Sensitivity Analysis of Variational Quantum Classifiers for Identifying Dummy Power Traces in Side-Channel Analysis

Seungun Park and Yunsik Son *

Posted Date: 4 March 2026

doi: 10.20944/preprints202603.0185.v1

Keywords: quantum machine learning; quantum neural network; variational quantum classifier; side-channel analysis; pattern recognition

 doi:10.20944/preprints202603.0185.v1

Article

Sensitivity Analysis of Variational Quantum Classifiers for Identifying Dummy Power Traces in Side-Channel Analysis

Seungun Park and Yunsik Son *

Department of Computer Science and Artificial Intelligence, Dongguk University, Seoul 04620, Republic of Korea
* Correspondence: sonbug@dongguk.edu

Abstract

The application of quantum machine learning (QML) to security-relevant problems has attracted growing attention, yet its practical behavior in realistic workloads remains insufficiently characterized. This paper investigates the feasibility and limitations of variational quantum classifiers (VQCs) in a representative side-channel analysis (SCA) setting focused on distinguishing real from dummy power traces. A controlled benchmark framework is developed to examine training stability, sensitivity to key design parameters, and resource–performance trade-offs under realistic constraints. Hardware-relevant factors, including finite measurement budgets and device noise, are incorporated to approximate execution beyond idealized simulation and to quantify inference-time robustness. Experimental results indicate that VQCs can extract meaningful discriminative patterns from structured side-channel inputs, while robustness and performance depend on encoding choices, circuit depth, and measurement conditions. These findings provide an empirically grounded perspective on the applicability and limitations of QML in side-channel security and offer practical guidance for future research in this area.

Keywords: quantum machine learning; quantum neural network; variational quantum classifier; side-channel analysis; pattern recognition

1. Introduction

Quantum Machine Learning (QML) has emerged as a promising research direction at the intersection of quantum computing and data-driven modeling [1,2]. As quantum hardware advances toward larger, more reliable devices, QML is often viewed as a promising methodology that may eventually complement or accelerate certain learning tasks by leveraging quantum-state representations and measurements. At the same time, the current landscape is dominated by noisy intermediate-scale quantum (NISQ) devices, where limited qubit counts, finite sampling budgets, and hardware noise impose nontrivial constraints on practical learning performance [3]. These constraints motivate a careful, empirical understanding of what QML can and cannot do today, and under which design and resource conditions it can become useful for real-world applications [4].

Learning-based methods are increasingly shaping modern information security [5–7]. Side-channel analysis (SCA) remains a major threat to cryptographic implementations [8–10]. SCA exploits unintended physical leakage, such as power consumption, electromagnetic emission, or execution time, to infer sensitive information or to distinguish internal execution states. In response, many countermeasures aim to reduce exploitable leakage by obfuscating observable behavior, for example, by inserting dummy operations or otherwise randomizing execution patterns [11]. However, the effectiveness of such countermeasures relies crucially on whether an adversary can still separate real traces from dummy traces in the resulting traces. If dummy and real operations are distinguishable at the signal level, an attacker may recover a more informative subset of trace segments, possibly undermining the intended protection.

 doi:10.20944/preprints202603.0185.v1

While QML has been explored across a range of pattern recognition and classification settings, its use in SCA-related tasks remains comparatively less established and, more importantly, insufficiently characterized under realistic NISQ constraints [6,12]. Side-channel traces pose challenges that differ from many standard benchmark datasets: they are noisy, often non-stationary, and exhibit a time-ordered structure in which subtle local differences can carry security-relevant information. Moreover, applying variational quantum algorithms (VQAs), one of the most widely used QML paradigms on NISQ hardware, to such data raises practical questions that go beyond the choice of a model family [2,4,13–15]. In particular, performance can be highly sensitive to classical-to-quantum data encoding, normalization ranges, quantum circuit depth, optimizer dynamics, and the measurement budget required to obtain stable gradients and predictions. Without a systematic evaluation of these factors, it is difficult to assess whether QML-based approaches are merely conceptually appealing or genuinely actionable in SCA contexts.

This study presents a practical benchmark investigation of applying a variational quantum classifier (VQC) to distinguish dummy traces from real traces in SCA using power measurements. The VQC is adopted as a canonical QML baseline to examine feasibility, training stability, sensitivity to design parameters, and resource–performance trade-offs under NISQ constraints [2,4]. Experimental results indicate that VQC models can extract discriminative patterns from side-channel inputs and achieve meaningful classification performance. Because robustness under degraded oprating conditions is a recurring concern across machine learning (ML) domains [16], this paper explicitly examine when VQC training remains stable, when degradation emerges under practical execution conditions, and which architectural or optimization choices most strongly influence model behavior.

To that end, a structured exploration of the VQC design space and deployment-relevant constraints on real quantum devices is conducted. First, the effect of the feature scaling range on learnability in normalization is analyzed, motivated by the fact that typical encodings, such as rotation-angle encodings, directly map normalized values to quantum gate parameters. Second, the impact of circuit depth, controlled via the number of repeated layers, is examined in terms of expressivity versus trainability, including the potential for optimization difficulties as circuits deepen. Third, multiple optimizers commonly used within variational algorithms are compared to highlight differences during convergence behavior and robustness. Finally, finite-shot budgets and a hardware-informed noise model are investigated as unavoidable deployment factors. In the experiments, these effects are quantified primarily through repeated inference runs with fixed trained parameters to isolate execution-induced variability. Limited shots introduce estimation variance in measured expectations, and device noise can distort both the training signal and the resulting decision boundary. Performance trends across these dimensions provide an empirically grounded perspective on QML behavior in a security-relevant, noisy, and structured dataset setting.

Overall, this work serves as a feasibility assessment of VQC-based learning on side-channel inputs, a controlled benchmark framework for examining experimentally relevant design factors, and a set of practical guidelines for applying QML to SCA settings, including the evaluation of countermeasures.

The contributions of this work are as follows:

- Feasibility of QML for side-channel dummy detection: A canonical VQC is applied to power-based side-channel trace instances, demonstrating empirically that a small VQC can learn discriminative patterns that distinguish real operations from dummy operations.
- Practical considerations for security-oriented VQCs under NISQ constraints: A controlled sensitivity analysis is conducted over normalization and encoding ranges, circuit depth, optimizer selection, and training-data availability. The impact of finite-shot budgets and a hardware-informed noise model on inference robustness is also quantified. Based on these results, configuration-aware guidelines and limitations for applying QML to side-channel security settings are provided.

 doi:10.20944/preprints202603.0185.v1

2. Related Works

2.1. Learning-Based Side-Channel Analysis

SCA exploits unintended physical leakage, such as power consumption, electromagnetic emission, or execution time, to infer internal states of a computation or recover secrets from cryptographic implementations. Classical SCA includes simple and differential analyses, which correlate measured power with key-dependent intermediate values to extract secret keys [8]. Established foundations describe SCA threat models, leakage sources, and evaluation methodologies [9,10]. Beyond traditional key-recovery attacks, modern SCA increasingly encompasses broader security objectives, including program behavior inference and reverse engineering from side-channel signals.

Over the last several years, learning-based approaches have become a central pillar in SCA. In profiling settings, deep learning (DL) models learn discriminative representations directly from traces, often reducing the need for handcrafted feature extraction and enabling attacks under more realistic noise and variability conditions. A systematic consolidation of this trend is provided by the systematization of knowledge (SoK) literature on DL-based SCA (DL-SCA), which categorizes how learning is used across the attack pipeline and shows both capabilities and limitations of DL-based adversaries [6,12]. Early end-to-end profiling strategies based on convolutional neural networks (CNNs) demonstrated that DL can be effective without explicit alignment or manual point-of-interest selection, and that augmentation strategies can improve robustness against jitter-style misalignment [17]. Subsequent work further investigated architectural and methodological choices for DL-SCA, proposing systematic methodologies for CNN design and evaluation under common SCA constraints [18]. These results collectively suggest that learning-based models are well-suited to the noisy, high-dimensional, and temporally structured characteristics of side-channel traces.

A closely related line of research extends DL-SCA from key recovery to side-channel-based disassembly (SCBD), where the goal is to infer program-level information, such as executed instructions, control flow, and higher-level behaviors, from leakage. This direction is security-relevant for malware analysis, firmware and binary reverse engineering, and the evaluation of software confidentiality in embedded and IoT deployments. Early studies showed the feasibility of inferring instruction-level behavior from power leakage for Internet of Things (IoT) devices and proposed instruction-level disassembly approaches leveraging power traces [19–22]. Later work expanded the scope by addressing instruction sequence identification on pipelined platforms and highlighting robustness considerations [23–26]. SCBD has also been explored at design time, where simulated traces are used to assess vulnerability prior to fabrication, enabling security feedback earlier in the hardware design process [27]. Practical feasibility has further been studied on complex platforms such as systems-on-chip, emphasizing challenges in real measurement settings and the granularity of recoverable information [28].

