication (masked). Boolean masking defeated. Profiling via MLP (BN-Dense-BN-ReLU-Dense-Softmax; Adam, CE loss; inputs normalized to [-1,1]). Labels = 3Byte of $s_1[i]$ (unmasked) or all 256 byte-values of masked shares. Unmasked: 2000 profiling traces, 8000 attack traces $\rightarrow$ 100% full-key recovery. Masked: 9000 profiling, 8000 attack traces $\rightarrow$ 100% **full-key recovery** (single-trace). [48]
Falcon (SW): CDT sampling; Visible early-stop leakage on AVR and invisible Hamming-weight leakage on ARM defeat comparison-op "constant-time" CDT. BN-Dense-BN-Dense-BN-Dense trained on 112k traces (736-sample windows; Adam lr= 10^{-4}; 10 epochs). Test accuracy 99.97%, F1=1.0. Single-trace recovery of CDT output (14 classes). Leakage assessment only, no key/message recovery. [23]

(SW)Firmware, (HW)FPGA, (M)Masked

Table 13. Summary of CNN SCAs

CNN (Convolutional neural network)
Kyber (SW): Decapsulation; Blind SCA via DL-BSCA using CNN trained with MC-labeling (GMM on 100 PoIs). 80k profiling traces; 20k attack; same noisy-label encoding $(B+1)h_m+h_y$. Hyperparameter search over 100 architectures. Best CNN+MC configuration reaches GE=2.01, successfully recovering s_0 under blind conditions. Has outperformed its MLP version [82].
Kyber (M,SW,HW): Decapsulation; FO-based re-encryption leakage via PRF executions. CNN distinguisher (Conv1-Conv6 + FC, Sigmoid) trained to classify PRF inputs as reference vs. random (label). CNN attains ≈99.8-99.9% accuracy on unprotected SW/HW, enabling a strong plaintext-checking oracle and full key recovery within ≤60k traces. Under TI-masked HW, accuracy collapses to ≈51%, rendering the oracle ineffective [98].
Kyber (SW): Decapsulation; labels = secret-key coefficient s_0 (0-3328). CNN with ciphertext-knowledge (5 Conv layers + Dense fusion + 5-layer classifier; RMSprop, lr 10^{-6}) trained on 200k-500k traces (600 samples/trace). Attack phase requires ≤50 traces (500k profiling) to reach mean-rank 0 and recover targeted 12-bit key coefficients. Ciphertext knowledge (2 or 4 coeffs) reduces profiling/attack traces needed. No masking/shuffling; **full key recovery** achieved by iterating per-coefficient attack. [36]
Kyber (SW): Decapsulation; FO re-encryption leakage from PRF/RO execution (software, power/EM). Multi-valued PC oracle via deep NN (6 Conv layers + BN/AvgPool + FC1-FC3 with SELU/Softmax). Training uses μ-class labels from valid ciphertexts. Achieves full key recovery with **576 traces** (vs. 3,072 baseline), an **81%** reduction at 99.9999% success probability. [93]
Kyber (A, SW): Decapsulation; Adaptive Slimmed Pyramid Network (ASPN) distinguisher. Binary classification/labeling of m_0/m_1 from re-encryption EM traces; 100k-sample traces; 800 train / 200 test per class; accuracies 0.99 (18 mm) to 0.893 (24 mm with clone); full key recovery with 5818.5 traces; ASPN3 optimal; train 39.41 s / test 0.49 s; T-test reduces model size by 94% while reaching 0.973 accuracy [22].
Dilithium-2 (SW): Targets Montgomery-reduction leakage to recover cs_1 (labeled). Architecture: 3 Conv layers (filters 12/24/48; kernels 64/128/256; strides 1/6/1), BN + avg-pool after each, followed by one Dense-Softmax layer (Adam, $lr = 4\times10^{-4}$, 150 epochs, batch=1600). Trained on 200k segments; single-trace cs_1 recovery SR=70.5-74.3%. When fused with LR-based y recovery and COBRA/ILP equation solving, the attack achieves full s_1 **key recovery** in 1-3 signatures [78].
Dilithium-2/3/5 (SW): Signing; leakage during INTT and Montgomery reduction. Labels = ct_0 coefficients (sign and value). CNN distinguisher: (i) 1-layer CNN for sign; (ii) 2 × Conv (32,64, 3 × 3) + BN + MaxPool + FC(128,64,ReLU) for value (Adam, CrossEntropyLoss). Trained on 40k ct_0 traces; tested on 10k cs_1/cs_2 samples. Success rates (range [–31,31]): $\mathrm{cs}_1/\mathrm{cs}_2$ = 26.7/36.2% (Dilithium2), 20.5/28.8% (Dilithium3), 22.4/31.5% (Dilithium5). **Full $\mathbf{s}_1$,$\mathbf{s}_2$ recovery** in 10/27/18 traces using residual-analysis solver [79].

(SW)Firmware, (HW)FPGA, (M)Masked, (A)Anti-Tampering Cover

Table 14. Summary of othes AI-based Techniques SCAs

RF (Random Forest)

Kyber (SW): Decapsulation; labels = HW(m), HW(y) from basemul/fqmul in pointwise multiplication. Random Forest HW classifier trained on clone-device PoIs (CPA-selected); O0 HW accuracy 69–80% (input), 81–85% (output); O3 accuracy ≈70–72%. Simulated attack achieves coefficient GE=0 in ≈671 traces (perfect HW) or ≈8524 traces (95% HW). Practical evaluation recovers 20 secret coefficients with 35–5000 traces. **Full-key recovery** feasible under blind setting. [81]

Kyber512/768/1024 (O): Decapsulation; decryption-failure oracle exploited to build bilateral linear inequalities on secret key coefficients. Oracle remains extractable even with masked/constant-time ciphertext-comparison and sanity-check countermeasures. ML-based contraction uses a Random Forest (300 trees, depth 30) trained on JSD sequences over BP iterations. Filtered/unfiltered cases need 5.2k/8k inequalities. **Full key recovery** with ≈5.5k (512), 7k (768), 6.75k (1024) inequalities; query factor ≈2.5-2.6 and ≈40-48% query-complexity reduction [77].

LLM (Large Language Model)

Kyber-256 (SW): Decapsulation; *operation-type* labeling for SPA (256 ops). ChatGPT used as zero-shot classifier (no NN training); segmented traces (max 36 ops/query). 0-shot/general prompts give OPSR=0, but expert prompting (strategies 1&2) enables **full private-key recovery**, with OPSR≈0.5 over 10 runs [111].

Kyber (R): Post-analysis fortification; SCAR employs an LLM (Falcon-7B) to automatically generate leakage-mitigated RTL by inserting Boolean masking into vulnerable lines flagged by the GNN. The LLM rewrites RTL modules (e.g., `intmul`) into masked variants; validation via post-synthesis dynamic-power analysis confirms reduced leakage at the modified sites. No AI-based attack performed; LLM used solely for countermeasure synthesis. [91]

GNN (Graph Neural Network)

Kyber (R): Leakage assessment (not attacking); SCAR analyzes Kyber's RTL CDFG using a GNN (GCN1-GCN2 + FC + sigmoid), trained on AES-Comp (lr=0.01, batch=20, dropout=0.3, 32 epochs). Node-features: vulnerable-path count, degree, Hamming-distance, AND/OR/XOR/MUX ops. Achieves 88.89% accuracy, 95.38% precision, 91.20% recall for Kyber; identifies the `intmul` module as leakage-prone. [91]

