# Quantum Machine Learning versus Machine Learning for Intrusion Detection in 5G Environments

Ioannis D. Leonidas*†, Panagiotis Radoglou-Grammatikis†‡, Evangelos K. Markakis§, Michail-Alexandros Kourtis††, Thomas Lagkas¶, Georgios Th. Papadopoulos‖, Vasileios Argyriou** and Panagiotis Sarigiannidis‡

***Abstract*—Quantum machine learning (QML) is often motivated by the promise of richer feature representations and similarity measures, but its practical value for realistic, multiclass telecom-security data remains unclear under near-term constraints. This paper presents a unified benchmark of classical, quantum-inspired, and trainable QML approaches for multiclass intrusion detection on Packet Forwarding Control Protocol (PFCP) control-plane traffic in 5G Core networks. We evaluate classical baselines, including a Radial Basis Function Support Vector Machine (RBF-SVM) and Extreme Gradient Boosting (XGBoost), a quantum-inspired $k$-nearest neighbors classifier based on an analytic SWAP-test kernel, parametrized quantum classifiers with hardware-efficient and data-reuploading ansatzes, and trainable quantum kernels optimized via the quantum kernel alignment method. Across multiple aggregation windows of the PFCP APP Layer dataset, the quantum-inspired kernel method is the most consistently competitive approach. In particular, on the 240 s window, qkNN achieves 62.1% accuracy and 61.6% macro F1, outperforming XGBoost, which reaches 59.9% accuracy and 60.3% macro F1, and RBF-SVM, which reaches 55.1% accuracy and 51.7% macro F1. Fully trainable quantum models yield more modest performance and exhibit higher sensitivity to scaling and optimization. Overall, the results indicate that quantum-inspired kernel methods currently provide the most practical benefits for PFCP intrusion detection at realistic dataset sizes, whereas trainable quantum machine learning models require more robust optimization and improved resource scaling to deliver reliable improvements.**

***Index Terms*—5G Core Security, Intrusion Detection, Multiclass Classification, PFCP, Quantum Machine Learning, Quantum-Inspired Algorithms**

## I. Introduction

Machine Learning (ML) is widely used in network Intrusion Detection Systems (IDS) to classify traffic as benign or malicious, including multiclass settings that must distinguish between multiple attack types. In the context of 5G security and network monitoring, research spans clustering [1], decision trees [2], neural networks [3], [4], metaheuristics [5], [6], and ensembles [7], [8]. For multiclass intrusion detection on realistic datasets, strong out-of-the-box baselines include random forests, XGBoost variants, and neural architectures [9], [10], [11].

Despite these advances, control-plane protocols in modern telecom cores create challenging regimes for IDS, characterized by high-dimensional and heterogeneous statistics, multiclass attack taxonomies, and operational constraints that limit model size, latency, and retraining frequency. These constraints motivate exploring whether quantum machine learning (QML) and quantum-inspired similarity measures can improve representation and classification in settings where classical models may require extensive feature engineering or large amounts of labeled data.

The 5G Core (5GC) introduces a service-based architecture (SBA) that decouples network functions and enables flexible, cloud-native deployment of mobile core services. Key control-plane functions such as the Access and Mobility Management Function (AMF), Session Management Function (SMF), and Policy Control Function (PCF) interact through standardized interfaces to support dynamic session management, network slicing, and ultra-low latency services. While this architectural shift improves scalability and programmability, it also expands the attack surface of the core network by increasing signaling complexity and inter-function communication. As a result, securing control-plane protocols in the 5GC has become a critical challenge, particularly for mechanisms responsible for steering and maintaining user-plane traffic.

Packet Forwarding Control Protocol (PFCP) is a core control-plane protocol in 5G networks, coordinating session

* This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement No, 101225776 (SecQdevOps- Trustworthy Quantum DevOps for Secure Software and Hardware Quantum Engineering) - Disclaimer: Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or [name of the granting authority]. Neither the European Union nor the granting authority can be held responsible for them.

† I. D. Leonidas and P. Radoglou-Grammatikis are with K3Y Ltd, Studentski District, Vitosha Quarter, Bl. 9, 1700 Sofia, Bulgaria - E-Mail: ileonidas@k3y.bg, pradoglou@k3y.bg

§ E. K. Markakis is with the Department of Electrical and Computer Engineering, Hellenic Mediterranean University, P.O. Box 1939, GR-71004 Heraklion, Crete, Greece - E-Mail: emarkakis@hmu.gr

†† M. A. Kourtis is with the Institute of Informatics and Telecommunications, National Centre for Scientific Research "Demokritos", P.O. Box 60037, 15310 Aghia Paraskevi, Greece - E-Mail: akis.kourtis@iit.demokritos.gr

¶ T. Lagkas is with the Department of Informatics, Faculty of Science, Democritus University of Thrace, EX B14, Kavala Campus, Kavala, Greece (e-mail: tlagkas@cs.duth.gr).

¶ G. Th. Papadopoulos is with the Department of Informatics and Telematics, Harokopio University of Athens, 9 Omirou St., Tavros, GR-17778, Athens, Greece - E-Mail: g.th.papadopoulos@hua.gr

** V. Argyriou is with the Department of Networks and Digital Media, Kingston University, London, U.K. - E-Mail: vasileios.argyriou@kingston.ac.uk

‡ P. Radoglou-Grammatikis and P. Sarigiannidis is with the Department of Electrical and Computer Engineering, University of Western Macedonia, ZEP Campus, 50100, Kozani, Greece - E-Mail: pradoglou@uowm.gr, psarigiannidis@uowm.gr

establishment, modification, and deletion. Because PFCP operates on the 5G Core, successful attacks can disrupt user-plane forwarding and service availability. Intrusion detection systems for PFCP must therefore differentiate between benign traffic and multiple attack types in a multiclass setting, often under tight resource and latency constraints.

This paper focuses on the classification of multiclass PFCP attacks, using PFCP datasets with varying aggregation windows [12], and systematically compares classical baselines, quantum-inspired kernels, and trainable quantum machine learning models in a shared preprocessing and evaluation pipeline. Our contributions are:

- A unified PFCP multiclass benchmark that spans classical baselines, quantum-inspired kNN, parametrized quantum classifiers (PQCs), and trainable quantum kernels using the quantum kernel alignment method (QKA).
- Empirical evidence that quantum-inspired kernels can outperform classical baselines on PFCP, while trainable PQCs and QKA remain sensitive to data size and kernel scaling choices.
- A reproducible pipeline for PFCP datasets benchmarking to support follow-on QML research in 5G security.