Recent SCBD-oriented studies emphasize the importance of realistic datasets and systematic evaluation frameworks. In a fully simulated IoT environment, feature engineering techniques based on rolling windows have been proposed, including moving log-transformed temporal interaction features, which improve both countermeasure detection and instruction inference and demonstrate strong performance in these tasks [29]. Complementary to direct disassembly, contextual leakage modeling attempts to predict power traces from assembly-level information. A context-aware DL framework has been introduced to generate instruction-level power traces from assembly code, supporting automated and scalable leakage evaluation and enabling downstream SCBD analyses [30,31]. These studies indicate that learning-based methods can capture fine-grained relationships between execution semantics and observed leakage, motivating careful evaluation of how modeling choices affect robustness and generalization.

2.2. Countermeasures and Their Distinguishability

Defending against SCA usually involves reducing exploitable leakage or disrupting an attacker's ability to align and correlate traces. Classical countermeasure families include masking, which

 doi:10.20944/preprints202603.0185.v1

randomizes sensitive intermediate values to decorrelate leakage from secrets, and hiding, which aims to obscure leakage by introducing temporal jitter, additional noise, or execution randomization [9]. Related work has also explored countermeasures across leakage modalities, where both measurement-side and implementation-side defenses are relevant [10]. Even when the leakage source cannot be eliminated, many practical defenses attempt to increase the effective difficulty of analysis by perturbing the observable signal structure.

Among hiding techniques, dummy operation insertion and shuffling are commonly used to randomize execution and reduce alignment. In particular, dummy insertion aims to increase the attacker's complexity by interleaving real operations with non-secret-dependent operations. However, the security benefit is critically dependent on distinguishability. If an adversary can reliably separate real and dummy segments, they can filter traces, re-establish alignment, or recover informative subsets, undermining the intended protection. Vulnerabilities have been identified in which dummy and real operations remain distinguishable across multiple implementation methods and compiler optimization levels. Case studies on AES insertion and shuffling demonstrate that these hiding techniques may still leak distinguishable side-channel patterns, and corresponding countermeasure strategies have been explored [11]. Learning-based approaches have also been developed to detect dummy operations directly from side-channel signals. Dummy detection can be formulated as a supervised learning problem, enabling automated classification to identify dummy operations. Experimental results show that dummy insertion can be effectively attacked using DL-based classifiers [32]. These results motivate treating distinguishability not as an incidental artifact, but as a primary security metric for evaluating hiding schemes.

DL strengthens this perspective by enabling models to capture subtle and localized leakage patterns that persist even under intentional obfuscation. Prior work has demonstrated that convolutional neural network-based profiling attacks with data augmentation can weaken misalignment-based countermeasures by learning invariance to jitter and temporal perturbations [17]. Systematic evaluations have shown that architectural and methodological choices significantly affect the robustness of DL-SCA under common defense mechanisms. These results highlight the importance of model design and training methodology when evaluating attack effectiveness [18]. Recent studies focusing on dummy-based hiding have shown that models trained on datasets augmented with generated dummy traces can effectively distinguish genuine and dummy operations [33]. Learning-based pipelines have also demonstrated the ability to recover execution semantics and detect hiding schemes from side-channel signals in simulated environments. These results suggest that hiding countermeasures can be evaluated and potentially bypassed using data-driven analysis [29]. Overall, prior work indicates that distinguishing dummy and real operations is both practically relevant and closely related to the effectiveness of hiding countermeasures under realistic conditions.

2.3. Quantum Machine Learning on NISQ Hardware

QML investigates how quantum computation methods can be used to perform or accelerate learning tasks. To situate QML in a security-oriented audience, it is useful to recall a few operational concepts. Quantum computation manipulates qubits, which can exist within superpositions of basis states and can be correlated through entanglement. Computation is implemented as a sequence of quantum gates, and information is extracted by measurement. Importantly, measurement results are probabilistic. Therefore, estimating quantities such as expectation values typically requires repeated circuit executions, often referred to as shots. Current devices are commonly described as NISQ hardware, where limited qubit counts and non-negligible noise constrain circuit depth and algorithmic reliability [3].

In the long term, fault-tolerant quantum computing (FTQC) is expected to mitigate noise through quantum error correction [34–36]. Foundational proposals include quantum error-correcting codes and fault-tolerance frameworks, as well as topological approaches that aim to encode information in more noise-resilient degrees of freedom [37–39]. However, near-term QML research largely operates without full error correction and must therefore explicitly account for finite sampling and hardware noise. This

 doi:10.20944/preprints202603.0185.v1

makes it essential to characterize not only algorithmic expressivity but also resource-performance trade-offs and training stability in realistic conditions.

Within QML, supervised learning can be implemented via various paradigms. Standard references provide a broad conceptual and algorithmic introduction to supervised learning on quantum computers and discuss key design choices such as data encoding and measurement strategies [2]. A widely cited review further contextualizes QML as a field, summarizing algorithmic building blocks and the challenges that arise in connecting theoretical speedups and practical implementations [1]. On NISQ devices, one of the most prevalent practical approaches is VQAs, which use parameterized quantum circuits (PQCs) optimized by a classical optimizer in a hybrid loop [4]. VQCs can be viewed as VQA instances designed for classification, where classical inputs are embedded into quantum circuits and measurement results define decision functions. Related work on quantum circuit learning provides early formulations of hybrid training for near-future devices and clarifies how parameterized circuits can represent learnable models [40]. Quantum feature-space methods, including quantum-enhanced feature maps, further motivate the use of quantum circuits as nonlinear embeddings for supervised learning [13].

Variational models face challenges during training. One significant issue is the emergence of barren plateaus, where gradients vanish exponentially with increasing system size for certain circuit families and initialization choices, leading to optimization stagnation [14,15]. In addition, finite-shot estimation introduces stochasticity into objective evaluations and gradients, and device noise can distort both training signals and predictions. Consequently, practical performance on NISQ hardware depends strongly on choices, such as data normalization and encoding ranges, circuit depth, optimizer type, and shot budgets, factors that often matter less in classical deterministic inference.

QML has begun to attract attention in broader cybersecurity applications, including efforts to assess whether QML could impact standard machine-learning pipelines used in security tasks and to clarify the conditions under which quantum advantage may become reasonable [7,41]. Nevertheless, the application of QML to physical side-channel workloads remains comparatively underexplored and insufficiently characterized under NISQ constraints. Side-channel traces are noisy, non-stationary, and time-structured, and their security relevance depends on subtle differences that may be affected by both encoding choices and quantum measurement noise. This motivates controlled benchmark studies that explicitly probe how QML design parameters and resource constraints interact with SCA datasets.

2.4. Summary of Research Gap

Prior work shows that learning-based methods can extract meaningful information from physical leakage in both key-recovery and program-inference settings, and that SCBD has matured into a practical reverse-engineering and evaluation toolchain [6,19,27,28,42]. At the same time, countermeasure effectiveness increasingly depends on distinguishability: hiding defenses such as dummy insertion can fail if an attacker learns to separate dummy from real behaviors [11,32,33]. These trends establish a clear security motivation for studying dummy detection as a benchmark problem, especially under realistic noise and alignment conditions.

In parallel, QML has developed a strong methodological foundation, featuring variational quantum models emerging as a practical tool family for near-term supervised learning [2–4]. However, the gap lies in the intersection: there is limited empirical characterization of how VQCs behave on side-channel datasets, where performance may be dominated by encoding ranges, circuit depth, trainability, optimizer stability, and measurement budgets. Moreover, hardware-relevant effects such as finite shots and noise are not optional details but core determinants of learnability in NISQ settings.

Therefore, a key open problem is the development of a controlled evaluation framework that bridges security-relevant side-channel workloads and the distinguishability between dummy and real side-channel traces by QML models under the practical constraints and sensitivities of NISQ-era QML systems. Addressing this gap can clarify what QML can reliably learn from side-channel inputs, which design regimes are stable, and what resource costs are required for meaningful performance.

 doi:10.20944/preprints202603.0185.v1

3. Methodology

3.1. Scope and Benchmark Objective

This work is designed as a controlled benchmark study to characterize how the behavior of a VQC changes under practical design choices and resource constraints. This paper treats a representative binary side-channel classification task as a convenient workload and systematically examines how feature scaling ranges, circuit depth, optimizer dynamics, and sampling and noise conditions affect training robustness and predictive performance. Figure 1 shows the end-to-end workflow of the benchmark.

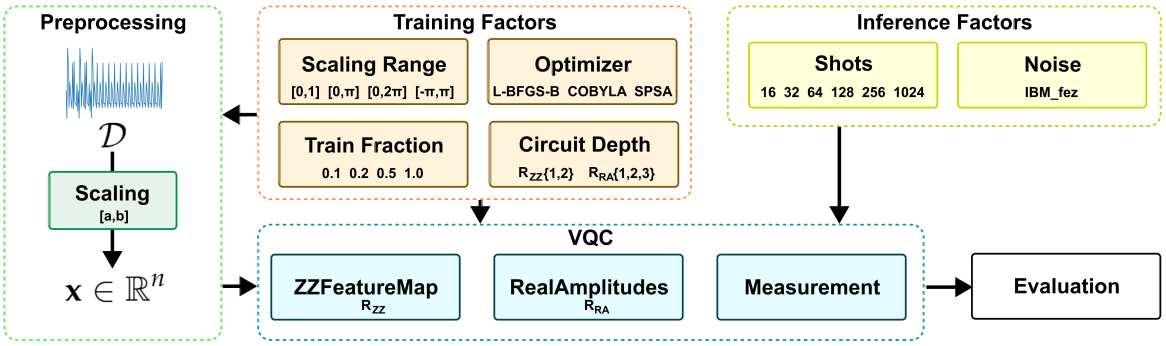

Figure 1. Benchmark workflow and experimental sweeps for VQC configuration and inference robustness.