DQN

Kyber (SW,HW): Decapsulation; *no leakage labeling* (RL is label-free). SPA-GPT uses Deep Q-Network (DQN) with a 4-layer **MLP** (BN + Dense(40)×3) to automatically segment a *single* power trace, since MLP-based Q-networks approximate the Q-table efficiently for large action spaces. Reward = negative mean Euclidean distance between segments. Achieves **100% segmentation accuracy** on Kyber traces and recovers the *intermediate value crucial for deriving the secret key*; no masking defeated, no full-key recovery demonstrated. [104]

DNN (Deep Neural Network)

Kyber512/768/1024 (M, SW): Decapsulation; $(m \neq m')$ labeling. Neural-network-based multi-valued PC-oracle built from a fully connected feed-forward NN (3 hidden layers, 1024 units each; ReLU; Softmax; Adam, lr=0.001; batch=64; 10 epochs). Input uses $t_1 = 3$ PoIs; each oracle query uses $t_1 t_2 = 9$ traces. NN accuracy $\approx$ 98.97%, rising to $\geq$ 99.96% with majority voting. **Full-key recovery** requires 432 / 648 / 864 traces for Kyber512/768/1024. [103]

RNN (Recurrent Neural Network)

Kyber-768 (M,SW): **Fails to recover key**. Decapsulation; polytomsg(m)[i] labeling; first-order masked poly tomsg attacked. 50k traces; bitwise segmentation ($256 \times 3 \times 225$); <70% accuracy and fails to converge. Attack requires chosen ciphertexts, preprocessing and ECT-based error correction [31]. See Table 11 for successfully MLP attack of this paper.

(SW) Firmware, (HW) FPGA, (R) RTL, (M) Masked, O(Oracle Based)

Neural / Deep Classical Reinforcement / Decision

A Full Methodology Details

This chapter delineates the systematic methodology employed to identify, select, appraise, and synthesize the existing body of literature concerning the application of Artificial Intelligence (AI) techniques to conduct Side-Channel Attacks (SCAs) against the cryptographic algorithms standardized or finalized by the NIST Post-Quantum Cryptography (PQC) project. The protocol is designed to be transparent, replicable, and rigorous, following the guidelines for systematic literature reviews proposed by Kitchenham and Charters [50,51], ensuring the resulting review is comprehensive and unbiased. The review process is structured into three primary phases: planning, conducting, and reporting [50]. This section details the first two phases.

A.1 Search Strategy

A multi-stage search strategy was executed to ensure a comprehensive and unbiased collection of relevant primary studies. The search was concluded on the cutoff date of August 16, 2025.

Database Selection The selection of information sources is critical for capturing high-quality, peer-reviewed research in this fast-moving field. The following digital libraries and archives were chosen based on their prevalence as publication venues for top-tier cryptography, hardware security, and computer security research:

- **IEEE Xplore:** A comprehensive database covering key conferences and journals in hardware security and electronics [40].
- **ACM Digital Library:** The main source for publications from the Association for Computing Machinery, including flagship security conferences like CCS and ASIACCS [7].
- **Engineering Village:** A comprehensive engineering database covering a wide range of technical literature, also provided by Elsevier [26].
- **ScienceDirect & Scopus:** Major interdisciplinary databases from Elsevier, providing broad coverage of scientific and technical literature [28,27].
- **SpringerLink (including LNCS):** Publisher of major cryptography conference proceedings, most notably the *Transactions on Cryptographic Hardware and Embedded Systems* (TCHES), the flagship journal for side-channel research [90].
- **IACR ePrint Archive:** The primary repository for pre-print cryptographic research, essential for capturing the latest findings [41].

Search String Construction To ensure the search query is systematic and directly addresses the research questions, the PICOC (Population, Intervention, Comparison, Outcome, Context) framework was used to deconstruct the query into its core concepts and corresponding keywords [51]. This structured approach is vital for achieving high recall, as the terminology in this interdisciplinary field is highly specific.

Search String The keywords from the PICOC framework (Table 15) were combined using Boolean operators (AND, OR) to form the final search query (Fig. 3). It is a standard and necessary step in a systematic review to refine the comprehensive list of concepts from the PICOC framework into a practical search string optimized to balance recall (finding all relevant studies) and precision (excluding irrelevant studies) [51].

This refinement involves three key adjustments:

1. **Ensuring Comprehensive Coverage:** The PICOC list contains both the submission names (e.g., "Kyber") and the final FIPS standard names (e.g., "ML-KEM"). While the majority of the research literature still refers to the submission names, especially in works published before or shortly after standardization in 2024 [69], newer publications increasingly adopt the FIPS designations. To capture both historical and future works, our final search string explicitly included *both* forms (e.g., "Kyber" OR "ML-KEM"), thereby maximizing recall and ensuring consistency across publication timelines.
2. **Improving Precision:** The PICOC list includes broad categorical terms like "lattice-based" or "hash-based." Including these in the primary search string would retrieve a vast number of articles on cryptographic schemes that are not part of the NIST PQC final selection, significantly reducing the precision of the search. The final string was therefore focused on the specific names of the selected algorithms to anchor the search to the core population of this review.
3. **Excluding Unused PICOC Dimensions:** Although the PICOC framework also defines *Comparison* and *Outcome*, these elements were not included in the final search string. This decision improved precision, since they either introduced excessive noise (e.g., generic countermeasure terms) or were already implicitly captured by the selected *Population*, *Intervention*, and *Context* dimensions.

To enhance the precision of the search and ensure the relevance of the retrieved studies, the query was executed against the title, abstract, and keyword fields of the articles within the selected databases. The final string was designed to be adaptable to the specific syntax of each database.

Gold Set Validation To validate the effectiveness of the search query, a "Gold Set" of 8 highly relevant articles, identified during the initial exploratory phase and detailed in Table 16, was used. This set serves as a ground truth — often referred to as a quasi-gold standard in SLR methodology [110] — to empirically test the search string's ability to retrieve cornerstone papers in the field. The search query was considered effective as it successfully retrieved all 8 articles, confirming its high recall.

A.2 Study Selection Criteria and Process

A systematic, multi-stage screening process was applied to the results of the literature search to identify studies that meet the predefined eligibility criteria.

Table 15. PICOC Framework for Search Query Definition

PICOC Element	Definition for this SLR	Keywords and Synonyms
Population	NIST Post-Quantum Cryptography (PQC) Round 4 finalists and standardized algorithms.	"post-quantum cryptography", "PQC", "quantum-resistant", "Kyber", "ML-KEM", "Dilithium", "ML-DSA", "SPHINCS+", "SLH-DSA", "Falcon", "FN-DSA", "HQC"
Intervention	The application of AI, Machine Learning, or Deep Learning models as the primary analysis tool or distinguisher in an attack.	"artificial intelligence", "AI", "machine learning", "ML", "deep learning", "DL", "neural network", "CNN", "recurrent neural network", "RNN", "generative adversarial network", "GAN", "transformer", "reinforcement learning", "graph neural network", "GNN", "support vector machine", "SVM", "random forest", "gradient boosting", "computational intelligence", "data mining", "predictive modeling"
Context	Side-channel analysis of physical cryptographic implementations.	"side-channel", "side channel", "SCA", "power analysis", "timing attack", "electromagnetic analysis"

Inclusion and Exclusion Criteria The inclusion and exclusion criteria provide the explicit rules that govern the selection of studies. This formalization is essential to eliminate subjective bias and ensure the process is transparent and replicable [50,51]. The criteria is detailed in Table 17.

The screening process was conducted in two main phases by two independent reviewers: a title and abstract screening, followed by a full-text screening. Any disagreements were resolved through discussion to reach a consensus.

A.3 Quality Assessment and Data Extraction

For each study that passed the screening process, a quality assessment was performed, and a structured data extraction process was initiated.

