Quantum Machine Intelligence (2026) 8:27
https://doi.org/10.1007/s42484-026-00379-4

RESEARCH ARTICLE

Benchmarking quantum machine learning methods for intrusion detection on noisy quantum computers

Franco Cirillo[1] · **Christian Esposito**[1] · **Jung Taek Seo**[2]

Received: 4 November 2025 / Accepted: 22 February 2026

Abstract
Intrusion detection systems (IDS) are essential for identifying cyber threats in complex digital environments. Machine learning (ML) is widely used to improve IDS by detecting anomalies, but classical ML methods often struggle with high-dimensional data and evolving threats. Quantum machine learning (QML) has been proposed as a potential paradigm to overcome some of these limitations, but is constrained by noisy intermediate-scale quantum (NISQ) challenges, affecting quality. This study systematically evaluates three QML models, Pegasos Quantum Support Vector Classifier (QSVC), Variational Quantum Classifier (VQC), and a Hybrid Quantum–Classical Neural Network (HQNN), for network anomaly detection. The models were optimized and tested on the ToN_IoT and NSL-KDD datasets using IBM quantum simulators under both ideal and noisy conditions. Performance was analyzed through the F1-score distribution as a function of circuit complexity, revealing how entanglement and noise affect robustness across different backends. Comparisons with classical models contextualize the current maturity of QML for cybersecurity, while computational time was used as an indicator of model complexity to explore accuracy–efficiency trade-offs. Among all configurations, Pegasos-QSVC achieved the best results, with 94.60% accuracy and an F1-score of 94.13%. The findings provide practical guidelines for designing noise-resilient QML models and highlight their potential for reliable intrusion detection under realistic quantum conditions.

Keywords Cybersecurity · Hybrid quantum-classical neural networks (HQNN) · QSVM · Quantum noise · Quantum noisy simulators · VQC

1 Introduction

Machine Learning (ML) has proven to be highly effective for Intrusion Detection Systems (IDS) (Thakkar and Lohiya 2023; Muneer et al. 2024) due to its ability to analyze large volumes of data, detect patterns, and identify both known and unknown threats with high accuracy. By leveraging advanced algorithms, ML can adapt to evolving attack vectors and provide real-time threat detection, making it an essential tool in modern cybersecurity. However, classical ML still faces challenges in handling high-dimensional data, adapting to sophisticated attack strategies, and maintaining efficiency as datasets grow in complexity (Yamasaki et al. 2023; Anschuetz et al. 2024).

Quantum Machine Learning (QML), on the other hand, has been proposed as a potential paradigm to overcome some of these limitations (Peral-García et al. 2024). While QML theoretically could offer advantages through quantum parallelism and richer data representations (Cerezo et al. 2022), these benefits remain unproven in current hardware and practical applications. This work therefore does not aim to demonstrate quantum advantage but focuses on systematic characterization of QML behavior under realistic noise, establishing design principles for noise-resilient circuits.

Currently, the most prominent QML models (Misra and Rani 2024) include Quantum Support Vector Machines (QSVM), Variational Quantum Classifiers (VQC), and hybrid quantum-classical neural networks. These

✉ Franco Cirillo
fracirillo@unisa.it

Christian Esposito
esposito@unisa.it

Jung Taek Seo
seojt@gachon.ac.kr

[1] Computer Science, University of Salerno, Fisciano, Salerno, Italy

[2] Department of Computer Engineering, Gachon University, Seongnam-si, Republic of Korea

Published online: 11 March 2026

models leverage the unique properties of quantum systems to enhance the efficiency and accuracy of traditional machine learning algorithms, making them ideal candidates for cybersecurity applications. The TON_IoT (UNSW Sydney 2024) dataset is a next-generation dataset designed for Industry 4.0, IoT, and IIoT environments. Together with NSL-KDD dataset (Tavallaee et al. 2009), they are tailored for evaluating AI-based cybersecurity applications.

Most existing works test their QML models on simple simulators that do not account for the impact of noise (Bhattacharya et al. 2024; Gautam et al. 2024). They fail to reflect the challenges posed by real-world quantum hardware in the Noisy Intermediate-Scale Quantum (NISQ) era, such as decoherence, qubit errors, and gate imperfections (Cheng et al. 2023). As a result, the performance of QML models evaluated under ideal conditions may not accurately represent their effectiveness on noisy quantum devices (Das and Chakrabarti 2024). Moreover, while some studies have attempted to implement QML for intrusion detection on real quantum hardware (Kukliansky et al. 2024), their performance remains limited due to these noise-related challenges. Additionally, testing on actual quantum hardware, while necessary to validate theoretical models, is costly. A practical solution is the use of noisy simulators (Cirillo and Esposito 2025c), such as the ones introduced by IBM (Ibm Quantum 2021). These simulators emulate the behavior of real quantum hardware under noisy conditions, allowing researchers to test and refine their QML models without the high costs associated with physical quantum computers.