Formally, a labeled dataset is considered,

$$\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}, \quad \mathbf{x}_i \in \mathbb{R}^d, \quad y_i \in \{0,1\} \tag{1}$$

where N is the total number of samples and d is the feature dimension. y_i denotes a binary label indicating whether the sample is dummy or a real trace. An $\mathbf{x}_i$ is a low-dimensional feature vector derived from power measurements. In this paper, the feature dimension is fixed to $d = 2$, corresponding to the trace position and power value, and the number of qubits is set to match d, enabling a minimal yet interpretable VQC configuration analysis under NISQ-era constraints.

3.2. Preprocessing

Angle-parameterized encodings map normalized classical values to rotation angles or phase parameters in quantum gates. Consequently, the normalization range can significantly affect circuit behavior, gradient magnitude, and optimization dynamics.

Let x denote a raw feature value for one dimension, and let $x_{\min}$ and $x_{\max}$ be the minimum and maximum values estimated from the training set for that dimension. To map x into a target range $[a, b]$, min–max scaling is applied:

$$\tilde{x} \;=\; a + \frac{x - x_{\min}}{x_{\max} - x_{\min}}(b - a) \tag{2}$$

Four scaling ranges are evaluated:

$$[a,b] \in \Big\{[0,1],\ [0,\pi],\ [0,2\pi],\ [-\pi,\pi]\Big\} \tag{3}$$

The $[0, 2\pi]$ range is commonly used for rotation-angle encodings, while $[-\pi, \pi]$ enables a symmetric angular range around zero. The $[0, 1]$ and $[0, \pi]$ ranges are included to assess whether restricting angles to a small interval improves trainability at the possible cost of reduced expressivity.

 doi:10.20944/preprints202603.0185.v1

3.3. Variational Quantum Classifier

3.3.1. Quantum Computing Preliminaries

A quantum circuit on n qubits acts on a complex state vector initialized to $|0\rangle^{\otimes n}$. A parameterized circuit produces a state:

$$|\psi(\mathbf{x}, \boldsymbol{\theta})\rangle = U(\mathbf{x}, \boldsymbol{\theta})\, |0\rangle^{\otimes n} \tag{4}$$

where $U(\mathbf{x}, \boldsymbol{\theta})$ is a composition of gates determined by the input $\mathbf{x}$ and trainable parameters $\boldsymbol{\theta}$. The final state is measured in the computational basis, producing a bitstring $z \in \{0,1\}^n$. The probability of observing z is

$$p_{\boldsymbol{\theta}}(z \mid \mathbf{x}) = |\langle z \rangle\, \psi(\mathbf{x}, \boldsymbol{\theta})|^2 \tag{5}$$

In practice, especially on NISQ hardware, probabilities are not directly accessible and are instead estimated from repeated circuit executions, called shots.

3.3.2. Circuit Structure

A canonical VQC structure is adopted, consisting of a data-encoding feature map and a trainable variational ansatz:

$$U(\mathbf{x}, \boldsymbol{\theta}) = U_{\mathrm{RA}}(\boldsymbol{\theta}; R_{\mathrm{RA}})\, U_{\mathrm{ZZ}}(\mathbf{x}; R_{\mathrm{ZZ}}) \tag{6}$$

where U_{ZZ} denotes the ZZFeatureMap with repetition number R_{ZZ}, and U_{RA} denotes the RealAmplitudes ansatz with repetition number R_{RA}. These repetition numbers control the encoding depth and variational expressivity, respectively.

ZZFeatureMap

Qiskit's second-order Pauli-Z evolution feature map (ZZFeatureMap) is used with full entanglement and repetition depth R_{ZZ}. For an input $\mathbf{x} \in \mathbb{R}^d$, a single repetition is implemented as a Hadamard layer followed by phase-encoding and pairwise ZZ interactions realized through the standard CX–P–CX pattern:

$$U_{\mathrm{ZZ}}^{(1)}(\mathbf{x}) = \left(\prod_{i<j} \mathrm{CX}_{i,j}\, P_j(\lambda_{ij}(\mathbf{x}))\mathrm{CX}_{i,j} \right) \left(\prod_{k=1}^{n} P_k(\lambda_k(\mathbf{x})) \right) H^{\otimes n} \tag{7}$$

where $P(\cdot)$ denotes a single-qubit phase gate and $\lambda_k(\mathbf{x})$ and $\lambda_{ij}(\mathbf{x})$ are the phase angles defined by the Qiskit feature-map template. The full feature map is $U_{\mathrm{ZZ}}(\mathbf{x}; R_{\mathrm{ZZ}}) = \left(U_{\mathrm{ZZ}}^{(1)}(\mathbf{x})\right)^{R_{\mathrm{ZZ}}}$.

RealAmplitudes

Qiskit's RealAmplitudes ansatz is used with full entanglement and repetition depth R_{RA}. The circuit alternates parameterized single-qubit R_y rotation layers with entangling layers composed of CX gates:

$$U_{\mathrm{RA}}(\boldsymbol{\theta}; R_{\mathrm{RA}}) = U_{\mathrm{rot}}(\boldsymbol{\theta}_0) \prod_{r=1}^{R_{\mathrm{RA}}} (U_{\mathrm{ent}}\, U_{\mathrm{rot}}(\boldsymbol{\theta}_r)) \tag{8}$$

where $U_{\mathrm{rot}}(\boldsymbol{\theta}_r) = \prod_{k=1}^{n} R_y(\theta_{r,k})$ and $U_{\mathrm{ent}} = \prod_{i<j} \mathrm{CX}_{i,j}$ under full entanglement. The parameter vector $\boldsymbol{\theta}$ contains all trainable angles across the repeated layers, and R_{RA} controls the variational depth.

3.3.3. Output Rule for Binary Classification

As described in the preliminaries, circuit execution followed by measurement induces a probability distribution over computational basis states. In the VQC implementation used in this work, the

 doi:10.20944/preprints202603.0185.v1

sampled measurement statistics are internally mapped to class probabilities through the built-in interpretation mechanism.

Given an input $\mathbf{x}$, the classifier produces estimated class probabilities $p_{\boldsymbol{\theta}}(y \mid \mathbf{x})$, from which the predicted label is obtained as:

$$\hat{y}(\mathbf{x}) = \arg\max_{c \in \{0,1\}} p_{\boldsymbol{\theta}}(y = c \mid \mathbf{x}) \tag{9}$$

where $\mathcal{I} : \{0,1\}^n \to \{0,1\}$ denotes the interpretation map; then $p_{\boldsymbol{\theta}}(y = c \mid \mathbf{x}) = \sum_{z:\mathcal{I}(z)=c} p_{\boldsymbol{\theta}}(z \mid \mathbf{x})$.

3.4. Training Objective and Optimizers

Training adjusts $\boldsymbol{\theta}$ to minimize an empirical risk on the training set. Using class probabilities, the standard binary cross-entropy loss $\mathcal{L}$ is adopted. Optimization follows the hybrid quantum–classical loop: a classical optimizer proposes $\boldsymbol{\theta}$, and the quantum device or simulator evaluates $\mathcal{L}(\boldsymbol{\theta})$ via repeated circuit executions. This study compares three representative optimizers that capture common trade-offs in QML, especially under expensive function evaluations and measurement noise.

L-BFGS-B

A limited-memory quasi-Newton method that uses gradient information and enforces simple box constraints. It typically converges in relatively few iterations on smooth objectives and is attractive when gradients are available. However, its performance can degrade when the objective is noisy because quasi-Newton curvature updates rely on consistent gradient and step information.

COBYLA

A gradient-free constrained optimizer that builds local linear approximations of the objective and constraints. It is commonly used in VQAs when analytic gradients are unavailable or unreliable. COBYLA can handle constraints naturally, but it may require many function evaluations and can scale unfavorably with the number of parameters.

SPSA

A stochastic approximation method that estimates a gradient using random simultaneous perturbations. Crucially, it requires only two objective evaluations per iteration regardless of parameter dimension, making it well-suited to QML settings where each evaluation entails many shots and is affected by noise. SPSA is generally robust to measurement noise, although it may exhibit larger iteration-to-iteration variance and depends on step-size hyperparameters.

3.5. Finite-Shot and Noisy Execution Conditions

Variational quantum circuits are evaluated through repeated measurements, and the resulting class probabilities are estimated from a finite number of shots. Let $\{z^{(s)}\}_{s=1}^{S}$ denote the measured bitstrings obtained from S independent circuit executions for a fixed input $\mathbf{x}$. The empirical distribution over outcomes is estimated as:

$$\widehat{p}_{\boldsymbol{\theta}}(z \mid \mathbf{x}) = \frac{1}{S} \sum_{s=1}^{S} \mathbb{I}\left[z^{(s)} = z\right] \tag{10}$$

where $z \in \{0,1\}^n$ is a computational-basis bitstring and $\mathbb{I}[\cdot]$ denote the indicator function. Given S independent shots, $z^{(s)}$ denotes the measured outcome at shot s.