Quality Assessment Checklist The quality of each included study was appraised to assess its rigor, validity, and risk of bias. This is a critical step in the systematic review process, as recommended by established guidelines [50], to ensure a robust synthesis of the literature. The specific criteria used for this appraisal are detailed in the quality assessment checklist presented in Table 18.

To systematically evaluate each study against these questions, a three-point scoring scale was used: a score of **1.0 for 'Yes'** if the criterion was fully met, **0.5 for 'Partially'** if the criterion was addressed but lacked sufficient detail, and **0.0 for 'No'** if the criterion was not met. This approach provides a quantitative measure of each study's methodological rigor and transparency [51].

It is important to note that while this scoring allows for a quantitative comparison, no quality cutoff score was used to exclude studies. The primary purpose of the quality assessment in this review is not to filter the literature but to aid in

Fig. 3. Base search string.

```
("post-quantum cryptography" OR "PQC" OR "quantum-resistant"
    ↪ OR "Kyber" OR "Dilithium" OR "SPHINCS+" OR
    ↪ "F\textsc{alcon}" OR "Hamming-Quasi-Cyclic" OR
    ↪ "ML-KEM" OR "ML-DSA" OR "SLH-DSA" OR "FN-DSA" OR
    ↪ "HQC")
AND
("artificial intelligence" OR "AI" OR "machine learning" OR
    ↪ "ML" OR "deep learning" OR "DL" OR "neural network"
    ↪ OR "CNN" OR "reinforcement learning" OR "generative
    ↪ AI" OR "transformer" OR "generative adversarial
    ↪ network" OR "GAN" OR "recurrent neural network" OR
    ↪ "RNN" OR "support vector machine" OR "SVM" OR
    ↪ "gradient boosting" OR "random forest" OR "graph
    ↪ neural network" OR "GNN" OR "computational
    ↪ intelligence" OR "data mining" OR "predictive
    ↪ modeling")
AND
("side-channel" OR "side channel" OR "SCA" OR "power
    ↪ analysis" OR "timing attack" OR "electromagnetic
    ↪ analysis" )
```

the synthesis and interpretation of the results [50]. The scores are used to contextualize findings and weigh the strength of evidence, particularly when comparing studies with conflicting results.

Data Extraction Form A data extraction form was designed to systematically collect relevant information from each included study. This structured approach, detailed in Table 19, ensures consistency and facilitates the aggregation and comparison of data across studies to answer the research questions. Systematically documenting study characteristics and findings in this manner is a crucial step that supports clarity, precise reporting, and the ability to replicate the examination, as mandated by SLR guidelines [50].

Table 16. Revised Gold Set for Database Search Validation

#	Article Title / Reference	Justification for Inclusion
1	**Deep learning enhanced side channel analysis on CRYSTALS-Kyber** (Hoang et al., 2024) [36]	This paper presents the first known side-channel attack on Kyber using a Convolutional Neural Network (CNN), recovering the private key with only 50 traces in a practical, black-box scenario.
2	**Breaking the Blindfold: Deep Learning-based Blind Side-channel Analysis** (Rezaeezade et al., 2024) [82]	A cutting-edge paper from a top-tier venue (USENIX Security) that introduces a novel deep learning framework and validates it with a successful attack on Kyber.
3	**Single-Trace Side-Channel Attacks on CRYSTALS-Dilithium: Myth or Reality?** (Wang et al., 2023) [102]	This paper details a profiled deep learning-assisted power analysis attack against CRYSTALS-Dilithium using a Multilayer Perceptron (MLP), achieving the first successful full key recovery from a single power trace by targeting the secret key unpacking procedure.
4	**A side-channel attack on a masked hardware implementation of CRYSTALS-Kyber** (Ji & Dubrova, 2025) [44]	This journal paper describes a profiled deep learning-assisted attack using a Multilayer Perceptron (MLP) against a first-order masked hardware (FPGA) implementation of Kyber, demonstrating the first practical message recovery attack on a protected hardware version of the algorithm.
5	**Breaking a Fifth-Order Masked Implementation of CRYSTALS-Kyber by Copy-Paste** (Dubrova et al., 2023) [25]	This highly significant paper from APKC'23 demonstrates a neural network attack that breaks a very high-order (fifth-order) masked implementation of Kyber using a Multilayer Perceptron (MLP) trained with a novel "recursive learning" method, pushing the state-of-the-art.
6	**Curse of Re-encryption: A Generic Power/EM Analysis on Post-Quantum KEMs** (Ueno et al., 2022) [98]	This foundational paper details a generic side-channel attack against numerous post-quantum KEMs, including Kyber, and validates its practicality by using a deep learning-based (CNN/MLP) distinguisher to create a plaintext-checking oracle.
7	**Single Trace Analysis of Visible vs. Invisible Leakage for Comparison-Operation-Based CDT Sampling** (Choi et al., 2024) [23]	This paper proposes a deep learning-based side-channel attack using a Multilayer Perceptron (MLP) to recover CDT sampling values from the Falcon signature scheme on an ARM Cortex-M4 by exploiting a novel "invisible" vulnerability in constant-time comparison operations.

Table 17. Inclusion and Exclusion Criteria

Criteria	Inclusion (Must Have)	Exclusion (Must Not Have)
Subject Matter	The study must describe a side-channel attack that uses an AI/ML/DL model as the primary distinguisher or analysis tool.	The study describes a conventional SCA (e.g., CPA, TA) without an AI/ML component. The study describes a different type of attack (e.g., fault injection, cryptanalysis). The study is about AI security in general, not SCA.
Target Algorithm	The target of the attack must be an implementation of a NIST PQC Round 4 finalist or a standardized algorithm (Kyber, Dilithium, SPHINCS+, Falcon, HQC, etc.).	The target is exclusively a pre-quantum algorithm (e.g., AES, RSA) UNLESS the technique is explicitly presented as a foundational method later applied to PQC in other works.
Publication Type	The study must be a full-text, peer-reviewed research paper (journal, conference, or workshop) or a relevant technical pre-print (e.g., from IACR ePrint).	The publication is a poster, abstract, presentation slide deck, editorial, or summary without sufficient technical detail.
Language	The study must be written in English.	The study is written in any language other than English.
Availability	The full text of the study must be accessible.	The full text cannot be obtained.

Table 18. Quality Assessment Checklist

QA#	Quality Assessment Question
QA1	Are the research objectives and attack goals clearly stated?
QA2	Is the target PQC algorithm, implementation (software/hardware), and platform described in sufficient detail to allow for replication?
QA3	Is the side-channel data acquisition setup (e.g., equipment, sampling rate, number of traces) clearly documented?
QA4	Are the AI/ML model architecture, hyperparameters, and training process adequately specified?
QA5	Are the results evaluated using standard, appropriate metrics for both SCA (e.g., Guessing Entropy, Success Rate) and ML (e.g., accuracy, loss)?
QA6	Does the study discuss its limitations and potential threats to the validity of its findings?
QA7	Is the source code or dataset made publicly available to support reproducibility?