The remainder of this paper is organized as follows. Section II reviews related work, Section III presents background, Section IV describes the methodology, and Section V reports and discusses the results.

## II. Related Work

Existing literature provides a broad context for applying machine learning and quantum machine learning to intrusion detection. For mobile and telecom networks, AI-based IDS surveys summarize common feature extraction pipelines, learning paradigms, and deployment constraints [13]. On the QML side, overview works discuss where quantum models may offer representational advantages and where near-term limitations, including noise, barren plateau and limited number of qubits constrain practical impact [14], [15], [16]. Most directly related to this work, the systematic literature review in [17] surveys QML approaches for network IDS and highlights that evaluations are often performed on small or non-telecom datasets and with limited multiclass realism.

Within the 5G Core setting, IDS research must address high-dimensional, heterogeneous control-plane statistics and multiclass attack taxonomies. The PFCP APP Layer dataset introduced by Amponis *et al.* provides a realistic benchmark by emulating benign 5G Core operation and multiple PFCP attack scenarios [12]. Building on this dataset, Radoglou *et al.* propose a multiclass PFCP IDS and complement the predictive model with explainability mechanisms for analyst support [18]. These works establish the data foundation and classical IDS baselines for our study.

Early work on quantum-enhanced intrusion detection explored using quantum kernels and quantum-inspired similarity measures within classical learning pipelines [19]. More recent contributions propose hybrid or ensemble approaches that aim to improve robustness and performance on security datasets, including quantum-secure ensemble IDS designs for cloud environments [20] and hybrid QML IDS implementations evaluated on public datasets [21]. Other studies evaluate near-term quantum neural network models for anomaly detection under realistic noise and hardware constraints [22], [23].

Complementary lines of work investigate specialized QML formulations for intrusion detection, such as outlier-analysis-based quantum IDS designs [24] and quantum reservoir computing approaches targeting edge/IoT constraints [25]. At a more general algorithmic level, our work is also connected to quantum kernel methods and trainable feature maps [26], [27], [28], as well as parametrized quantum classifiers based on hardware-efficient and data-reuploading ansatzes for multiclass learning [29], [30], [31].

Compared with prior studies mentioned above, our work differs in three main aspects. First, we focus on PFCP control-plane traffic from 5G Core networks rather than generic enterprise, cloud, or IoT intrusion-detection datasets, making the benchmark closer to telecom-security deployment scenarios. Second, we evaluate classical, quantum-inspired, and trainable quantum approaches under the same preprocessing, splitting, and evaluation protocol, reducing confounding effects caused by different feature engineering pipelines or dataset partitions. Third, rather than treating QML as a single model family, we separate analytic quantum-inspired kernels, parametrized quantum classifiers, and trainable quantum kernels, and discuss their different optimization and scaling behavior. This positions the study as a comparative benchmark of practical QML design choices for 5G security rather than as a claim of unconditional quantum advantage.

Motivated by the above, we contribute a unified benchmark for multiclass PFCP intrusion detection that compares strong classical baselines against quantum-inspired kernels, parametrized quantum classifiers, and trainable quantum kernels under a shared preprocessing and evaluation pipeline. We further provide empirical evidence on when quantum-inspired kernels are competitive on realistic PFCP data and release a reproducible evaluation workflow to support follow-on research.

## III. Background

Table I summarizes the main symbols and notation used throughout the paper.

### A. Multiclass Network Intrusion Detection

Network IDS aim to classify network traffic as benign or as belonging to specific attack categories. They are typically categorized as signature-based or anomaly-based: the former often yields low false-positive rates but cannot detect zero-day attacks, while the latter can generalize to previously unseen attacks at the cost of higher false-positive rates [9].

In the multiclass setting, such systems map an input feature vector to a discrete label,

$$\mathbf{x} \in \mathbb{R}^d \;\mapsto\; y \in \{1, \ldots, C\}, \tag{1}$$

where $\mathbf{x}$ denotes the $d$-dimensional feature vector, $y$ is the predicted class label, and $C$ is the number of attack types.

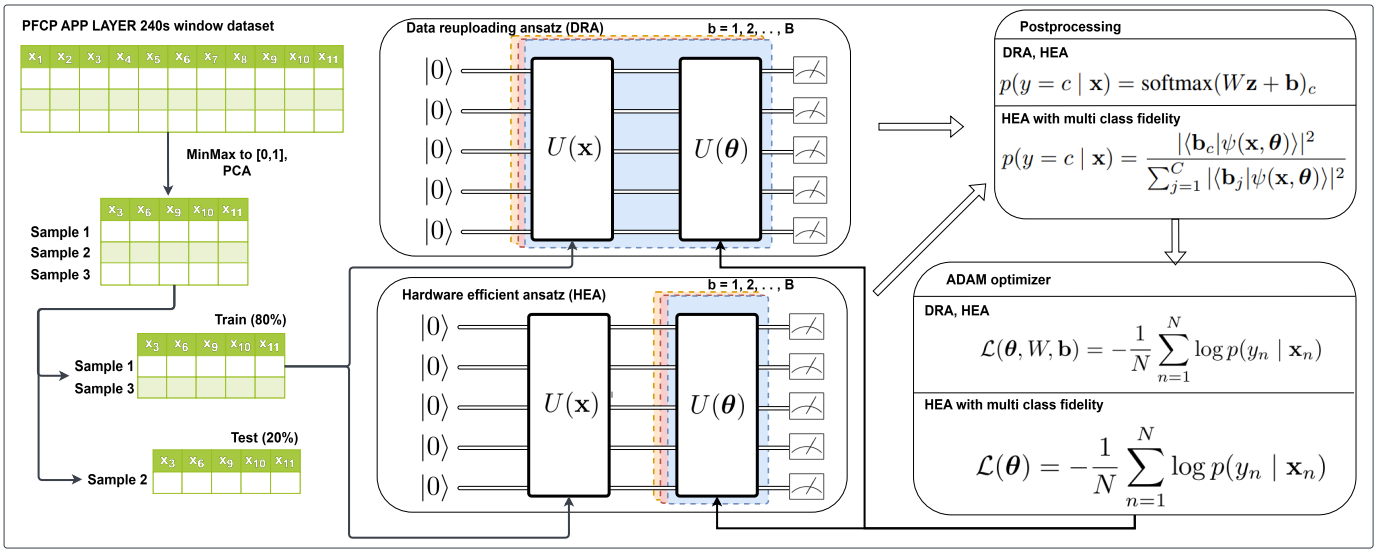



Fig. 1. Training pipeline for the parametrized quantum classifiers. The hardware-efficient ansatz encodes the data once and applies blocks of trainable parameters, followed by either a softmax readout or a multiclass fidelity readout. The data-reuploading ansatz alternates between data-encoding and trainable blocks across multiple layers.