This sampling-based estimate replaces the ideal probability distribution available in statevector simulations. As the shot budget S decreases, the variance of the empirical estimator increases, introducing stochastic fluctuations into both the predicted class probabilities and the optimization trajectory. Consequently, the shot count functions as a resource parameter that directly affects inference robustness.

 doi:10.20944/preprints202603.0185.v1

In addition to sampling uncertainty, hardware noise further perturbs circuit behavior under realistic NISQ conditions. Let the noiseless output state be expressed as:

$$\rho(\mathbf{x}, \boldsymbol{\theta}) = U(\mathbf{x}, \boldsymbol{\theta})\, \rho_0\, U(\mathbf{x}, \boldsymbol{\theta})^\dagger \tag{11}$$

To model noisy execution, we switch from the statevector notation to the density-operator formalism. For the noiseless case, $\rho(\mathbf{x}, \boldsymbol{\theta}) = |\psi(\mathbf{x}, \boldsymbol{\theta})\rangle\langle\psi(\mathbf{x}, \boldsymbol{\theta})|$, with $\rho_0 = (|0\rangle\langle 0|)^{\otimes n}$ and $(\cdot)^\dagger$ denoting the Hermitian adjoint.

Under noisy execution, the effective state becomes

$$\rho_{\text{noisy}}(\mathbf{x}, \boldsymbol{\theta}) = \mathcal{E}(\rho(\mathbf{x}, \boldsymbol{\theta})) \tag{12}$$

where $\mathcal{E}$ represents a quantum noise channel modeling gate errors and readout imperfections. Measurement is then performed on ρ_{noisy}, yielding a distorted outcome distribution.

Finite-shot estimation and hardware noise jointly influence the effective decision boundary learned by the classifier. Sampling noise introduces statistical variability, whereas hardware noise shifts the underlying measurement distribution. The experimental design, therefore, evaluates VQC configurations across multiple shot budgets and noise conditions to characterize sensitivity, stability, and robustness under deployment-relevant constraints. In this study, shot and noise effects are evaluated via inference-time sweeps with fixed trained parameters; shot/noise-aware training is left for future work.

3.6. Experimental Factors and Study Design

Table 1 summarizes the experimental factors. Each factor is varied while holding the others fixed, except where noted (e.g., depth sweeps are performed before optimizer comparisons when selecting a stable depth regime).

Table 1. Experimental factors and options considered in the benchmark.

Factor	Symbol / Option	Role in this study
Scaling range	$[a, b]$	Min–max scaling target range: $[0, 1]$, $[0, \pi]$, $[0, 2\pi]$, $[-\pi, \pi]$.
Feature map depth	R_{ZZ}	Repetitions in ZZFeatureMap; controls encoding depth and correlation structure.
Ansatz depth	R_{RA}	Repetitions in RealAmplitudes; controls expressivity vs. trainability.
Optimizer	L-BFGS-B / COBYLA / SPSA	Compares optimizer dynamics under the same circuit family and scaling.
Measurement budget	S shots	Evaluates stability/performance under finite-shot estimation.
Noise condition	$\mathcal{E}$	Evaluates robustness under noisy execution.

3.7. Evaluation Metrics

Standard binary classification metrics are reported on the held-out test set. Let TP, TN, FP, and FN denote the entries of the confusion matrix. The metrics are defined as follows:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{13}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{14}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{15}$$

 doi:10.20944/preprints202603.0185.v1

$$\mathrm{F1} = \frac{2 \cdot \mathrm{Precision} \cdot \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}} \tag{16}$$

4. Experiments

4.1. Experimental Setup

Experiments are conducted on a publicly released side-channel dataset that contains labeled instances corresponding to real operations and inserted dummy operations. The source dataset is organized into multiple execution contexts, including RSA, DES, and Hash, and is intended to support the evaluation of hiding techniques under learning-based distinguishability analyses [33]. The dataset is generated in a controlled environment and distributed publicly for reproducible evaluation. Due to the substantial size and the runtime costs, the majority of systematic sensitivity analyses are reported for the RSA context.

The implementation is based on Qiskit and Qiskit Machine Learning. Unless stated otherwise, results in this section correspond to analytic/statevector execution. Additional experiments vary shot budgets and noisy execution conditions; the corresponding results are reported separately. Table 2 summarizes the software and hardware environment used for the experiments.

Table 2. Software and hardware environment.

Component	Specification
Qiskit	2.2.3
Qiskit Machine Learning	0.9.0
Python	3.12.12
CPU	Intel Core i9-13900KF
Memory	128GB DDR4 RAM
OS	Ubuntu 24.04.2 LTS in WSL2

4.2. Circuit Depth

A grid sweep over circuit depth is conducted by varying the feature map repetitions ($R_{ZZ} \in \{1,2\}$) and ansatz repetitions ($R_{RA} \in \{1,2,3\}$) using L-BFGS-B as the optimizer. Table 3 reports test metrics and execution diagnostics for RSA under analytic/statevector execution and $[0, 2\pi]$ scaling.

Table 3. Classification performance versus circuit depth on the RSA dataset.

R_{ZZ}	R_{RA}	Accuracy	Precision	Recall	F1-score	Iterations	Time (s)
1	1	0.570	0.566	0.602	0.583	9	1514
1	2	**0.893**	**0.886**	**0.901**	**0.894**	12	2999
1	3	0.889	0.888	0.890	0.889	15	5284
2	1	0.657	0.793	0.426	0.554	14	2521
2	2	0.741	0.879	0.560	0.684	18	5018
2	3	0.742	0.875	0.564	0.686	16	6080

The highest performance in this sweep is obtained at $(R_{ZZ}, R_{RA}) = (1,2)$ with an accuracy of 0.893 and an F1-score of 0.894. Increasing the ansatz depth to $R_{RA} = 3$ at the same feature-map depth yields a slightly lower F1-score (0.889). Configurations with $R_{RA} = 1$ show substantially lower F1-scores (0.583 for $(1,1)$ and 0.554 for $(2,1)$), indicating that the shallow ansatz is insufficient for stable discrimination in this setting. Increasing the feature map depth to $R_{ZZ} = 2$ reduces F1 in all cases in this sweep; the best configuration remains below 0.70 F1.

Figure 2 summarizes mean F1-score trends across RSA, Hash, and DES. RSA and Hash exhibit a consistent qualitative pattern: performance improves markedly when moving from $R_{RA} = 1$ to $R_{RA} \in \{2,3\}$ under $R_{ZZ} = 1$, while increasing the feature-map repetitions to $R_{ZZ} = 2$ degrades F1 across the grid. For Hash, the best observed setting is $(R_{ZZ}, R_{RA}) = (1,3)$ with an F1-score of 0.808,

 doi:10.20944/preprints202603.0185.v1

closely followed by $(1,2)$ at 0.801. For DES, results are reported under a reduced training budget (train fraction = 0.2) due to runtime constraints; the best observed setting under this budget is $(1,2)$ with an F1-score of 0.696.

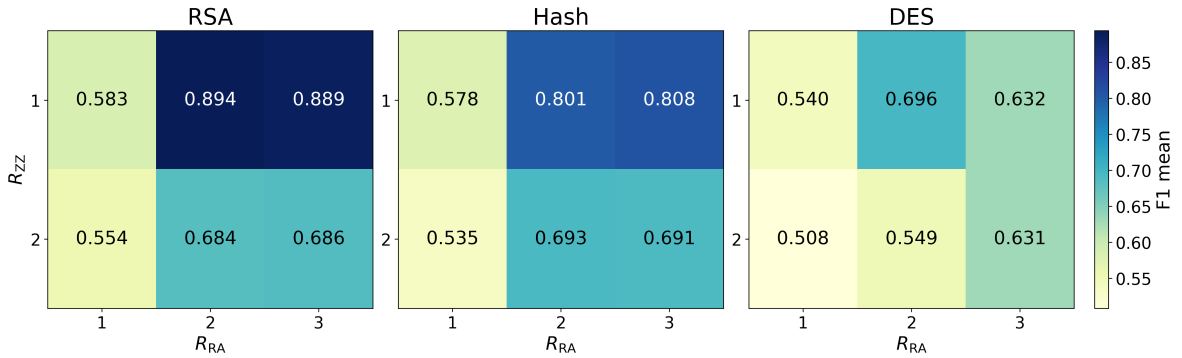

Figure 2. Mean F1-score over the (R_{ZZ}, R_{RA}) depth grid across RSA, Hash, and DES.

Runtime generally increases with deeper circuits, particularly when increasing the ansatz repetitions. For example, $(1,2)$ requires 2999 seconds, while $(1,3)$ increases to 5284 seconds and $(2,3)$ to 6080 seconds in the recorded environment. Although runtime is not strictly monotonic over all configurations, deeper settings such as $R_{RA} = 3$ and $R_{ZZ} = 2$ consistently incur higher cost than the selected reference setting $(1,2)$.