Table 19. Data Extraction Form

Category	Data Field	Description
Bibliographic Data	Study ID	Unique identifier for the study within this SLR.
	Authors & Year	Authors and publication year.
	Title & Venue	Full title and publication venue (e.g., TCHES 2025).
PQC Target Details	PQC Algorithm	Kyber, Dilithium, SPHINCS+, Falcon, HQC.
	Security Level	e.g., NIST Level 1 (Kyber-512), Level 3, Level 5.
	Attack Vector	e.g. Re-encrption of Decriptation, Fujisaki–Okamoto (FO) transformation, etc...
	Implementation Type	FPGA, ASIC, RTL, Software.
	Target Platform	8-bit AVR, ARM Cortex-M4, Artix-7 FPGA.
	Target Platform Details	The specific processor, board or setup.
	Countermeasures	None, Masking, Shuffling, Hiding, Constant-Time, Anti-tampering or a combination.
	Countermeasures Details	e.g. Masking order.
Side-Channel Context	Side-Channel Type	Power, Electromagnetic (EM), Timing.
	Fault Injection	Yes or No.
	Tool	Setup. e.g. ChipWhisperer, Ociloscope, etc...
AI/ML Intervention	Model Type	e.g., CNN, MLP, SVM, RNN.
	Model Details	Key details (e.g., number of layers, filter sizes for CNNs).
	Profiling Traces	Number of traces used for training the model.
	Attack Traces	Number of traces used to perform the key recovery attack.
Attack Outcome	Primary Metric	Guessing Entropy (GE), Success Rate (SR), Key Rank.
	Result	The reported value of the primary metric (e.g., "GE drops to 0 after 50 traces").
	Key Findings	A qualitative summary of the main conclusions drawn by the authors.

B Study Selection

This section presents the findings obtained by executing the review protocol detailed in the previous section. We begin by reporting the results of the literature search and selection process, followed by a summary of the quality assessment. Then we synthesize the key findings from the selected primary studies, focusing on the most relevant trends and insights regarding AI-based side-channel attacks against post-quantum cryptographic algorithms. The section concludes with a bibliometric analysis derived from the data systematically collected from the final set of primary studies.

B.1 Literature Search and Initial Results

The initial phase of the systematic literature review consisted of querying seven electronic databases: *IEEE Xplore*, *ACM Digital Library*, *Engineering Village*, *ScienceDirect*, *Springer Link*, the *IACR ePrint Archive*, and *Scopus*. This comprehensive search strategy yielded a total of **1,495 articles**. A subsequent de-duplication process eliminated 157 duplicate records, resulting in a set of **1,338 unique primary studies**.

These studies were then assessed according to the predefined inclusion and exclusion criteria. As shown in Table 20, the majority of records originated from *Springer Link* (1,000), followed by *ScienceDirect* (175) and *Engineering Village* (89). After de-duplication, *Springer Link* contributed 990 unique entries, while *IEEE Xplore* and *ScienceDirect* accounted for 60 and 175 unique entries, respectively.

Table 20. Summary of the Study Selection Process by Source

Source	Found	Unique*	No Access	Potentially Relevant	Selected
IEEE Xplore	61	60	0	10	10
ACM Digital Library	5	3	0	1	1
Engineering Village	89	54	4	3	3
ScienceDirect	175	175	0	1	1
Springer Link	1000	990	5	2	2
IACR ePrint Archive	56	36	0	7	7
Scopus	109	20	0	3	3
Total	**1495**	**1338**	**9**	**27**	**27**

(*) After de-duplication process Cutoff date of August 16, 2025

During the screening process, **nine articles were excluded due to lack of access**, leaving **1,329 accessible papers** for further evaluation. Of these, **27 were identified as potentially relevant** and subsequently confirmed as the final set of **27 selected studies** for detailed analysis.

B.2 Quality Assessment Results

The Quality Assessment Checklist defined in the methodology (Table 18) have been applied to all 27 selected primary studies. The purpose of this step was not to exclude low-quality papers, but to provide a systematic appraisal of the rigor of the included research and to inform the data synthesis phase.

The overall score is reported in Table 21. The results show that a large majority of papers (23 out of 27) scored 5.0 or higher. The most common reason for a score below 7 was the missing of source code or dataset to allow replication (QA7).

Table 21. Distribution of Quality Assessment Scores

Quality Score	Number of Papers
7.0	1
6.5	2
6.0	2
5.5	9
5.0	8
< 5.0	5
Total	**27**

C Additional Tables

Table 22. NIST Standardized and Forthcoming PQC Algorithms

Standard Name	FIPS Standard	FIPS Date / Status	NIST IR	NIST IR Date	Underlying Algorithm	Cryptographic Family	Primary Function
ML-KEM	FIPS 203	2024-08-13 [69]	8413	2022-07-05 [3]	CRYSTALS-Kyber	Lattice-Based (MLWE)	Key Encapsulation (Primary)
ML-DSA	FIPS 204	2024-08-13 [69]	8413	2022-07-05 [3]	CRYSTALS-Dilithium	Lattice-Based (MLWE)	Digital Signature (Primary)
SLH-DSA	FIPS 205	2024-08-13 [69]	8413	2022-07-05 [3]	SPHINCS+	Hash-Based	Digital Signature (Backup)
HQC	Pending	Expected 2027 [71]	8545	2025-03-11 [4]	Hamming-Quasi-Cyclic	Code-Based	Key Encapsulation (Backup)
FN-DSA	FIPS 206 (Draft)	Expected 2025 [72]	8413	2022-07-05 [3]	FALCON	Lattice-Based (NTRU)	Digital Signature (Additional)

Table 23. Notable PQC Candidates Not Selected for Standardization

Algorithm	Family	NIST IR	NIST IR Date	Reason for Not Being Standardized / Dropped
SIKE	Isogeny-Based	8545	March 11, 2025 [4]	A practical key recovery attack was demonstrated using a classical computer, rendering the scheme insecure and leading to its withdrawal from the standardization process [19,86,87].
Classic McEliece	Code-Based	8545	March 11, 2025 [4]	Not selected for standardization due to concerns about very large public key sizes, skepticism about widespread adoption, and a desire to avoid creating standards incompatible with an ongoing ISO effort [4].
BIKE	Code-Based	8545	March 11, 2025 [4]	A Round 4 finalist, but not selected in favor of HQC, which had a more mature and stable security analysis [4,75].
Rainbow	Multivariate	8413	July 5, 2022 [3]	A practical attack in early 2022 broke the scheme, undermining confidence in multivariate signatures and leading to its withdrawal [15,96].

References

1. Alagic, G., Alperin-Sheriff, J., Apon, D., Cooper, D., Dang, Q., Kelsey, J., Liu, Y.K., Miller, C., Moody, D., Peralta, R., Perlner, R., Robinson, A., Smith-Tone, D.: Status report on the second round of the NIST post-quantum cryptography standardization process. Tech. Rep. NIST IR 8309, National Institute of Standards and Technology (2020). https://doi.org/10.6028/NIST.IR.8309
2. Alagic, G., Alperin-Sheriff, J., Apon, D., Cooper, D., Dang, Q., Miller, C., Moody, D., Peralta, R., Perlner, R., Robinson, A., Smith-Tone, D., Liu, Y.K.: Status report on the first round of the nist post-quantum cryptography standardization process (01 2019). https://doi.org/https://doi.org/10.6028/NIST.IR.8240, https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=927303
3. Alagic, G., Apon, D., Cooper, D., Dang, Q., Dang, T., Kelsey, J., Lichtinger, J., Liu, Y.K., Miller, C., Moody, D., Peralta, R., Perlner, R., Robinson, A., Smith-Tone, D.: Status report on the third round of the nist post-quantum cryptography standardization process. Tech. Rep. NIST IR 8413, National Institute of Standards and Technology (9 2022). https://doi.org/10.6028/NIST.IR.8413-upd1, https://doi.org/10.6028/NIST.IR.8413-upd1
4. Alagic, G., Bros, M., Ciadoux, P., Cooper, D., Dang, Q., Dang, T., Kelsey, J., Lichtinger, J., Liu, Y.K., Miller, C., Moody, D., Peralta, R., Perlner, R., Robinson, A., Silberg, H.: Status report on the fourth round of the nist post-quantum cryptography standardization process. Tech. Rep. NIST IR 8545, National Institute of Standards and Technology (3 2025). https://doi.org/10.6028/NIST.IR.8545
5. Alkim, E., Ducas, L., Pöppelmann, T., Schwabe, P.: Post-quantum key Exchange—A new hope. In: 25th USENIX Security Symposium (USENIX Security 16). pp. 327–343. USENIX Association, Austin, TX (08 2016), https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/alkim
6. Asghar, A., Becher, A., Ziener, D.: Backing the wrong horse: How bit-level netlist augmentation can counter power side channel attacks (2025). https://doi.org/10.48550/arXiv.2510.04640, https://arxiv.org/abs/2510.04640
7. Association for Computing Machinery: Acm digital library. Online Database (2025), https://dl.acm.org/, comprehensive database for publications from the ACM and affiliated organizations
8. Aumasson, J.P., Bernstein, D.J., Beullens, W., Dobraunig, C., Eichlseder, M., Fluhrer, S., Gazdag, S.L., Hülsing, A., Kampanakis, P., Kölbl, S., Kudinov, M., Lange, T., Lauridsen, M.M., Mendel, F., Niederhagen, R., Rechberger, C., Rijneveld, J., Schwabe, P., Westerbaan, B.: SPHINCS+. Submission to the NIST Post-Quantum Cryptography Standardization Process (10 2020), https://sphincs.org/data/sphincs+-r3.1-specification.pdf
9. Avanzi, R., Bos, J., Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schanck, J.M., Schwabe, P., Seiler, G., Stehlé, D.: CRYSTALS-Kyber (version 3.02). Submission to the NIST Post-Quantum Cryptography Standardization Process (08 2021), https://pq-crystals.org/kyber/data/kyber-specification-round3-20210804.pdf
10. Bai, S., Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schwabe, P., Seiler, G., Stehlé, D.: CRYSTALS-Dilithium (version 3.1). Submission to the NIST Post-Quantum Cryptography Standardization Process (08 2021), https://pq-crystals.org/dilithium/data/dilithium-specification-round3-20210208.pdf