TABLE I
SYMBOLS AND NOTATION.

| Symbol | Meaning |
|---|---|
| $\mathbf{x} \in \mathbb{R}^d$ | Feature vector with $d$ dimensions. |
| $d$ | Feature dimension. |
| $y \in \{1, \dots, C\}$ | Class label among $C$ classes. |
| $C$ | Number of classes. |
| $N$ | Number of samples. |
| $k$ | Number of neighbors in kNN. |
| $q$ | Number of qubits in the quantum circuit. |
| $B$ | Number of blocks/layers in a parametrized quantum ansatz. |
| $U(\mathbf{x})$ | Data-encoding circuit parameterized by $\mathbf{x}$. |
| $U(\boldsymbol{\theta})$ | Trainable circuit block with parameters $\boldsymbol{\theta}$. |
| $\lvert\psi(\mathbf{x})\rangle$ | Quantum state produced by encoding $\mathbf{x}$. |
| $K(\mathbf{x}, \mathbf{x}')$ | Kernel between samples $\mathbf{x}$ and $\mathbf{x}'$. |
| $z_i$ | Measured expectation value used for readout. |
| $W, \mathbf{b}$ | Trainable linear readout parameters. |
| $p(y = c \mid \mathbf{x})$ | Predicted probability of class $c$. |
| $\mathcal{L}(\boldsymbol{\theta})$ | Training loss. |
| $(K_y)_{ij}$ | Ideal label kernel entry. |
| $\mathcal{A}(\boldsymbol{\theta})$ | Kernel alignment objective. |
| $\langle A, B\rangle_F$ | Frobenius inner product. |
| $\lVert A\rVert_F$ | Frobenius norm. |

Common strategies for multiclass classification in the literature include One-vs.-Rest (OVR) and One-vs.-One (OVO). In OVR, $C$ binary classifiers are trained, each distinguishing one class from all others. In OVO, $C(C-1)/2$ binary classifiers are trained for all class pairs, and the final label is determined via a voting scheme. Neural networks, on the other hand, produce a probability distribution $\mathbf{p}$ over all classes; given a logit vector $\mathbf{z} \in \mathbb{R}^C$, the softmax assigns

$$p_i = \frac{e^{z_i}}{\sum_{j=1}^{C} e^{z_j}}. \tag{2}$$

## B. Classical Baselines

Classical machine learning methods are widely used in intrusion detection systems because they provide strong baselines with well-understood accuracy–complexity trade-offs. In particular, support vector machines (SVMs) are widely used in intrusion detection research due to their strong theoretical foundations and effectiveness in high-dimensional feature spaces. SVMs learn a decision boundary by maximizing the margin between classes, typically via

$$\min_{\mathbf{w},b,\boldsymbol{\xi}} \ \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N} \xi_i, \tag{3}$$

subject to

$$y_i(\mathbf{w}^T\phi(\mathbf{x}_i) + b) \geq 1 - \xi_i, \qquad \xi_i \geq 0. \tag{4}$$

A comprehensive comparative analysis of AI-based intrusion detection systems is presented in [18], where the authors compare tree-based and margin-based classifiers on the 5GC PFCP dataset, establishing the classical baselines used in this work.

## C. Quantum Machine Learning In The NISQ Era

Near-term quantum processors are characterized by limited qubit counts and imperfect gates, with decoherence and measurement noise constraining circuit depth and the number of reliable operations [32]. These limitations motivate hybrid quantum–classical learning pipelines in which a small quantum circuit is trained or queried as part of a classical optimization loop [15], [16].

From the perspective of intrusion detection, the motivation for quantum machine learning is not that quantum models are universally better, but that they can induce alternative similarity measures and representations for high-dimensional data. In particular, quantum feature maps embed a classical vector $\mathbf{x}$ into a quantum state $|\psi(\mathbf{x})\rangle$ living in a $2^q$-dimensional Hilbert space, potentially yielding decision boundaries that are difficult to obtain with simple classical metrics at comparable parameter counts [33], [26]. In the NISQ regime, any advantage is ultimately empirical and must be evaluated under realistic resource and noise constraints [16].

At a basic level, a single qubit is a normalized superposition $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ with $|\alpha|^2 + |\beta|^2 = 1$, and measurements return classical outcomes according to these amplitudes [34]. Multi-qubit registers generalize this notion, and quantum circuits apply unitary transformations to map an initial state to an output state whose measured statistics are used for learning.

In this paper we consider two common NISQ-compatible QML families. The first is parametrized quantum classifiers, which implement a trainable circuit $|\psi(\mathbf{x}, \boldsymbol{\theta})\rangle$ and optimize $\boldsymbol{\theta}$ by minimizing a classical loss based on measured observables [29], [35]. Data reuploading increases expressivity by re-encoding the input features across multiple layers and can yield universal classification with shallow circuits [30]. Multiclass predictions can be obtained through expectation-based readouts or by associating each class with a computational-basis bitstring and using a fidelity-style score [31].

The second family is quantum kernel methods, where the circuit defines a feature map $\mathbf{x} \mapsto |\psi(\mathbf{x})\rangle$ and a kernel given by the state fidelity $K(\mathbf{x}, \mathbf{x}') = |\langle\psi(\mathbf{x})|\psi(\mathbf{x}')\rangle|^2$ [26], [27]. The resulting kernel can be used in a classical SVM-like pipeline, while trainable quantum kernels tune circuit parameters to improve class separation, for example through kernel alignment objectives [28]. A practical challenge is that overly expressive feature maps can lead to concentration of kernel values, reducing discrimination and making training less reliable [36].

## IV. Methodology

This section describes the end-to-end evaluation pipeline used in this work. We first present the dataset preprocessing and feature embedding steps for each aggregation window, then describe the classical baselines, quantum-inspired methods, and QML models, and finally summarize the evaluation protocol used to produce the reported results.

### A. Data Preprocessing and Embedding

We use the PFCP APP Layer dataset introduced in [12], which contains aggregated PFCP control-plane statistics derived from emulated 5G Core traffic under both benign operation and multiple PFCP attack scenarios. The dataset is provided for five aggregation windows: 15, 20, 60, 120, and 240 seconds. Each window produces a separate dataset instance with identical feature semantics but different temporal granularity.