4.3. Optimizer Sensitivity

Based on the depth sweep, the configuration $(R_{ZZ}, R_{RA}) = (1,2)$ is selected as the reference depth for optimizer comparisons. Table 4 reports metrics aggregated across multiple random seeds.

Table 4. Optimizer comparison at $(R_{ZZ}, R_{RA}) = (1,2)$ (mean $\pm$ std over seeds).

Optimizer	Accuracy	F1-score	Iterations	Time (s)
L-BFGS-B	0.893 ± 0.001	0.894 ± 0.001	11.7	2999
COBYLA	0.887 ± 0.004	0.887 ± 0.005	75.0	1233
SPSA	0.883 ± 0.001	0.884 ± 0.001	300.0	15496

L-BFGS-B achieves the highest mean F1-score (0.894). In terms of optimizer dynamics, the average iteration counts differ substantially: L-BFGS-B converges in approximately 12 iterations, COBYLA performs 75 iterations, and SPSA reaches the maximum iteration budget (300).

Although L-BFGS-B converges in far fewer iterations than COBYLA, wall-clock time does not scale linearly with the number of iterations. In VQC training, the dominant cost is the number of circuit evaluations required by the optimizer to estimate the objective value at each iteration. Quasi-Newton methods such as L-BFGS-B may internally perform additional objective evaluations, making each iteration comparatively expensive. By contrast, COBYLA is a derivative-free trust-region method that often updates parameters using a lighter-weight sequence of function evaluations per step, which can reduce per-iteration cost even when the total iteration count is larger. As a result, COBYLA can achieve a shorter wall-clock time despite requiring more iterations, while SPSA incurs the largest runtime due to consuming the full iteration budget and repeatedly querying the circuit throughout the stochastic approximation procedure.

Figure 3 visualizes representative optimization trajectories. L-BFGS-B exhibits a steep early decrease in objective value and stabilizes within a few iterations, consistent with fast convergence under analytic/statevector execution. COBYLA also reduces the objective rapidly at the beginning, but shows larger transient fluctuations before stabilizing at a comparable final objective level. SPSA shows an initial drop to a similar objective value within the first few iterations, followed by a long plateau with only small variations. This behavior is consistent with SPSA quickly reaching a neighborhood of a

 doi:10.20944/preprints202603.0185.v1

stationary region under the chosen hyperparameters, after which the stochastic gradient approximation yields limited further improvement in the deterministic setting.

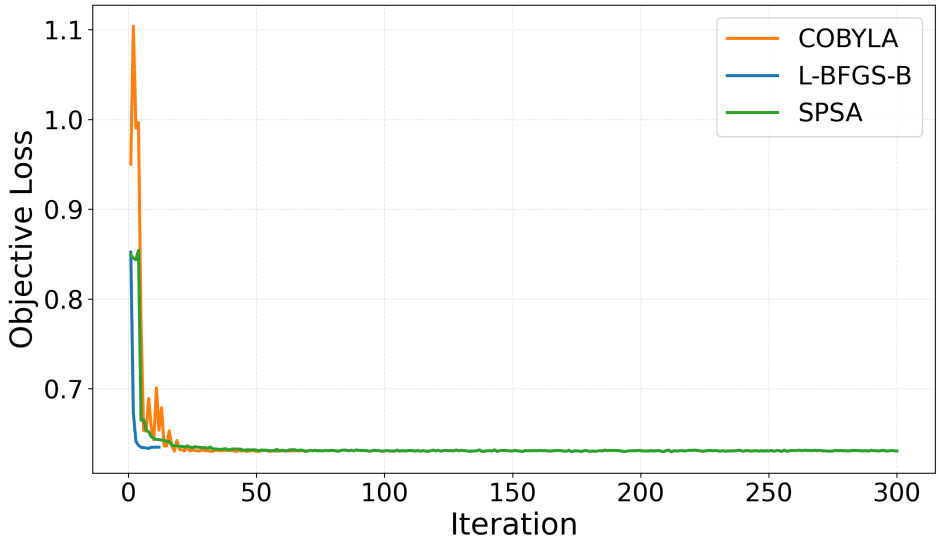

Figure 3. Representative training loss trajectories for optimizers at $R_{ZZ}, R_{RA} = (1, 2)$.

4.4. Training-Data Fraction Sensitivity

To assess robustness under reduced training data, the training set fraction is varied while keeping the circuit depth fixed at $(1, 2)$ and using the same set of optimizers. Table 5 reports results for training fractions 0.5, 0.2, and 0.1.

Table 5. Effect of training-data fraction.

Optimizer	Train fraction	Accuracy	F1-score	Iterations	Time (s)
L-BFGS-B	0.5	0.892 ± 0.003	0.893 ± 0.004	13.0	1645
	0.2	0.894 ± 0.001	0.895 ± 0.001	13.3	667
	0.1	0.702 ± 0.001	0.698 ± 0.001	14.0	352
COBYLA	0.5	0.888 ± 0.007	0.888 ± 0.008	76.3	650
	0.2	0.888 ± 0.004	0.888 ± 0.005	73.0	249
	0.1	0.669 ± 0.047	0.667 ± 0.048	74.0	130
SPSA	0.5	0.887 ± 0.001	0.886 ± 0.001	300.0	7991
	0.2	0.889 ± 0.001	0.889 ± 0.001	300.0	3129
	0.1	0.703 ± 0.001	0.699 ± 0.001	300.0	1574

A notable difference across optimizers emerges in variability under the low-data regime. At 0.1 training fraction, COBYLA exhibits substantially larger run-to-run variation (F1-score 0.667 ± 0.048) than L-BFGS-B (0.698 ± 0.001) and SPSA (0.699 ± 0.001). This suggests that, when training data are scarce, optimizer dynamics can materially affect stability even if mean performance is similar.

Training diagnostics further clarify the cost–stability trade-off. L-BFGS-B converges within a small number of iterations (approximately 13–14) across all fractions, while COBYLA performs roughly 73–76 iterations consistently, and SPSA reaches the maximum iteration budget (300) for all fractions. Runtime decreases as the training fraction is reduced for all optimizers, but the relative ordering is preserved: SPSA remains the most expensive option (e.g., 7991 s at 0.5) despite comparable predictive performance to COBYLA and L-BFGS-B in the moderate-data regimes.

4.5. Scaling-Range Sensitivity

To characterize sensitivity to min–max scaling ranges, experiments compare multiple target ranges, including $[0, 1]$, $[0, \pi]$, $[0, 2\pi]$, and $[-\pi, \pi]$. Since angle-based encodings directly map normalized

 doi:10.20944/preprints202603.0185.v1

values to gate parameters, the scaling range determines the effective parameter regime explored during optimization. Table 6 reports the resulting performance and training diagnostics under a fixed VQC configuration.

Table 6. Scaling-range sensitivity at the reference VQC configuration.

Scaling range	Accuracy	Precision	Recall	F1-score	Iterations	Time (s)
$[0, 1]$	0.558	0.545	0.700	0.613	16	4354
$[0, \pi]$	0.727	0.682	0.849	0.756	13	3552
$[0, 2\pi]$	0.893	0.886	0.901	0.894	12	2999
$[-\pi, \pi]$	0.891	0.888	0.894	0.891	17	4606

As shown in Table 6, the scaling range has a first-order impact on classification performance. The narrow range $[0, 1]$ yields the lowest performance (accuracy 0.558, F1-score 0.613), with a precision–recall imbalance (precision 0.545 vs. recall 0.700), indicating that restricting angles to a small interval can limit the effective separability learned by the circuit in this setting. Expanding the range to $[0, \pi]$ substantially improves performance (accuracy 0.727, F1-score 0.756), suggesting that a wider angular domain enables more discriminative decision boundaries while remaining trainable.

The strongest performance is obtained for the full-period range $[0, 2\pi]$ (accuracy 0.893, F1-score 0.894), with a balanced precision–recall profile (precision 0.886, recall 0.901). The centered range $[-\pi, \pi]$ achieves a similarly high accuracy (0.891) but a slightly lower F1-score (0.891), with marginally higher precision (0.888) and lower recall (0.894) compared to the $[0, 2\pi]$ setting. Notably, the optimization diagnostics differ across scaling choices: $[0, 2\pi]$ converges in fewer iterations (12) and lower runtime (2999 s) than $[-\pi, \pi]$ (17 iterations, 4606 s), while $[0, \pi]$ achieves moderate performance with relatively low runtime (3552 s).

These results indicate that scaling range selection is not a cosmetic preprocessing choice but a core configuration parameter for VQC training. In the evaluated setting, ranges that cover a full 2π span yield the best performance and the most efficient convergence, whereas overly narrow ranges can markedly degrade accuracy and F1-score.

4.6. Finite-Shot and Noisy Execution Results

Finite-shot sampling and noisy execution conditions are evaluated to approximate deployment beyond idealized simulation. In this setting, the trained model is held fixed, and performance is measured under repeated inference runs with different sampling seeds.