11. Bauer, A., Jaulmes, E., Prouff, E., Wild, J.: Horizontal and vertical side-channel attacks against secure rsa implementations. In: Proceedings of the 13th International Conference on Topics in Cryptology. pp. 1–17. CT-RSA'13, Springer-Verlag, Berlin, Heidelberg (2013). https://doi.org/10.1007/978-3-642-36095-4_1
12. Benadjila, R., Prouff, E., Strullu, R., Cagli, E., Dumas, C.: Deep learning for side-channel analysis and introduction to ASCAD database. Journal of Cryptographic Engineering **10**(2), 163–188 (06 2020). https://doi.org/10.1007/s13389-019-00220-8, https://doi.org/10.1007/s13389-019-00220-8
13. Bernstein, D.J.: Cache-timing attacks on aes (2005), https://cr.yp.to/papers.html#cachetiming
14. Bernstein, D.J., Lange, T.: Post-quantum cryptography. Nature **549**(7671), 188–194 (09 2017). https://doi.org/10.1038/nature23461
15. Beullens, W.: Breaking rainbow takes a weekend on a laptop. Cryptology ePrint Archive, Paper 2022/214 (2022), https://eprint.iacr.org/2022/214
16. Bommana, S.R., et al.: Mitigating side-channel attacks on fpga through deep learning and dynamic partial reconfiguration. Scientific Reports (2025). https://doi.org/10.1038/s41598-025-98473-3
17. Brier, É., Clavier, C., Olivier, F.: Correlation power analysis with a leakage model. In: Cryptographic Hardware and Embedded Systems - CHES 2004: 6th International Workshop Cambridge, MA, USA, August 11-13, 2004. Proceedings. Lecture Notes in Computer Science, vol. 3156, pp. 16–29. Springer (2004). https://doi.org/10.1007/978-3-540-28632-5_2
18. Cagli, E., Dumas, C., Prouff, E.: Convolutional neural networks with data augmentation against jitter-based countermeasures. In: Cryptographic Hardware and Embedded Systems - CHES 2017. pp. 45–68 (2017). https://doi.org/10.1007/978-3-319-66787-4_3
19. Castryck, W., Decru, T.: An efficient key recovery attack on sidh. Cryptology ePrint Archive, Paper 2022/975 (2022), https://eprint.iacr.org/2022/975
20. Chari, S., Rao, J.R., Rohatgi, P.: Template attacks. In: Cryptographic Hardware and Embedded Systems - CHES 2002. Lecture Notes in Computer Science, vol. 2523, pp. 13–28. Springer (2002). https://doi.org/10.1007/3-540-36400-5_3
21. Chen, L., Jordan, S., Liu, Y.K., Moody, D., Peralta, R., Perlner, R., Smith-Tone, D.: Report on post-quantum cryptography. Tech. Rep. NISTIR 8105, National Institute of Standards and Technology (2016). https://doi.org/10.6028/NIST.IR.8105
22. Chen, P., Li, J., Cheng, W., Cheng, C.: Uncover secrets through the cover: A deep learning-based side-channel attack against kyber implementations with anti-tampering covers. IEEE Transactions on Computers **74**(6), 2159–2167 (2025). https://doi.org/10.1109/TC.2025.3547610
23. Choi, K.h., Han, J., Han, D.: Single trace analysis of visible vs. invisible leakage for comparison-operation-based CDT sampling. Electronics (Switzerland) **13**(23) (2024). https://doi.org/10.3390/electronics13234681
24. Deutsch, D.: Quantum theory, the church-turing principle and the universal quantum computer. Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences **400**(1818), 97–117 (1985). https://doi.org/10.1098/rspa.1985.0070
25. Dubrova, E., Ngo, K., Gärtner, J., Wang, R.: Breaking a fifth-order masked implementation of CRYSTALS-kyber by copy-paste. In: Proceedings of the 10th ACM Asia Public-Key Cryptography Workshop. pp. 10–20. APKC '23, Association