Across all window sizes, each sample consists of 10 PFCP statistical features and belongs to one of five classes: `Normal`, `Mal_Del`, `Mal_Estab`, `Mal_Mod`, and `Mal_Mod2`. All dataset variants are class-balanced using the splits defined in [12], with a fixed training and test partition per window. Unless otherwise stated, we evaluate all models independently on each aggregation window to assess the impact of temporal resolution on classification performance.

For all methods, we apply MinMax normalization to scale each feature to the interval $[0, 1]$. Class labels are encoded as integer indices for multiclass learning. Classical baselines and the quantum-inspired k-nearest neighbors model operate on the full 10-dimensional feature space.

For parametrized quantum classifiers, we reduce the input dimensionality using Principal Component Analysis (PCA) to limit qubit requirements [37]. Specifically, we retain the first five principal components, which preserve approximately 99% of the total variance across all aggregation windows. The resulting five-dimensional feature vectors are mapped to quantum rotation angles by scaling to the interval $[0, \pi]$.

For trainable quantum kernel methods using the QKA method, input features are similarly scaled to $[0, \pi]$. All reported performance metrics include accuracy, macro-averaged precision, recall, and F1 score, computed on the held-out test split for each aggregation window.

### B. Quantum-Inspired k-Nearest Neighbors with SWAP-Test Kernels

The quantum-inspired k-nearest neighbors (qkNN) model is a classical instance-based learning algorithm that employs analytic kernels motivated by quantum amplitude encoding and the SWAP test. Rather than executing a quantum circuit, qkNN analytically computes the similarity that would arise from comparing quantum states prepared from classical data [38].

Given a feature vector $\mathbf{x} \in [0, 1]^d$, each scalar feature $x_j$ is mapped to a two-dimensional normalized representation using amplitudes $\sqrt{x_j}$ and $\sqrt{1 - x_j}$. The similarity between two samples $\mathbf{x}$ and $\mathbf{x}'$ is then defined as

$$K(\mathbf{x}, \mathbf{x}') = \prod_{j=1}^{d} \left( \sqrt{x_j x'_j} + \sqrt{(1 - x_j)(1 - x'_j)} \right). \quad (5)$$

For classification, qkNN follows the standard k-nearest neighbors paradigm. Given a query sample, its similarity to all training points is computed using the kernel above. The top-$k$ most similar samples are selected, and the predicted label is obtained via similarity-weighted voting, where neighbors with higher kernel values contribute more strongly to the final decision.

### C. Parametrized Quantum Classifiers

Parametrized quantum classifiers encode features using parameterized rotations. We compare a Data-Reuploading Ansatz (DRA) and a Hardware-Efficient Ansatz (HEA) with two readouts: Pauli-$Z$ expectations and a multi-qubit fidelity readout that maps each class to a computational-basis bitstring.

Inputs are scaled by $\pi$ before rotation. Unless otherwise stated, we use $B = 4$ blocks, batch size 32, learning rate 0.01, and 100 epochs.

Fig. 1 illustrates the full training pipeline used for the parametrized quantum classifiers. Starting from a PFCP aggregation-window dataset, we apply MinMax scaling to $[0, 1]$ and dimensionality reduction, then split the samples into training and test sets. The resulting feature vector is encoded by a data-embedding circuit $U(\mathbf{x})$ and processed by a trainable block $U(\boldsymbol{\theta})$ using either the data-reuploading ansatz or the hardware-efficient ansatz, repeated for $b = 1, \ldots, B$ blocks.

TABLE II
PFCP 15S WINDOW RESULTS.

| Method | Accuracy | Macro Precision | Macro Recall | Macro F1 | Weighted F1 |
|---|---|---|---|---|---|
| svm-rbf | 0.5420 | 0.5159 | 0.5420 | 0.5143 | 0.5143 |
| **qknn** | **0.6314** | **0.6464** | **0.6314** | **0.6346** | **0.6347** |
| qka trainable ZZ | 0.5750 | 0.5427 | 0.5750 | 0.5417 | 0.5417 |
| qka suzuki phi12 | 0.5875 | 0.5450 | 0.5875 | 0.5506 | 0.5506 |
| xgboost | 0.6150 | 0.6357 | 0.6150 | 0.6191 | 0.6191 |
| hardware efficient ansatz | 0.5511 | 0.4806 | 0.5511 | 0.4849 | 0.4849 |
| multi class ansatz | 0.5506 | 0.4806 | 0.5506 | 0.4845 | 0.4845 |
| data reuploading ansatz | 0.5511 | 0.4806 | 0.5511 | 0.4849 | 0.4849 |

TABLE III
PFCP 60S WINDOW RESULTS.

| Method | Accuracy | Macro Precision | Macro Recall | Macro F1 | Weighted F1 |
|---|---|---|---|---|---|
| svm-rbf | 0.5535 | 0.5118 | 0.5521 | 0.4716 | 0.4735 |
| qknn | 0.5800 | 0.5863 | 0.5800 | 0.5799 | 0.5799 |
| qka trainable ZZ | 0.5200 | 0.4850 | 0.5200 | 0.4944 | 0.4944 |
| qka suzuki phi12 | 0.5000 | 0.4836 | 0.5000 | 0.4796 | 0.4796 |
| **xgboost** | **0.5933** | **0.6076** | **0.5921** | **0.5961** | **0.5974** |
| hardware efficient ansatz | 0.5500 | 0.5381 | 0.5500 | 0.5239 | 0.5239 |
| multi class ansatz | 0.5600 | 0.5374 | 0.5600 | 0.5309 | 0.5309 |
| data reuploading ansatz | 0.5700 | 0.5398 | 0.5700 | 0.5393 | 0.5393 |

TABLE IV
PFCP 120S WINDOW RESULTS.

| Method | Accuracy | Macro Precision | Macro Recall | Macro F1 | Weighted F1 |
|---|---|---|---|---|---|
| svm-rbf | 0.5316 | 0.5202 | 0.5292 | 0.4823 | 0.4857 |
| **qknn** | **0.5949** | **0.6327** | **0.5929** | **0.6055** | **0.6080** |
| qka trainable ZZ | 0.5125 | 0.5238 | 0.5125 | 0.4665 | 0.4665 |
| qka suzuki phi12 | 0.5125 | 0.5238 | 0.5125 | 0.4665 | 0.4665 |
| xgboost | 0.5823 | 0.6036 | 0.5801 | 0.5886 | 0.5911 |
| hardware efficient ansatz | 0.5250 | 0.5018 | 0.5250 | 0.4880 | 0.4880 |
| multi class ansatz | 0.4625 | 0.4615 | 0.4625 | 0.3968 | 0.3968 |
| data reuploading ansatz | 0.5250 | 0.5184 | 0.5250 | 0.4869 | 0.4869 |