4.6.1. Finite-Shot Sensitivity

Table 7 reports inference performance as a function of the shot budget under noiseless conditions. An analytic baseline (Ideal) is included for reference. As the shot budget increases, both accuracy and F1-score improve, and the variability across repeated runs decreases. At very low shot budgets (e.g., 16 and 32), the min–max range indicates large run-to-run fluctuations, whereas the dispersion becomes small at 512 and 1024 shots. At 1024 shots, the mean F1-score (0.892) approaches the analytic baseline (0.894), indicating that sampling-induced variability becomes negligible at this budget for the studied configuration.

 doi:10.20944/preprints202603.0185.v1

Table 7. Inference robustness under finite shots.

Shots	Accuracy	F1-score	Accuracy (min–max)	F1 (min–max)
Ideal	0.893 ± 0.001	0.894 ± 0.001	0.893–0.893	0.894–0.894
16	0.839 ± 0.033	0.835 ± 0.037	0.768–0.878	0.752–0.876
32	0.855 ± 0.024	0.854 ± 0.024	0.805–0.877	0.803–0.875
64	0.867 ± 0.018	0.867 ± 0.018	0.835–0.885	0.833–0.885
128	0.875 ± 0.011	0.876 ± 0.011	0.855–0.894	0.855–0.895
256	0.886 ± 0.007	0.887 ± 0.008	0.874–0.895	0.873–0.896
512	0.887 ± 0.006	0.888 ± 0.006	0.878–0.896	0.880–0.898
1024	0.891 ± 0.004	0.892 ± 0.004	0.884–0.897	0.886–0.898

4.6.2. Noise Model from Real Device

To complement noiseless shot sweeps, a hardware-informed noise emulation is considered using a noise model derived from an IBM backend (IBM_fez). A noise-strength scaling experiment is conducted by multiplying the calibrated hardware error rates by a scale factor. Figure 4 summarizes the combined effect of shot budgets and noise scaling on classification performance.

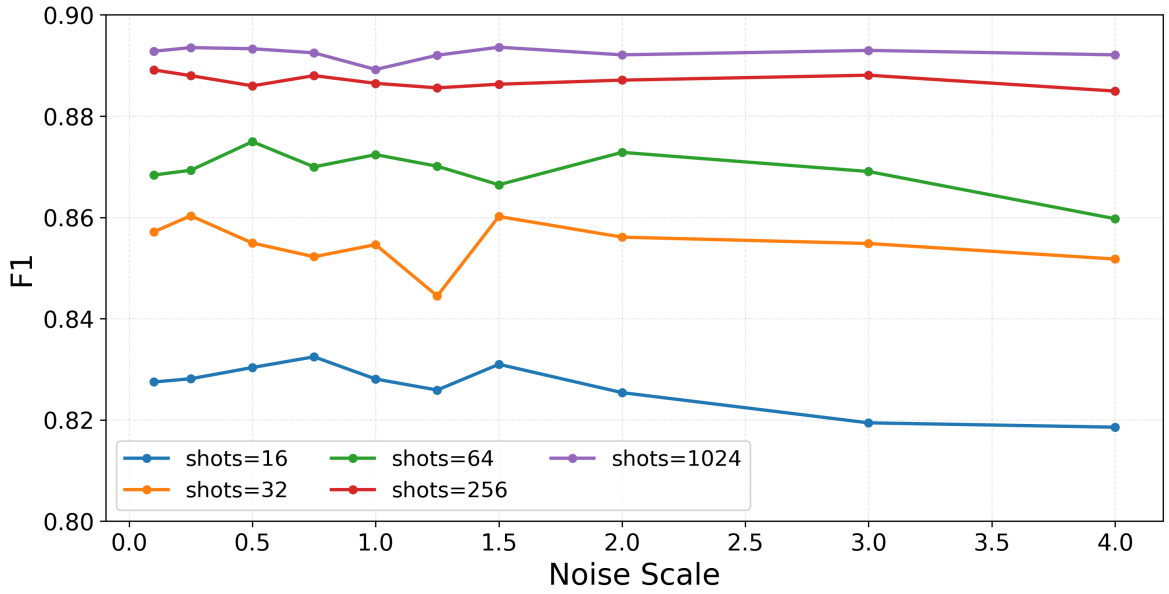

Figure 4. F1-score under combined shot budgets and IBM_fez-derived noise scaling.

Across the evaluated conditions, shot budget is the dominant factor affecting performance. Low-shot regimes (16 and 32 shots) exhibit noticeably lower F1-score compared to higher-shot regimes (256 and 1024 shots), regardless of noise scaling. In contrast, increasing the noise scale produces only gradual changes in performance, while the ordering across shot budgets remains consistent. These results indicate that sampling uncertainty is the primary limiting factor in low-shot regimes, whereas moderate increases in hardware-level noise do not induce abrupt performance degradation.

Table 8 provides detailed numerical results at 1024 shots, together with the corresponding physical error-rate parameters of the IBM_fez noise model. Here, e_{1q}, e_{2q}, and e_{ro} denote the average error rates of single-qubit gates, two-qubit entangling gates, and measurement readout, respectively. These parameters are uniformly scaled according to the applied noise factor to emulate progressively stronger noise conditions relative to the calibrated hardware baseline.

 doi:10.20944/preprints202603.0185.v1

Table 8. Noise-scale sweep at 1024 shots using an IBM_fez-derived noise model.

Scale	Accuracy	F1-score	e_{1q}	e_{2q}	e_{ro}
0.10	0.892	0.893	0.000031	0.000263	0.001447
0.25	0.892	0.894	0.000078	0.000658	0.003616
0.50	0.892	0.893	0.000155	0.001315	0.007233
0.75	0.892	0.893	0.000233	0.001973	0.010849
1.00	0.888	0.889	0.000310	0.002631	0.014465
1.25	0.891	0.892	0.000388	0.003289	0.018082
1.50	0.892	0.894	0.000466	0.003946	0.021698
2.00	0.891	0.892	0.000621	0.005262	0.028931
3.00	0.892	0.893	0.000931	0.007893	0.043396
4.00	0.891	0.892	0.001242	0.010523	0.057861

At 1024 shots, performance remains stable across the tested noise scaling range, despite substantial increases in gate and readout error rates. This observation further confirms that, under sufficient sampling budgets, the evaluated VQC configuration maintains robust classification performance under realistic hardware noise conditions.

5. Discussion

5.1. Depth-Dependent Regimes and Performance Trends

The depth sweep highlights that circuit depth is a first-order factor in VQC behavior, but improvements are not monotonic [4]. In the tested grid, the best performance is obtained at a moderate depth, $(R_{ZZ}, R_{RA}) = (1, 2)$, achieving an F1-score of 0.894. Increasing the ansatz depth to $R_{RA} = 3$ at the same feature-map depth yields a comparable but slightly lower F1-score (0.889) while the shallow ansatz setting $(1, 1)$ performs substantially worse (F1 0.583). These results suggest that a minimal ansatz may be insufficient to model the discriminative structure present in the power traces, whereas moderate expressivity is beneficial [15].

By contrast, increasing the feature-map depth from $R_{ZZ} = 1$ to $R_{ZZ} = 2$ reduces performance across all tested ansatz depths in the RSA sweep. This behavior is consistent with the broader understanding that deeper variational constructions can introduce optimization difficulty and sensitivity in NISQ-era regimes [4,14]. While the present experiments are conducted with a small number of qubits, the observed degradation indicates that additional data re-uploading and entangling structure in the feature map can still shift the optimization landscape in unfavorable ways, even at a small scale.

From a practical perspective, the depth sweep supports a conservative strategy for configuration selection: begin with a shallow feature map and tune ansatz depth within a small range. For the present workload, $R_{ZZ} = 1$ combined with $R_{RA} \in \{2, 3\}$ forms a stable high-performing region, whereas $R_{ZZ} = 2$ and $R_{RA} = 1$ are comparatively unreliable settings in this testbed.

5.2. Optimizer Choice

Optimizer selection materially changes both convergence behavior and computational cost, even under the same circuit structure. At the selected depth $(1, 2)$, L-BFGS-B attains the highest mean F1-score (0.894 ± 0.001), followed by COBYLA (0.887 ± 0.005) and SPSA (0.884 ± 0.001). Although the mean performance gap is moderate, training dynamics differ sharply. L-BFGS-B converges in roughly 12 iterations on average, COBYLA requires about 75 iterations, and SPSA reaches the maximum iteration budget (300) in the reported runs.

Runtime diagnostics indicate that iteration counts alone are insufficient to compare optimizers, as per-iteration costs can differ substantially. In the experiments, SPSA incurs a markedly larger wall-clock cost than the others, while COBYLA remains the fastest despite a larger iteration count, reflecting the practical importance of measuring cost in addition to final predictive metrics. These observations support treating optimizer choice as an integral part of the VQC configuration space rather than as a secondary implementation detail [3,4].

 doi:10.20944/preprints202603.0185.v1

A pragmatic takeaway is that L-BFGS-B is a strong default for small VQC instances when analytic execution is available and stable optimization is desired, whereas COBYLA can be attractive for rapid scanning due to its low wall-clock cost. SPSA remains relevant for scenarios where stochasticity is unavoidable, but the high iteration cost observed here motivates careful budgeting and early-stopping criteria when using SPSA.