for Computing Machinery (2023). https://doi.org/10.1145/3591866.3593072, http://dx.doi.org/10.1145/3591866.3593072, journal Abbreviation: Proceedings of the 10th ACM Asia Public-Key Cryptography Workshop Published: Cryptology ePrint Archive, Paper 2022/1713 event-place: Melbourne, VIC, Australia
26. Elsevier: Engineering village. Online Database (2025), https://www.engineeringvillage.com/, comprehensive database for engineering and applied science literature
27. Elsevier: Sciencedirect. Online Database (2025), https://www.sciencedirect.com/, database of scientific, technical, and medical research
28. Elsevier: Scopus. Online Database (2025), https://www.scopus.com/, large abstract and citation database of peer-reviewed literature
29. Fouque, P.A., Hoffstein, J., Kirchner, P., Lyubashevsky, V., Pornin, T., Prest, T., Ricosset, T., Seiler, G., Whyte, W., Zhang, Z.: FALCON: Fast-fourier lattice-based compact signatures over NTRU. Submission to the NIST Post-Quantum Cryptography Standardization Process (10 2020), https://falcon-sign.info/falcon.pdf
30. Gaborit, P., Aguilar-Melchor, C., Aragon, N., Bettaieb, S., Bidoux, L., Blazy, O., Deneuville, J.C., Persichetti, E., Zémor, G., Bos, J., Dion, A., Lacan, J., Robert, J.M., Véron, P., Barreto, P.L., Ghosh, S., Gueron, S., Güneysu, T., Misoczki, R., Richter-Brokmann, J., Sendrier, N., Tillich, J.P., Vasseur, V.: Hamming quasi-cyclic (hqc) fourth round version. NIST PQC Standardization Process Submission (08 2025), https://pqc-hqc.org/doc/hqc_specifications_2025_08_22.pdf
31. Ganesh, B.S., Ahmed, M.M., Mady, A.: Higher order leakage assessment and neural network-based attack on CRYSTALS-kyber. In: Di Vimercati, S., Samarati, P. (eds.) Proceedings of the International Conference on Security and Cryptography. pp. 373 – 380. Science and Technology Publications, Lda (2024). https://doi.org/10.5220/0012715700003767
32. Gidney, C., Ekerå, M.: How to factor 2048 bit rsa integers in 8 hours using 20 million noisy qubits. Quantum **5**, 433 (2021). https://doi.org/10.22331/q-2021-04-15-433
33. Grover, L.K.: A fast quantum mechanical algorithm for database search. In: Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing. pp. 212–219 (1996). https://doi.org/10.1145/237814.237866
34. Han, J., Lee, T., Kwon, J., Lee, J., Kim, I.J., Cho, J., Han, D.G., Sim, B.Y.: Single-trace attack on NIST round 3 candidate dilithium using machine learning-based profiling. IEEE Access **9**, 166283–166292 (2021). https://doi.org/10.1109/ACCESS.2021.3135600, http://dx.doi.org/10.1109/ACCESS.2021.3135600
35. Hernandez-Alvarez, L., de la Torre, M.A.G., Hernandez, E.I., Encinas, L.H.: How to attack a galaxy: From star wars to star trek. In: 2023 Congress in Computer Science, Computer Engineering, Applied Computing (CSCE). pp. 2347–2354. IEEE Computer Society, Los Alamitos, CA, USA (07 2023). https://doi.org/10.1109/CSCE60160.2023.00381
36. Hoang, A.T., Kennaway, M., Pham, D.T., Mai, T.S., Khalid, A., Rafferty, C., O'Neill, M.: Deep learning enhanced side channel analysis on CRYSTALS-kyber. In: 2024 25th International Symposium on Quality Electronic Design (ISQED). pp. 1–8 (2024). https://doi.org/10.1109/ISQED60706.2024.10528674, ISSN: 19483287 Journal Abbreviation: Proceedings - International Symposium on Quality Electronic Design, ISQED
37. Hospodar, G., Gierlichs, B., De Mulder, E., Verbauwhede, I., Vandewalle, J.: Machine learning in side-channel analysis: a first study. Journal of Cryp-

tographic Engineering **1**(4), 293–302 (12 2011). https://doi.org/10.1007/s13389-011-0023-x, https://doi.org/10.1007/s13389-011-0023-x
38. Huang, Z., Wang, H., Cao, B., He, D., Wang, J.: A comprehensive side-channel leakage assessment of CRYSTALS-kyber in IIoT. Internet of Things **27**, 101331 (2024). https://doi.org/https://doi.org/10.1016/j.iot.2024.101331
39. Iavich, M., Gnatyuk, S., Mukasheva, A.: Decoding the CRYSTALS-kyber attack using artificial intelligence: Examination and strategies for resilience. In: Sokolov, V., Ustimenko, V., Radivilova, T., Nazarkevych, M. (eds.) Proceedings of the Workshop on Cybersecurity Providing in Information and Telecommunication Systems II (CPITS II 2024). CEUR Workshop Proceedings, vol. 3826, pp. 342–349. Kyiv, Ukraine (online) (2024), https://ceur-ws.org/Vol-3826/short26.pdf
40. Institute of Electrical and Electronics Engineers: Ieee xplore digital library. Online Database (2025), https://ieeexplore.ieee.org/, primary database for publications in electrical engineering and computer science
41. International Association for Cryptologic Research: Cryptology eprint archive (2024), https://eprint.iacr.org/, primary repository for pre-print cryptographic research.
42. Jao, D., De Feo, L.: Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. In: Yang, B.Y. (ed.) Post-Quantum Cryptography. pp. 19–34. Springer Berlin Heidelberg, Berlin, Heidelberg (2011)
43. Jendral, S., Ngo, K., Wang, R., Dubrova, E.: Breaking SCA-protected CRYSTALS-kyber with a single trace. In: 2024 IEEE International Symposium on Hardware Oriented Security and Trust (HOST). pp. 70–73 (2024). https://doi.org/10.1109/HOST55342.2024.10545390, ISSN: 2765-8406 Journal Abbreviation: Proceedings of the 2024 IEEE International Symposium on Hardware Oriented Security and Trust, HOST 2024 Published: Cryptology ePrint Archive, Paper 2023/1587
44. Ji, Y., Dubrova, E.: A side-channel attack on a masked hardware implementation of crystals-kyber. Journal of Cryptographic Engineering **15**(1), 7 (04 2025). https://doi.org/10.1007/s13389-025-00375-7
45. Ji, Y., Wang, R., Ngo, K., Dubrova, E., Backlund, L.: A side-channel attack on a hardware implementation of CRYSTALS-kyber. In: 2023 IEEE European Test Symposium (ETS). pp. 1–5 (2023). https://doi.org/10.1109/ETS56758.2023.10174000, ISSN: 15301877 Journal Abbreviation: Proceedings of the European Test Workshop Published: Cryptology ePrint Archive, Paper 2022/1452
46. Karabulut, E., Aysu, A.: Falcon down: Breaking falcon post-quantum signature scheme through side-channel attacks. In: 2021 58th ACM/IEEE Design Automation Conference (DAC). pp. 691–696 (2021), https://ieeexplore.ieee.org/document/9586131
47. Kim, H., Kim, T.H., Yoon, J.C., Hong, S.: Practical second-order correlation power analysis on the message blinding method and its novel countermeasure for rsa. ETRI Journal **32**(1), 102–111 (2010). https://doi.org/https://doi.org/10.4218/etrij.10.0109.0249, https://onlinelibrary.wiley.com/doi/abs/10.4218/etrij.10.0109.0249
48. Kim, I.J., Lee, T.H., Han, J., Sim, B.Y., Han, D.G.: Novel single-trace ML profiling attacks on NIST 3 round candidate dilithium (2020), https://eprint.iacr.org/2020/1383, published: Cryptology ePrint Archive, Paper 2020/1383
49. Kim, J., Picek, S., Heo, A., Taha, M., Choi, D.G.: Make some noise. unleashing the power of convolutional neural networks for profiled side-channel analysis. IACR