For reuploading, the circuit alternates between data encoding and trainable blocks:

$$|\psi_{\text{DRA}}(\mathbf{x},\boldsymbol{\theta})\rangle = \prod_{b=1}^{B} U_b(\boldsymbol{\theta})\, U_b(\mathbf{x})\, |0\rangle^{\otimes q}. \tag{6}$$

We use single-qubit rotations for encoding, $U(\mathbf{x}) = \prod_{i=1}^{q} R_X(\tilde{x}_i)$ with $\tilde{x}_i = \pi x_i$, and trainable layers $U(\boldsymbol{\theta}) = \prod_{i=1}^{q} R_Z(\theta_{b,i}^{(1)}) R_Y(\theta_{b,i}^{(2)}) R_Z(\theta_{b,i}^{(3)})$ interleaved with linear entanglers. The HEA variant uses fewer trainable rotations per block and encodes data once,

$$|\psi_{\text{HEA}}(\mathbf{x},\boldsymbol{\theta})\rangle = \prod_{b=1}^{B} U_b(\boldsymbol{\theta})\, U(\mathbf{x})\, |0\rangle^{\otimes q}. \tag{7}$$

For expectation-based readouts, the circuit outputs $z_i = \langle Z_i \rangle$ and predicts

$$p(y = c \mid \mathbf{x}) = \text{softmax}(W\mathbf{z} + \mathbf{b})_c. \tag{8}$$

For the multi-qubit fidelity readout, each class $c$ is assigned a bitstring $\mathbf{b}_c$ and the probability is

$$p(y = c \mid \mathbf{x}) = \frac{|\langle \mathbf{b}_c | \psi(\mathbf{x},\boldsymbol{\theta})\rangle|^2}{\sum_{j=1}^{C} |\langle \mathbf{b}_j | \psi(\mathbf{x},\boldsymbol{\theta})\rangle|^2}. \tag{9}$$

All PQC variants are optimized end-to-end with Adam on the multi-class cross-entropy loss,

$$\mathcal{L}(\boldsymbol{\theta}) = -\frac{1}{N} \sum_{n=1}^{N} \log p(y_n \mid \mathbf{x}_n). \tag{10}$$

### D. Quantum Kernel Alignment

QKA learns feature-map parameters $\boldsymbol{\theta}$ by aligning a quantum kernel with the ideal label kernel. We use trainable ZZ and Suzuki $\phi_{12}$ feature maps [39] and define

$$K_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{x}') = |\langle \psi_{\boldsymbol{\theta}}(\mathbf{x}) | \psi_{\boldsymbol{\theta}}(\mathbf{x}')\rangle|^2. \tag{11}$$

Let $(K_y)_{ij} = 1$ if $y_i = y_j$ and 0 otherwise. Following the multi-class alignment objective in [28], we maximize the centered, normalized kernel alignment

$$\mathcal{A}(\boldsymbol{\theta}) = \frac{\langle \tilde{K}_{\boldsymbol{\theta}}, \tilde{K}_y \rangle_F}{\|\tilde{K}_{\boldsymbol{\theta}}\|_F \, \|\tilde{K}_y\|_F}. \tag{12}$$

The objective is estimated on randomly sampled sub-kernels to reduce compute. We use COBYLA with subset size 32, four sub-kernels per iteration, and 100 iterations.

## V. Results and Discussion

All results reported in this section are obtained from noiseless classical simulations of the corresponding quantum circuits, in order to isolate algorithmic behavior from hardware-induced noise. Fig. 2 summarizes the kernel-based methods on the PFCP-240s window, while Fig. 3 compares XGBoost against the parametrized quantum classifier variants on the same window.

### A. Classical Baseline Performance

Table V summarizes the classical baselines on the PFCP-240s window. XGBoost is the strongest classical method, reaching accuracy 59.9% and macro F1 60.3%, while the RBF SVM is competitive but slightly lower. Across aggregation windows, XGBoost remains consistently strong, suggesting that tree ensembles capture nonlinear feature interactions that are not fully exploited by margin-based classifiers under default hyperparameters.

### B. Performance of the Quantum-Inspired k-Nearest Neighbors Classifier

The quantum-inspired qkNN achieves the best overall performance on the PFCP-240s window, outperforming both XGBoost and SVM-RBF. This indicates that the analytic SWAP-test similarity can define a useful metric for PFCP traffic in a low-sample regime, effectively acting as an expressive kernel without requiring circuit execution.

Across window sizes, qkNN remains competitive and typically improves over classical SVM-RBF. Notably, performance varies with the aggregation window, reflecting a trade-off between shorter windows and longer windows. The runtime–accuracy view in Fig. 2 further shows that qkNN achieves the highest test accuracy among the kernel-based methods on the 240 s window, while remaining substantially cheaper than trainable quantum-kernel optimization. Its main computational cost is the pairwise kernel computation between test and