5.3. Data-Efficiency Behavior Under Reduced Training Fractions

Training-data fraction sweeps show that performance is relatively stable in moderate-data regimes but degrades under aggressive data reduction [6,12]. At training fractions of 0.5 and 0.2, F1 remains near the full-data regime for all three optimizers, with limited run-to-run variability except for modest fluctuations under COBYLA. At a training fraction of 0.1, mean F1 decreases substantially to 0.667–0.699, and COBYLA exhibits noticeably larger variability.

This pattern suggests that, for the present representation and circuit family, the learned decision boundary depends on a nontrivial amount of labeled data to remain stable. For experimental design, this supports explicitly reporting training fractions and discouraging interpreting single-run results as representative in low-data settings. More broadly, this finding reinforces the importance of reporting mean and variance statistics when characterizing VQC behavior, as data scarcity can amplify stochasticity and increase sensitivity to initialization [4].

5.4. Implications of Scaling Ranges and NISQ Constraints

Scaling-range sensitivity is a first-order factor because angle-based encodings map normalized values directly to gate parameters [2,13]. In our experiments, $[0, 2\pi]$ yields the strongest performance (F1 = 0.894), while overly narrow ranges such as $[0, 1]$ markedly degrade separability (F1 = 0.613). Intermediate ranges $[0, \pi]$ show intermediate behavior, and the centered range $[-\pi, \pi]$ achieves similar accuracy but slightly lower F1 than $[0, 2\pi]$. These results indicate that scaling range is not a cosmetic preprocessing choice but a core configuration parameter that should be explicitly documented.

Regarding deployment-relevant constraints, inference-time sweeps with fixed trained parameters show that shot budget is the dominant driver of performance variability: low-shot regimes (e.g., 16–32 shots) produce substantial run-to-run fluctuations, while budgets of 512–1024 shots approach the analytic baseline and yield stable predictions. Under sufficient shot budgets, an IBM_fez-derived noise model with scaled error rates induces only gradual performance changes, suggesting that sampling uncertainty can be a more immediate bottleneck than moderate noise increases for the studied configuration [3].

5.5. Limitations and Avenues for Future Work

Several limitations should be considered when interpreting the findings. First, the input representation is intentionally minimal to isolate VQC configuration effects, but richer time-series representations may reveal additional structure and different sensitivity regimes. Second, the primary results are derived from a simulated quantum computing environment and analytic execution, and the extent to which trends transfer to physical measurement conditions depends on noise characteristics and sampling constraints. Third, only a single circuit family and a limited depth grid are explored; alternative encodings, ansatz families, initialization schemes, and regularization strategies could yield different stability trade-offs.

Future work can address these limitations by extending inputs to windowed or higher-dimensional representations while managing qubit constraints via dimensionality reduction or structured encodings, expanding circuit families and studying entanglement patterns and re-uploading strategies, performing shot/noise-aware training and evaluating error mitigation techniques, and running selected configurations on real quantum backends to validate sensitivity trends under hardware calibration dynamics. These directions would further strengthen practical guidance for applying variational models to security-relevant side-channel data.

 doi:10.20944/preprints202603.0185.v1

6. Conclusions

This paper presents a controlled experimental study of VQCs on a security-relevant side-channel workload, with two objectives: to assess whether VQCs can learn discriminative patterns from power-based side-channel inputs to distinguish real from dummy operations, and to characterize how key VQC design choices and NISQ-era execution constraints affect performance and robustness. Using a small number of qubits input representation and a canonical circuit family, systematic experiments are conducted across circuit depth, optimizers, and training data fractions.

The experiments identify clear configuration-dependent regimes. Performance improves substantially when increasing the ansatz depth from $R_{\mathrm{RA}} = 1$ to $R_{\mathrm{RA}} = 2$, whereas further increasing to $R_{\mathrm{RA}} = 3$ yields no additional benefit in our sweep while incurring higher runtime. In contrast, increasing the feature-map repetitions consistently degrades performance, indicating that deeper data encoding is not necessarily beneficial even in small-qubit settings. Optimizer selection significantly affects both convergence behavior and runtime cost. Training-data fraction sweeps show that performance remains stable at moderate fractions but declines substantially at aggressive reductions, reinforcing the need for variability-aware reporting.

Overall, the study provides empirically grounded guidance for configuring VQCs on side-channel inputs: depth and optimizer choices should be treated as primary knobs, and deployment-relevant constraints such as scaling ranges, finite shots, and noise should be explicitly documented and evaluated. These findings help clarify the practical behavior and limitations of QML models in a security-relevant dataset setting and support more reproducible, configuration-aware experimentation at the intersection of QML and SCA.

Author Contributions: Conceptualization, S.P. and Y.S.; methodology, S.P. and Y.S.; software, S.P.; validation, S.P.; formal analysis, S.P. and Y.S.; investigation, S.P.; resources, S.P.; data curation, S.P.; writing—original draft preparation, S.P.; writing—review and editing, S.P. and Y.S.; visualization, S.P.; supervision, Y.S.; project administration, Y.S.; funding acquisition, Y.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the ITRC(Information Technology Research Center) support program(IITP-2026-RS-2020-II201789) supervised by the IITP(Institute for Information & Communications Technology Planning & Evaluation). This work was supported by the IITP(Institute of Information & Communications Technology Planning & Evaluation)-ICAN(ICT Challenge and Advanced Network of HRD) grant funded by the Korea government(Ministry of Science and ICT)(IITP-2026-RS-2023-00260248). This work was supported by the Commercialization Promotion Agency for R&D Outcomes(COMPA) grant funded by the Korea government(Ministry of Science and ICT) (2710086167). This work was supported by the Commercialization Promotion Agency for R&D Outcomes(COMPA) grant funded by the Korea government(Ministry of Science and ICT) (RS-2025-02412990).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The SCADataset used in this study is publicly available at https://github.com/PLASS-Lab/SCADataset (accessed on 27 February 2026) [33].

Acknowledgments: A preliminary version of this work was presented at the 26th International Symposium on Advanced Intelligent Systems (ISIS2025), Cheongju, Korea, under the title "Quantum Neural Network Approach for Identifying Dummy Power Traces in Side-Channel Analysis". This article substantially extends the conference paper through additional experiments, expanded analysis, and significant methodological refinements.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Biamonte, J.; Wittek, P.; Pancotti, N.; Rebentrost, P.; Wiebe, N.; Lloyd, S. Quantum Machine Learning. *Nature* **2016**, *549*. https://doi.org/10.1038/nature23474.
2. Schuld, M.; Petruccione, F. *Supervised Learning with Quantum Computers*, 1st ed.; Springer Publishing Company, Incorporated, 2018. https://doi.org/10.1007/978-3-319-96424-9.