Transactions on Cryptographic Hardware and Embedded Systems **2019**(3), 148–179 (2019). https://doi.org/10.13154/tches.v2019.i3.148-179
50. Kitchenham, B., Charters, S.: Guidelines for performing systematic literature reviews in software engineering. Tech. Rep. EBSE-2007-01, Keele University and University of Durham (2007), https://legacyfileshare.elsevier.com/promis_misc/525444systematicreviewsguide.pdf
51. Kitchenham, B., Pearl Brereton, O., Budgen, D., Turner, M., Bailey, J., Linkman, S.: Systematic literature reviews in software engineering – a systematic literature review. Information and Software Technology **51**(1), 7–15 (2009). https://doi.org/https://doi.org/10.1016/j.infsof.2008.09.009, https://www.sciencedirect.com/science/article/pii/S0950584908001390, special Section - Most Cited Articles in 2002 and Regular Research Papers
52. Kocher, P.C.: Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems. In: Advances in Cryptology — CRYPTO '96. pp. 104–113. Springer (1996). https://doi.org/10.1007/3-540-68697-5_9
53. Kocher, P.C., Jaffe, J., Jun, B.: Differential power analysis. In: Advances in Cryptology — CRYPTO' 99. Lecture Notes in Computer Science, vol. 1666, pp. 388–397. Springer (1999). https://doi.org/10.1007/3-540-48405-1_25
54. Koeune, F., Standaert, F.X.: A Tutorial on Physical Security and Side-Channel Attacks, pp. 78–108. Springer Berlin Heidelberg, Berlin, Heidelberg (2005). https://doi.org/10.1007/11554578_3, https://doi.org/10.1007/11554578_3
55. Lerman, L., Poussier, R., Markowitch, O., Standaert, F.X.: Template attacks versus machine learning revisited and the curse of dimensionality in side-channel analysis: extended version. Journal of Cryptographic Engineering **8**(4), 301–313 (11 2018). https://doi.org/10.1007/s13389-017-0162-9, https://doi.org/10.1007/s13389-017-0162-9
56. Li, S.: Overview and discussion of attacks on CRYSTALS-kyber (2023), https://eprint.iacr.org/2023/1952, published: Cryptology ePrint Archive, Paper 2023/1952
57. Maghrebi, H., Portigliatti, T., Prouff, E.: Breaking cryptographic implementations using deep learning techniques. In: Security, Privacy, and Applied Cryptography Engineering - SPACE 2016. Lecture Notes in Computer Science, vol. 10076, pp. 3–26. Springer (2016). https://doi.org/10.1007/978-3-319-49445-6_1
58. Marzougui, S., Ulitzsch, V., Tibouchi, M., Seifert, J.P.: Profiling side-channel attacks on dilithium: A small bit-fiddling leak breaks it all (2022), https://eprint.iacr.org/2022/106, published: Cryptology ePrint Archive, Paper 2022/106
59. Mascelli, J., Rodden, M.: "harvest now, decrypt later": Examining post-quantum cryptography and the data privacy risks for distributed ledger networks. Tech. Rep. FEDS 2025-093, Federal Reserve Board, Washington, D.C. (09 2025). https://doi.org/10.17016/FEDS.2025.093, finance and Economics Discussion Series (FEDS 2025-093)
60. Masure, L., Dumas, C., Prouff, E.: A comprehensive study of deep learning for side-channel analysis. Cryptology ePrint Archive, Paper 2019/439 (2019), https://eprint.iacr.org/2019/439
61. McEliece, R.J.: A public-key cryptosystem based on algebraic coding theory. DSN Progress Report **42-44**, 114–116 (1978), https://tmo.jpl.nasa.gov/progress_report/42-44/44N.PDF
62. Mosca, M.: Cybersecurity in an era with quantum computers: Will we be ready? IEEE Security & Privacy **16**(5), 38–41 (2018). https://doi.org/10.1109/MSP.2018.3761723

63. National Institute of Standards and Technology: Post-quantum cryptography, https://csrc.nist.gov/projects/post-quantum-cryptography
64. National Institute of Standards and Technology: Post-quantum cryptography standardization, https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization
65. National Institute of Standards and Technology: Evaluation criteria. NIST PQC Standardization Website (2016), https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/evaluation-criteria
66. National Institute of Standards and Technology: Security (evaluation criteria). NIST PQC Standardization Website (2016), https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/evaluation-criteria/security-(evaluation-criteria)
67. National Institute of Standards and Technology: Module-lattice-based digital signature standard. Tech. Rep. FIPS 204, National Institute of Standards and Technology (2024). https://doi.org/10.6028/NIST.FIPS.204
68. National Institute of Standards and Technology: Module-lattice-based key-encapsulation mechanism standard. Tech. Rep. FIPS 203, National Institute of Standards and Technology (2024). https://doi.org/10.6028/NIST.FIPS.203
69. National Institute of Standards and Technology: Nist releases first 3 finalized post-quantum encryption standards. NIST News (8 2024), https://www.nist.gov/news-events/news/2024/08/nist-releases-first-3-finalized-post-quantum-encryption-standards
70. National Institute of Standards and Technology: Stateless hash-based digital signature standard. Tech. Rep. FIPS 205, National Institute of Standards and Technology (2024). https://doi.org/10.6028/NIST.FIPS.205
71. National Institute of Standards and Technology: Nist pqc standardization process: Hqc announced as a 4th round selection. NIST News (3 2025), https://www.nist.gov/news-events/news/2025/03/nist-pqc-standardization-process-hqc-announced-4th-round-selection
72. National Institute of Standards and Technology: Nist selects hqc as fifth algorithm for post-quantum encryption. NIST News (3 2025), https://www.nist.gov/news-events/news/2025/03/nist-selects-hqc-fifth-algorithm-post-quantum-encryption, states that a draft of the fourth standard, built around the FALCON algorithm, will be released shortly as FIPS 206.
73. Ngo, K., Dubrova, E., Johansson, T.: A side-channel attack on a masked and shuffled software implementation of saber. Journal of Cryptographic Engineering **13**(4), 443–460 (11 2023). https://doi.org/10.1007/s13389-023-00315-3
74. Nielsen, M.A., Chuang, I.L.: Quantum Computation and Quantum Information: 10th Anniversary Edition. Cambridge University Press, Cambridge (2010)
75. PQShield: Nist selects hqc for standardization. PQShield Blog (2025), https://pqshield.com/nist-selects-hqc-for-standardization/, discusses the selection of HQC over BIKE and Classic McEliece.
76. Proos, J., Zalka, C.: Shor's discrete logarithm quantum algorithm for elliptic curves. Quantum Info. Comput. **3**(4), 317–344 (Jul 2003), https://www.rintonpress.com/xqic3/qic-3-4/317-344.pdf
77. Qiao, K., Wang, Z., Chang, H., Sun, S., Wu, Z., Cheng, J., Ou, C., Wang, A., Zhu, L.: A closer look at the belief propagation algorithm in side-channel attack on cca-secure pqc kem. Science China Information Sciences **67**(11), 212302 (10 2024). https://doi.org/10.1007/s11432-024-4150-3

78. Qiao, Z., Liu, Y., Zhou, Y., Zhao, Y., Chen, S.: Single trace is all it takes: Efficient side-channel attack on dilithium (2024), https://eprint.iacr.org/2024/512, published: Cryptology ePrint Archive, Paper 2024/512
79. Qiao, Z., Liu, Y., Zhou, Y., Zhao, Y., Yuan, H., Du, D.: Efficient CNN-based side-channel attacks on dilithium without device access. In: 2025 IEEE International Symposium on Circuits and Systems (ISCAS). pp. 1–5 (2025). https://doi.org/10.1109/ISCAS56072.2025.11043271, ISSN: 02714310 Journal Abbreviation: Proceedings - IEEE International Symposium on Circuits and Systems
80. Ravi, P., Chattopadhyay, A., D'Anvers, J.P., Baksi, A.: Side-channel and fault-injection attacks over lattice-based post-quantum schemes (kyber, dilithium): Survey and new results. ACM Trans. Embed. Comput. Syst. **23**(2) (Mar 2024). https://doi.org/10.1145/3603170
81. Ravi, P., Jap, D., Bhasin, S., Chattopadhyay, A.: Invited paper: Machine learning based blind side-channel attacks on PQC-based KEMs - a case study of kyber KEM. In: 2023 IEEE/ACM International Conference on Computer Aided Design (ICCAD). pp. 01–07 (2023). https://doi.org/10.1109/ICCAD57390.2023.10323721, ISSN: 10923152 Journal Abbreviation: IEEE/ACM International Conference on Computer-Aided Design, Digest of Technical Papers, ICCAD Published: Cryptology ePrint Archive, Paper 2024/169
82. Rezaeezade, A., Yap, T., Jap, D., Bhasin, S., Picek, S.: Breaking the blindfold: deep learning-based blind side-channel analysis (2025), https://dl.acm.org/doi/10.5555/3766078.3766375
83. Rivain, M., Prouff, E.: Provably secure higher-order masking of aes. In: Mangard, S., Standaert, F.X. (eds.) Cryptographic Hardware and Embedded Systems, CHES 2010. pp. 413–427. Springer Berlin Heidelberg, Berlin, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15031-9_28
84. Rostovtsev, A., Stolbunov, A.: Public-key cryptosystem based on isogenies. IACR Cryptol. ePrint Arch. p. 145 (2006), http://eprint.iacr.org/2006/145
85. Sajadi, A., Zidaric, N., Stefanov, T., Mentens, N.: A systematic comparison of side-channel countermeasures for risc-v-based socs. In: 2024 IEEE Nordic Circuits and Systems Conference (NorCAS). pp. 1–7 (2024). https://doi.org/10.1109/NorCAS64408.2024.10752477
86. Schneier, B.: Sike broken. Schneier on Security Blog (8 2022), https://www.schneier.com/blog/archives/2022/08/sike-broken.html
87. SecurityWeek: Nist post-quantum algorithm finalist cracked using classical pc. SecurityWeek News (2022), https://www.securityweek.com/nist-post-quantum-algorithm-finalist-cracked-using-classical-pc/
88. Shor, P.W.: Algorithms for quantum computation: Discrete logarithms and factoring. In: Proceedings of the 35th Annual Symposium on Foundations of Computer Science. pp. 124–134. IEEE Computer Society Press (1994)
89. Shor, P.W.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM Journal on Computing **26**(5), 1484–1509 (1997). https://doi.org/10.1137/S0097539795293172
90. Springer: Springer link. Online Database (2025), https://link.springer.com/, database for scientific journals, books, and proceedings
91. Srivastava, A., Das, S., Choudhury, N., Psiakis, R., Silva, P.H., Pal, D., Basu, K.: SCAR: Power side-channel analysis at RTL level. IEEE Transactions on Very Large Scale Integration (VLSI) Systems **32**(6), 1110–1123 (2024). https://doi.org/10.1109/TVLSI.2024.3390601