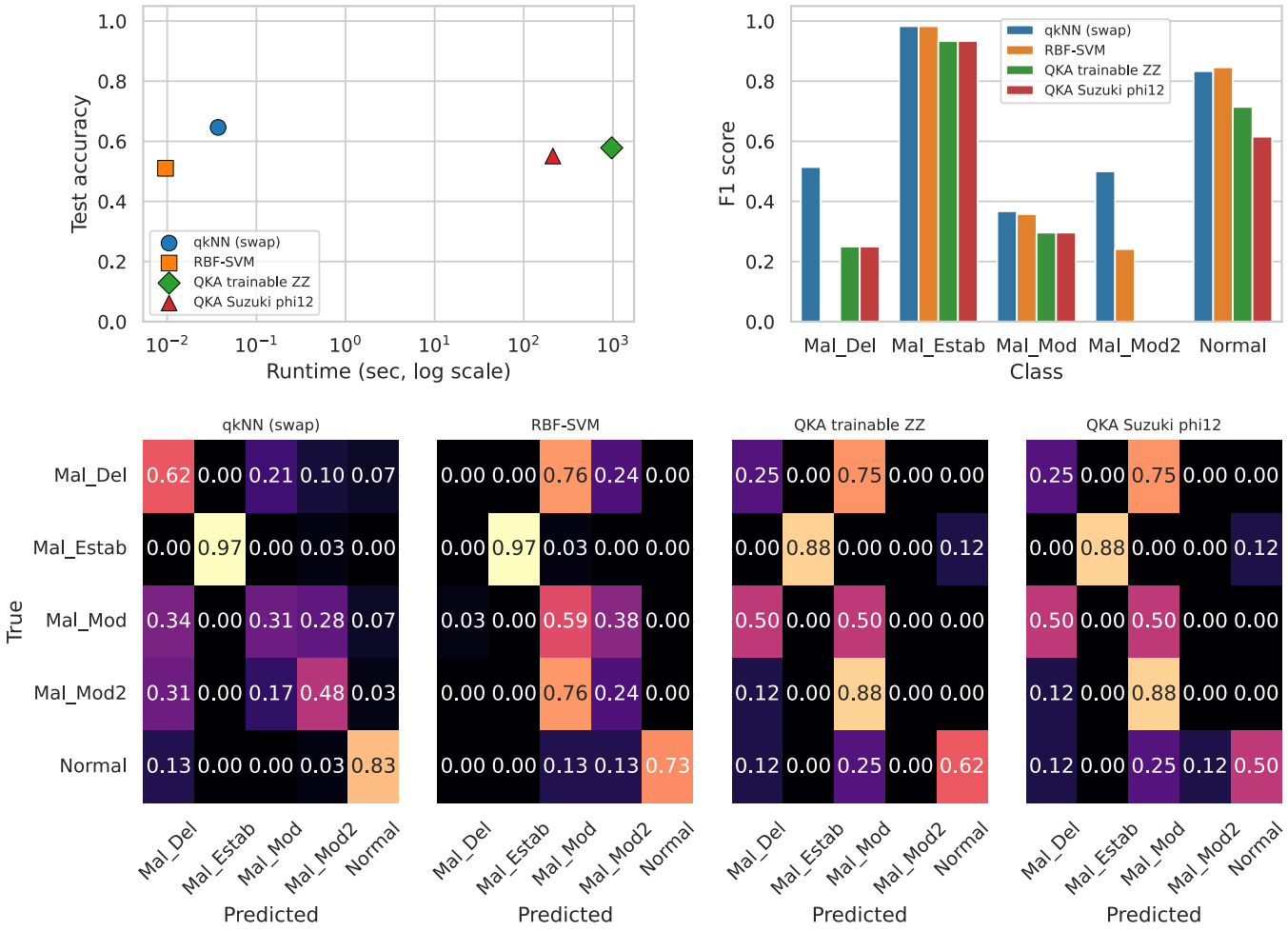



Fig. 2. Results for the classical kernel, quantum-inspired kernel, and trainable quantum kernel methods using the PFCP APP Layer dataset with 240-second time windows.

TABLE V
PFCP 240S WINDOW RESULTS.

| Method | Accuracy | Macro Precision | Macro Recall | Macro F1 | Weighted F1 |
|---|---|---|---|---|---|
| svm-rbf | 0.5510 | 0.5429 | 0.5478 | 0.5169 | 0.5216 |
| **qknn** | **0.6211** | **0.6355** | **0.6175** | **0.6157** | **0.6195** |
| qka trainable ZZ | 0.4125 | 0.5221 | 0.4125 | 0.4263 | 0.4263 |
| qka suzuki phi12 | 0.4500 | 0.4981 | 0.4500 | 0.4460 | 0.4460 |
| xgboost | 0.5986 | 0.6215 | 0.5952 | 0.6029 | 0.6068 |
| hardware efficient ansatz | 0.5027 | 0.4937 | 0.4989 | 0.4613 | 0.4668 |
| multi class ansatz | 0.5177 | 0.5462 | 0.5139 | 0.4949 | 0.5000 |
| data reuploading ansatz | 0.5075 | 0.5126 | 0.5037 | 0.4740 | 0.4793 |

training samples, since prediction is instance-based. However, because the SWAP-test kernel is evaluated analytically and does not require quantum-circuit execution or iterative parameter training, the method is comparatively simple to reproduce. In practical deployments, this inference-time overhead could be reduced through approximate-nearest-neighbor search, prototype selection, or caching of kernel values.

### C. Trainable Quantum Classifiers

Trainable quantum classifiers are more sensitive to optimization and resource constraints. On PFCP-240s, parametrized quantum classifier variants are below qkNN and XGBoost but remain competitive with classical SVMs. The multi-class fidelity readout gives the best PQC performance on this window, while reuploading and HEA achieve lower macro F1 values. Fig. 3 provides the corresponding comparison at 240s, including the runtime–accuracy trade-off on a logarithmic runtime scale. The figure shows that the parametrized quantum models require substantially higher runtime than XGBoost while not improving the final test accuracy, highlighting the current practical cost of trainable circuit-based models in this benchmark.

QKA underperforms on this benchmark in our runs, with accuracy 41.3%–45.0% depending on the feature map. This gap is consistent with known difficulties of optimizing trainable kernels under limited data and noisy objective estimates, as well as kernel scaling effects and concentration in expressive feature maps [36]. From a computational perspective, PQCs require repeated circuit evaluations over many epochs, while QKA repeatedly estimates sub-kernels during the alignment procedure. Since full kernel construction scales quadratically with the number of training samples, QKA is particularly sensitive to dataset size and sub-kernel sampling choices.