 doi:10.20944/preprints202603.0185.v1

3. Preskill, J. Quantum Computing in the NISQ era and beyond. *Quantum* **2018**, *2*, 79. https://doi.org/10.22331/q-2018-08-06-79.
4. Cerezo, M.; Arrasmith, A.; Babbush, R.; Benjamin, S.; Endo, S.; Fujii, K.; Mcclean, J.; Mitarai, K.; Yuan, X.; Cincio, L.; et al. Variational quantum algorithms. *Nature Reviews Physics* **2021**, *3*, 1–20. https://doi.org/10.1038/s42254-021-00348-9.
5. Kim, J.; Park, S.; Cha, J.; Son, E.; Son, Y. Novel Synthetic Dataset Generation Method with Privacy-Preserving for Intrusion Detection System. *Applied Sciences* **2025**, *15*. https://doi.org/10.3390/app151910609.
6. Picek, S.; Perin, G.; Mariot, L.; Wu, L.; Batina, L. SoK: Deep Learning-based Physical Side-channel Analysis. *ACM Comput. Surv.* **2023**, *55*. https://doi.org/10.1145/3569577.
7. Küçükkara, M.Y.; Atban, F.; Bayılmış, C. Quantum-Neural Network Model for Platform Independent Ddos Attack Classification in Cyber Security. *Advanced Quantum Technologies* **2024**, *7*, 2400084. https://doi.org/doi.org/10.1002/qute.202400084.
8. Kocher, P.; Jaffe, J.; Jun, B. Differential Power Analysis. In Proceedings of the Advances in Cryptology — CRYPTO' 99; Wiener, M., Ed., Berlin, Heidelberg, 1999; pp. 388–397. https://doi.org/10.1007/3-540-48405-1_25.
9. Mangard, S.; Oswald, E.; Popp, T. *Power Analysis Attacks: Revealing the Secrets of Smart Cards*; Springer-Verlag: Berlin, Heidelberg, 2007. https://doi.org/10.1007/978-0-387-38162-6.
10. Quisquater, J.J.; Samyde, D. ElectroMagnetic Analysis (EMA): Measures and Counter-measures for Smart Cards. In Proceedings of the Smart Card Programming and Security; Attali, I.; Jensen, T., Eds., Berlin, Heidelberg, 2001; pp. 200–210. https://doi.org/10.1007/3-540-45418-7_17.
11. Lee, J.; Han, D.G. Security analysis on dummy based side-channel countermeasures—Case study: AES with dummy and shuffling. *Applied Soft Computing* **2020**, *93*, 106352. https://doi.org/doi.org/10.1016/j.asoc.2020.106352.
12. Masure, L.; Dumas, C.; Prouff, E. A Comprehensive Study of Deep Learning for Side-Channel Analysis. *IACR Transactions on Cryptographic Hardware and Embedded Systems* **2019**, *2020*, 348–375. https://doi.org/10.13154/tches.v2020.i1.348-375.
13. Havlíček, V.; Córcoles, A.; Temme, K.; Harrow, A.; Kandala, A.; Chow, J.; Gambetta, J. Supervised learning with quantum-enhanced feature spaces. *Nature* **2019**, *567*, 209–212. https://doi.org/10.1038/s41586-019-0980-2.
14. McClean, J.; Boixo, S.; Smelyanskiy, V.; Babbush, R.; Neven, H. Barren plateaus in quantum neural network training landscapes. *Nature Communications* **2018**, *9*. https://doi.org/10.1038/s41467-018-07090-4.
15. Schuld, M.; Sweke, R.; Meyer, J.J. Effect of data encoding on the expressive power of variational quantum-machine-learning models. *Phys. Rev. A* **2021**, *103*, 032430. https://doi.org/10.1103/PhysRevA.103.032430.
16. Park, S.; Kuai, J.; Kim, H.; Ko, H.; Jung, C.; Son, Y. A Lightweight Degradation-Aware Framework for Robust Object Detection in Adverse Weather. *Electronics* **2026**, *15*. https://doi.org/10.3390/electronics15010146.
17. Cagli, E.; Dumas, C.; Prouff, E. Convolutional Neural Networks with Data Augmentation Against Jitter-Based Countermeasures. In Proceedings of the Cryptographic Hardware and Embedded Systems – CHES 2017; Fischer, W.; Homma, N., Eds., Cham, 2017; pp. 45–68. https://doi.org/10.1007/978-3-319-66787-4_3.
18. Zaid, G.; Bossuet, L.; Habrard, A.; Venelli, A. Methodology for Efficient CNN Architectures in Profiling Attacks. *IACR Transactions on Cryptographic Hardware and Embedded Systems* **2019**, *2020*, 1—36. https://doi.org/10.13154/tches.v2020.i1.1-36.
19. Park, J.; Xu, X.; Jin, Y.; Forte, D.; Tehranipoor, M. Power-based Side-Channel Instruction-level Disassembler. In Proceedings of the 2018 55th ACM/ESDA/IEEE Design Automation Conference (DAC), 2018, pp. 1–6. https://doi.org/10.1109/DAC.2018.8465848.
20. Strobel, D.; Bache, F.; Oswald, D.; Schellenberg, F.; Paar, C. SCANDALee: A side-ChANnel-based DisAssembLer using local electromagnetic emanations. In Proceedings of the 2015 Design, Automation & Test in Europe Conference & Exhibition (DATE), 2015, pp. 139–144. https://doi.org/10.7873/DATE.2015.0639.
21. Eisenbarth, T.; Paar, C.; Weghenkel, B., Building a Side Channel Based Disassembler. In *Transactions on Computational Science X: Special Issue on Security in Computing, Part I*; Springer Berlin Heidelberg: Berlin, Heidelberg, 2010; pp. 78–99. https://doi.org/10.1007/978-3-642-17499-5_4.
22. Park, J.; Tyagi, A. Using Power Clues to Hack IoT Devices: The power side channel provides for instruction-level disassembly. *IEEE Consumer Electronics Magazine* **2017**, *6*, 92–102. https://doi.org/10.1109/MCE.2017.2684982.
23. Krishnankutty, D.; Li, Z.; Robucci, R.; Banerjee, N.; Patel, C. Instruction Sequence Identification and Disassembly Using Power Supply Side-Channel Analysis. *IEEE Transactions on Computers* **2020**, *69*, 1639–1653. https://doi.org/10.1109/TC.2020.3018092.

 doi:10.20944/preprints202603.0185.v1

24. Narimani, P.; Akhaee, M.A.; Habibi, S.A. Side-Channel based Disassembler for AVR Micro-Controllers using Convolutional Neural Networks. In Proceedings of the 2021 18th International ISC Conference on Information Security and Cryptology (ISCISC), 2021, pp. 75–80. https://doi.org/10.1109/ISCISC53448.2021.9720466.
25. Cristiani, V.; Lecomte, M.; Hiscock, T. A Bit-Level Approach to Side Channel Based Disassembling. In Proceedings of the Smart Card Research and Advanced Applications; Belaïd, S.; Güneysu, T., Eds., Cham, 2020; pp. 143–158. https://doi.org/10.1007/978-3-030-42068-0_9.
26. van Geest, J.; Buhan, I. A Side-Channel Based Disassembler for the ARM-Cortex M0. In Proceedings of the Applied Cryptography and Network Security Workshops; Zhou, J.; Adepu, S.; Alcaraz, C.; Batina, L.; Casalicchio, E.; Chattopadhyay, S.; Jin, C.; Lin, J.; Losiouk, E.; Majumdar, S.; et al., Eds., Cham, 2022; pp. 183–199. https://doi.org/10.1007/978-3-031-16815-4_11.
27. Fendri, H.; Macchetti, M.; Perrine, J.; Stojilović, M. A deep-learning approach to side-channel based CPU disassembly at design time. In Proceedings of the Proceedings of the 2022 Conference & Exhibition on Design, Automation & Test in Europe, Leuven, BEL, 2022; DATE '22, p. 670–675. https://doi.org/10.23919/DATE54114.2022.9774531.
28. Maillard, J.; Hiscock, T.; Lecomte, M.; Clavier, C. Side-channel disassembly on a System-on-Chip: A practical feasibility study. *Microprocessors and Microsystems* **2023**, *101*, 104904. https://doi.org/doi.org/10.1016/j.micpro.2023.104904.
29. Alabdulwahab, S.; Cheong, M.; Seo, A.; Kim, Y.T.; Son, Y. Enhancing deep learning-based side-channel analysis using feature engineering in a fully simulated IoT system. *Expert Systems with Applications* **2025**, *266*, 126079. https://doi.org/doi.org/10.1016/j.eswa.2024.126079.
30. Alabdulwahab, S.; Kim, J.; Kim, Y.T.; Son, Y. Advanced Side-Channel Evaluation Using Contextual Deep Learning-Based Leakage Modeling. *ACM Trans. Softw. Eng. Methodol.* **2026**, *35*. https://doi.org/10.1145/3734219.
31. Arguello, C.N.; Searle, H.; Rampazzi, S.; Butler, K.R.B. A Practical Methodology for ML-Based EM Side Channel Disassemblers, 2022, [arXiv:cs.CR/2206.10746].
32. Lee, J.; Han, D.G. DLDDO: Deep Learning to Detect Dummy Operations. In Proceedings of the Information Security Applications; You, I., Ed., Cham, 2020; pp. 73–85. https://doi.org/10.1007/978-3-030-65299-9_6.
33. Park, S.; Seo, A.; Cheong, M.; Kim, H.; Kim, J.; Son, Y. Evaluating the Vulnerability of Hiding Techniques in Cyber-Physical Systems Against Deep Learning-Based Side-Channel Attacks. *Applied Sciences* **2025**, *15*. https://doi.org/10.3390/app15136981.
34. Shor, P.W. Fault-tolerant quantum computation, 1997, [arXiv:quant-ph/quant-ph/9605011].
35. Larasati, H.T.; Choi, B.S. Towards fault-tolerant distributed quantum computation (FT-DQC): Taxonomy, recent progress, and challenges. *ICT Express* **2025**, *11*, 417–435. https://doi.org/doi.org/10.1016/j.icte.2025.03.007.
36. Preskill, J. Fault-tolerant quantum computation, 1997, [arXiv:quant-ph/quant-ph/9712048].
37. Shor, P.W. Scheme for reducing decoherence in quantum computer memory. *Phys. Rev. A* **1995**, *52*, R2493–R2496. https://doi.org/10.1103/PhysRevA.52.R2493.
38. Steane, A.M. Error Correcting Codes in Quantum Theory. *Phys. Rev. Lett.* **1996**, *77*, 793–797. https://doi.org/10.1103/PhysRevLett.77.793.
39. Kitaev, A. Fault-tolerant quantum computation by anyons. *Annals of Physics* **2003**, *303*, 2–30. https://doi.org/doi.org/10.1016/S0003-4916(02)00018-0.
40. Mitarai, K.; Negoro, M.; Kitagawa, M.; Fujii, K. Quantum circuit learning. *Phys. Rev. A* **2018**, *98*, 032309. https://doi.org/10.1103/PhysRevA.98.032309.
41. Bellante, A.; Fioravanti, T.; Carminati, M.; Zanero, S.; Luongo, A. Evaluating the potential of quantum machine learning in cybersecurity: A case-study on PCA-based intrusion detection systems. *Computers & Security* **2025**, *154*, 104341. https://doi.org/doi.org/10.1016/j.cose.2025.104341.
42. Park, J.; Rahman, F.; Vassilev, A.; Forte, D.; Tehranipoor, M. Leveraging Side-Channel Information for Disassembly and Security. *J. Emerg. Technol. Comput. Syst.* **2019**, *16*. https://doi.org/10.1145/3359621.