92. of Standards, N.I., Technology: Submission requirements and evaluation criteria for the post-quantum cryptography standardization process. Federal Register Notice (12 2016), https://csrc.nist.gov/csrc/media/projects/post-quantum-cryptography/documents/call-for-proposals-final-dec-2016.pdf
93. Tanaka, Y., Ueno, R., Xagawa, K., Ito, A., Takahashi, J., Homma, N.: Multiple-valued plaintext-checking side-channel attacks on post-quantum KEMs. IACR Transactions on Cryptographic Hardware and Embedded Systems **2023**(3), 473 – 503 (2023), https://eprint.iacr.org/2022/940, published: Cryptology ePrint Archive, Paper 2022/940
94. Tena-Sánchez, E., Potestad-Ordóñez, F.E., Jiménez-Fernández, C.J., Acosta, A.J., Chaves, R.: Gate-level hardware countermeasure comparison against power analysis attacks. Applied Sciences **12**(5) (2022). https://doi.org/10.3390/app12052390, https://www.mdpi.com/2076-3417/12/5/2390
95. Tena-Sánchez, E., Potestad-Ordóñez, F.E., Zúñiga-González, V., Acosta, A.J.: Low-cost full correlated-power-noise generator to counteract side-channel attacks. Applied Sciences **15**(6) (2025). https://doi.org/10.3390/app15063064, https://www.mdpi.com/2076-3417/15/6/3064
96. Teske, E.: Nist pqc finalists update: It's over for the rainbow. Cryptomathic Blog (3 2022), https://www.cryptomathic.com/blog/nist-pqc-finalists-update-its-over-for-the-rainbow
97. The White House: Report on post-quantum cryptography (7 2024), https://bidenwhitehouse.archives.gov/wp-content/uploads/2024/07/REF_PQC-Report_FINAL_Send.pdf
98. Ueno, R., Xagawa, K., Tanaka, Y., Ito, A., Takahashi, J., Homma, N.: Curse of re-encryption: A generic power/EM analysis on post-quantum KEMs. IACR Transactions on Cryptographic Hardware and Embedded Systems **2022**(1), 296 – 322 (2021), https://eprint.iacr.org/2021/849, published: Cryptology ePrint Archive, Paper 2021/849
99. Veyrat-Charvillon, N., Medwed, M., Kerckhof, S., Standaert, F.X.: Shuffling against side-channel attacks: A comprehensive study with cautionary note. In: Advances in Cryptology – ASIACRYPT 2012. pp. 740–757. Springer Berlin Heidelberg, Berlin, Heidelberg (2012)
100. Wang, J., Cao, W., Chen, H., Li, H.: Practical side-channel attack on masked message encoding in latticed-based KEM (2022), https://eprint.iacr.org/2022/859, published: Cryptology ePrint Archive, Paper 2022/859
101. Wang, R., Gärtner, J., Dubrova, E.: Decompressing dilithium's public key with fewer signatures using side channel analysis. In: 2025 IEEE 55th International Symposium on Multiple-Valued Logic (ISMVL). pp. 135–140 (2025). https://doi.org/10.1109/ISMVL64713.2025.00034, ISSN: 0195623X Journal Abbreviation: Proceedings of The International Symposium on Multiple-Valued Logic Published: Cryptology ePrint Archive, Paper 2024/2046
102. Wang, R., Ngo, K., Gärtner, J., Dubrova, E.: Single-trace side-channel attacks on CRYSTALS-dilithium: Myth or reality? (2023), https://eprint.iacr.org/2023/1931, published: Cryptology ePrint Archive, Paper 2023/1931
103. Wang, Y., Huang, F., Duan, X., Hu, H.: Second-order side-channel attacks on kyber: Targeting the masked hash function. Journal of Cryptologic Research **11**(6), 1415 – 1436 (2024). https://doi.org/10.13868/j.cnki.jcr.000745
104. Wang, Z., Ding, Y., Wang, A., Zhang, Y., Wei, C., Sun, S., Zhu, L.: Spa-gpt: General pulse tailor for simple power analysis based on reinforcement learning.

IACR Transactions on Cryptographic Hardware and Embedded Systems **2024**(4), 40–83 (09 2024). https://doi.org/10.46586/tches.v2024.i4.40-83
105. Wikipedia: Post-quantum cryptography. Online Encyclopedia (2025), https://en.wikipedia.org/wiki/Post-quantum_cryptography
106. Wikipedia: Side-channel attack. Online Encyclopedia (2025), https://en.wikipedia.org/wiki/Side-channel_attack
107. Xu, D., Wang, K., Tian, J.: A hardware-friendly shuffling countermeasure against side-channel attacks for kyber. IEEE Transactions on Circuits and Systems II: Express Briefs **72**(3), 504–508 (2025). https://doi.org/10.1109/TCSII.2025.3528751
108. Yang, Y., Huang, J., Wang, Z., Ye, J., Sun, Z., Fan, J., Chen, S., Li, H., Li, X., Cao, Y.: A template attack on reduction without reference device on kyber. In: 2023 IEEE 32nd Asian Test Symposium (ATS). pp. 1–6 (2023). https://doi.org/10.1109/ATS59501.2023.10318019, ISSN: 10817735 Journal Abbreviation: Proceedings of the Asian Test Symposium
109. You, S.C.: Single-trace template attacks on permutation-based cryptography. Ph.D. thesis, University of Cambridge (12 2022), https://www.cl.cam.ac.uk/~scy27/PhD_thesis.pdf
110. Zhang, H., Babar, M.A., Tell, P.: Identifying relevant studies in software engineering. Inf. Softw. Technol. **53**(6), 625–637 (Jun 2011). https://doi.org/10.1016/j.infsof.2010.12.010, https://doi.org/10.1016/j.infsof.2010.12.010
111. Zhou, W., Wang, A., Ding, Y., Wei, C., Zhang, J., Zhu, L.: One solves all: Exploring ChatGPT's capabilities for fully automated simple power analysis on cryptosystems (2024), https://eprint.iacr.org/2024/2069, published: Cryptology ePrint Archive, Paper 2024/2069

Table of Contents