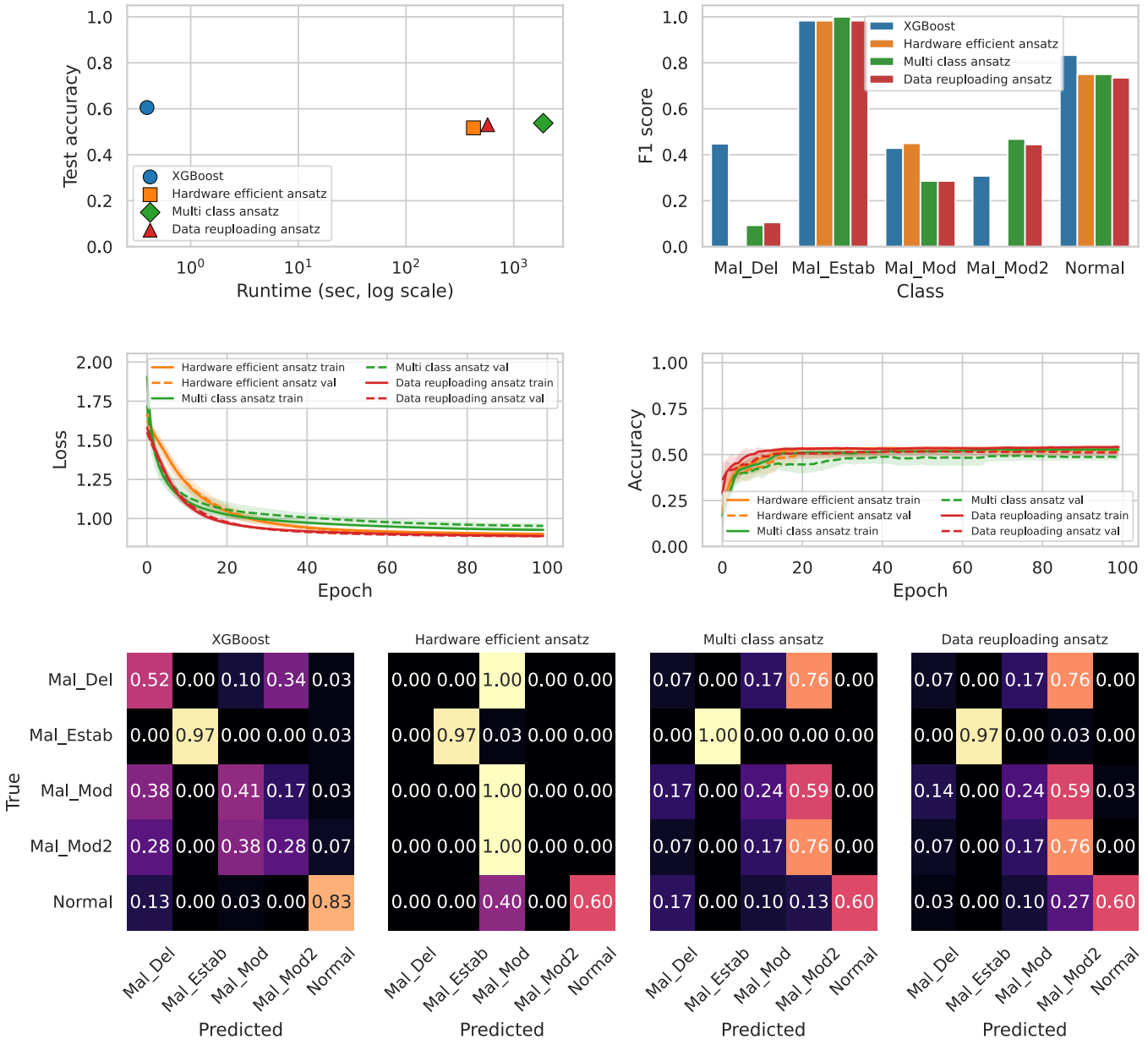



Fig. 3. Results for classical XGBoost and the parametrized quantum classifiers using the PFCP APP Layer dataset with 240-second time windows.

These training-time trade-offs help explain why the strongest practical outcome in this study is obtained by the analytic quantum-inspired kernel rather than by fully trainable quantum models.

## VI. Conclusion

We presented a unified benchmark for multiclass PFCP attack detection on the PFCP APP Layer dataset [12], comparing classical baselines, a quantum-inspired kernel method, parametrized quantum classifiers, and trainable quantum kernels under a consistent pipeline. Across aggregation windows and evaluation metrics, the analytic qkNN kernel is the strongest performer among the evaluated methods under the fixed benchmark protocol, outperforming both SVM-RBF and XGBoost on the 240 s window and remaining competitive across other window sizes. In contrast, PQC variants achieve only modest improvements over classical SVMs and remain below the quantum-inspired kernel, while QKA underperforms on this dataset, consistent with kernel-scaling effects and noisy alignment estimates in low-sample regimes.

Overall, the results suggest that quantum-inspired kernels are currently the most practical QML option for PFCP intrusion detection at realistic dataset sizes, while trainable quantum models require more robust optimization and larger training sets to deliver reliable improvements. Future work will extend the evaluation to larger PFCP datasets and additional window sizes, explore richer feature maps and scaling strategies, and assess robustness under realistic deployment constraints and hardware noise.

## REFERENCES


[1] K. Samunnisa, G. S. V. Kumar, and K. Madhavi, "Intrusion detection system in distributed cloud computing: Hybrid clustering and classification methods," *Measurement: Sensors*, vol. 25, p. 100612, 2023.

[2] Z. Azam, M. M. Islam, and M. N. Huda, "Comparative analysis of intrusion detection systems and machine learning-based model analysis through decision tree," *Ieee Access*, vol. 11, pp. 80 348–80 391, 2023.

[3] M. Maithem and G. A. Al-Sultany, "Network intrusion detection system using deep neural networks," in *Journal of Physics: Conference Series*, vol. 1804, no. 1. IOP Publishing, 2021, p. 012138.

[4] A. Halbouni, T. S. Gunawan, M. H. Habaebi, M. Halbouni, M. Kartiwi, and R. Ahmad, "Cnn-lstm: hybrid deep neural network for network intrusion detection system," *IEEE Access*, vol. 10, pp. 99 837–99 849, 2022.

[5] S. Antony Joseph Raj and M. Madiajagan, "Hamc-id: hybrid attention-based meta-classifier for intrusion detection," *Scientific Reports*, 2025.

[6] A. A. Najar, O. M. Khanday, V. Hnamte, M. N. Sugali, M. A. Jawad, N. Farooq, and M. R. Lone, "Lids: A novel lightweight intrusion detection system for ddos attacks," *Security and Privacy*, vol. 9, no. 1, p. e70148, 2026.

[7] C. Iwendi, S. Khan, J. H. Anajemba, M. Mittal, M. Alenezi, and M. Alazab, "The use of ensemble models for multiple class and binary class classification for improving intrusion detection systems," *Sensors*, vol. 20, no. 9, p. 2559, 2020.

[8] S. Sadhwani, M. A. H. Khan, R. Muthalagu, P. M. Pawar, and K. Suresh, "A hybrid bilstm-cnn approach for intrusion detection for iot applications," *Scientific Reports*, 2025.

[9] F. A. Khan, A. A. Shah, N. Alshammry, S. Saif, W. Khan, M. O. Malik, and Z. Ullah, "Balanced multi-class network intrusion detection using machine learning," *IEEE Access*, 2024.

[10] S. Shamim, Y. Kodera, and Y. Nogami, "Gan-based ids system for imbalanced datasets in sdn environments with offline and real-time traffic analysis," *IEEE Access*, vol. 13, pp. 190 786–190 806, 2025.

[11] H. Dong and I. Kotenko, "Towards ultra-low latency: Binarized neural network architectures for in-vehicle network intrusion detection," *arXiv preprint arXiv:2511.00828*, 2025.

[12] G. Amponis, P. Radoglou-Grammatikis, G. Nakas, M. Zevgara, S. Giannakidou, S. Ouzounidis, G. Kakamoukas, and A. Sarigiannidis, "5gc pfcp intrusion detection dataset," 2023. [Online]. Available: https://zenodo.org/records/7888347

[13] A. Boukerche and R. E. De Grande, "Artificial intelligence-based intrusion detection systems for mobile networks: A survey," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 2, pp. 1332–1364, 2021.

[14] M. Schuld and F. Petruccione, "Quantum machine learning: An overview of theory and applications," *Contemporary Physics*, vol. 59, no. 2, pp. 172–185, 2018.

[15] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, "Quantum machine learning," *Nature*, vol. 549, no. 7671, pp. 195–202, 2017.

[16] M. Cerezo, G. Verdon, H.-Y. Huang, L. Cincio, and P. J. Coles, "Challenges and opportunities in quantum machine learning," *Nature computational science*, vol. 2, no. 9, pp. 567–576, 2022.

[17] O. K. Nicesio, A. G. Leal, and V. L. Gava, "Quantum machine learning for network intrusion detection systems, a systematic literature review," in *2023 IEEE 2nd International Conference on AI in Cybersecurity (ICAIC)*, 2023, pp. 1–6.

[18] P. Radoglou-Grammatikis, G. Nakas, G. Amponis, S. Giannakidou, T. Lagkas, V. Argyriou, S. Goudos, and P. Sarigiannidis, "5gcids: An intrusion detection system for 5g core with ai and explainability mechanisms," in *2023 IEEE Globecom Workshops (GC Wkshps)*. IEEE, 2023, pp. 353–358.

[19] A. Gouveia and M. Correia, "Towards quantum-enhanced machine learning for network intrusion detection," in *2020 IEEE 19th International Symposium on Network Computing and Applications (NCA)*, 2020, pp. 1–8.

[20] D. B. Salvakkam, V. Saravanan, P. K. Jain, and R. Pamula, "Enhanced quantum-secure ensemble intrusion detection techniques for cloud based on deep learning," *Cognitive Computation*, vol. 15, no. 5, pp. 1593–1612, 2023.

[21] D. Abreu, D. Moura, C. Esteve Rothenberg, and A. Abelém, "Quantum-netsec: Quantum machine learning for network security," *International Journal of Network Management*, vol. 35, no. 4, p. e70018, 2025.

[22] A. Kukliansky, M. Orescanin, C. Bollmann, and T. Huffmire, "Network anomaly detection using quantum neural networks on noisy quantum computers," *IEEE Transactions on Quantum Engineering*, vol. 5, pp. 1–11, 2024.

[23] M. Hdaib, S. Rajasegarar, and L. Pan, "Quantum deep learning-based anomaly detection for enhanced network security," *Quantum Machine Intelligence*, vol. 6, no. 1, p. 26, 2024.

[24] T. H. Kim and S. Madhavi, "Quantum intrusion detection system using outlier analysis," *Scientific Reports*, vol. 14, no. 1, p. 27114, 2024.

[25] T. Ahmad, S. Khan, N. Nigar, E. Mohamed Obaid, and S. Naaz, "Quantum reservoir-based intrusion detection for edge-iot systems," *IEEE Transactions on Consumer Electronics*, vol. 71, no. 4, pp. 11 269–11 278, 2025.

[26] M. Schuld and N. Killoran, "Quantum machine learning in feature hilbert spaces," *Physical review letters*, vol. 122, no. 4, p. 040504, 2019.

[27] V. Havlicek, A. D. Córcoles, K. Temme, A. W. Harrow, A. Kandala, J. M. Chow, and J. M. Gambetta, "Supervised learning with quantum-enhanced feature spaces," arXiv preprint, 2018. [Online]. Available: https://arxiv.org/abs/1804.11326

[28] T. Hubregtsen, D. Wierichs, E. Gil-Fuster, P.-J. H. S. Derks, P. K. Faehrmann, and J. J. Meyer, "Training quantum embedding kernels on near-term quantum computers," arXiv preprint, 2021. [Online]. Available: https://arxiv.org/abs/2105.02276

[29] M. Schuld, A. Bocharov, K. Svore, and N. Wiebe, "Circuit-centric quantum classifiers," arXiv preprint, 2018. [Online]. Available: https://arxiv.org/abs/1804.00633

[30] A. Pérez-Salinas, A. Cervera-Lierta, E. Gil-Fuster, and J. I. Latorre, "Data re-uploading for a universal quantum classifier," *Quantum*, vol. 4, p. 226, 2020.

[31] Y. Du, Y. Yang, D. Tao, and M.-H. Hsieh, "Problem-dependent power of quantum neural networks on multi-class classification," arXiv preprint, 2022. [Online]. Available: https://arxiv.org/abs/2301.01597

[32] J. Preskill, "Quantum computing in the nisq era and beyond," *Quantum*, vol. 2, p. 79, 2018.

[33] M. Schuld, I. Sinayskiy, and F. Petruccione, "An introduction to quantum machine learning," *Contemporary Physics*, vol. 56, no. 2, pp. 172–185, 2015.

[34] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*, 10th ed. Cambridge University Press, 2010.

[35] E. Farhi and H. Neven, "Classification with quantum neural networks on near term processors," arXiv preprint, 2018. [Online]. Available: https://arxiv.org/abs/1802.06002

[36] S. Thanasilp, S. Wang, M. Cerezo, and Z. Holmes, "Exponential concentration in quantum kernel methods," *Nature Communications*, vol. 15, no. 1, p. 5200, 2024. [Online]. Available: https://www.nature.com/articles/s41467-024-49287-w

[37] M. Greenacre, P. J. Groenen, T. Hastie, A. I. d'Enza, A. Markos, and E. Tuzhilina, "Principal component analysis," *Nature Reviews Methods Primers*, vol. 2, no. 1, p. 100, 2022.

[38] A. Basheer, A. Afham, and S. K. Goyal, "Quantum $k$-nearest neighbors algorithm," *arXiv preprint arXiv:2003.09187*, 2020.

[39] Y. Suzuki, H. Yano, Q. Gao, S. Uno, T. Tanaka, M. Akiyama, and N. Yamamoto, "Analysis and synthesis of feature map for kernel-based quantum classifier," arXiv preprint, 2019. [Online]. Available: https://arxiv.org/abs/1906.10467