The main objective of this work is to systematically assess how different design choices affect the performance and noise resilience of QML models for intrusion detection. While previous studies have typically focused on limited configurations or ideal conditions, our work provides a comprehensive evaluation across multiple architectures and noise settings.

This work makes the following key contributions:

- Comprehensive benchmarking of QML models. We analyze three representative architectures (Pegasos-QSVC, Variational Quantum Classifier, and hybrid quantum–classical neural network) across a broad range of configurations in both ideal and noisy environments, identifying how design and parameter choices affect resilience and generalization.
- Quantitative and qualitative evaluation of noise effects. By examining the F1-score distribution as a function of circuit complexity (measured through CNOT count) and testing on multiple IBM noisy simulators, we identify the optimal complexity ranges for noise robustness, characterize instability regions, and derive insights on how entanglement impacts performance across different backends.
- Performance contextualization and practical guidelines. We compare quantum and classical models trained under identical settings to assess the current maturity of QML for intrusion detection, use simulator execution time as an alternative for model complexity to study accuracy–efficiency trade-offs, and propose a set of quantitative and qualitative guidelines on circuit design, parameter tuning, and backend selection to optimize QML performance under realistic conditions.

This work is organized into the following sections: Section 2 provides the background, explaining the QML models used, introducing the TON_IoT and NSL-KDD datasets, and Section 3 reviews the related work. Section 4 details the methodology, including data pre-processing and the use of quantum computing tools and noisy simulators. Section 5 presents the optimization and performance analysis of the Pegasos-QSVC, VQC, and hybrid neural network models, while Section 6 provides a time analysis of the models tested with the different configurations. Section 7 provides a comparison of the results with the classical ML models. Finally, Section 8 summarizes the findings and discusses the impact of noise, and Section 9 concludes and outlines future directions to improve QML models in real-world cybersecurity.

2 Background

2.1 Quantum machine learning

Quantum Machine Learning (QML) combines quantum computing principles with machine learning to potentially improve computational efficiency (Raubitzek and Mallinger 2023. In the context of intrusion detection, QML can provide expressive models capable of capturing complex patterns in network traffic.

Quantum Support Vector Machines (QSVM) is the quantum counterpart of classical SVMs, leveraging quantum feature spaces to classification (Akrom 2024). Classical data are mapped into a quantum Hilbert space via feature maps, and similarity is computed using a quantum kernel. The choice of feature map and kernel directly affects the model's ability to distinguish between normal and anomalous traffic. Pegasos-QSVC, a stochastic gradient-based variant of QSVM (Shalev-Shwartz et al. 2007), reduces computational cost and is particularly suitable for large intrusion detection datasets.

Variational Quantum Classifiers (VQC) extend classical neural networks to the quantum domain using parameterized quantum circuits trained with classical optimizers

(Maheshwari et al. 2021). The circuit structure (ansatz) is critical, as it determines expressivity and noise resilience. Common ansatzes include Two-Local, Pauli Two-Design, Real Amplitudes, and EfficientSU2 (Dao 2025). Optimizers, both gradient-based (e.g., Adam Kingma 2014, NFT Nakanishi et al. 2020) and gradient-free (e.g., COBYLA Powell 1994), are used to minimize cost functions derived from misclassification, balancing accuracy and computational efficiency. The choice of ansatz and optimizer impacts training stability, especially under realistic noisy conditions.

Hybrid networks integrate classical layers with parameterized quantum circuits as hidden layers (Killoran et al. 2019). Classical layers preprocess input features, while the quantum circuit introduces a highly expressive representation of the data. Measurement results are fed back into classical layers for final classification. This hybrid approach combines classical scalability with quantum representational power, which could be advantageous for detecting subtle anomalies in network traffic.

Overall, the selection of QML algorithms, circuit ansatzes, and optimizers is driven by their proven effectiveness and wide adoption in the state of the art for anomaly detection. QSVM, VQC, and hybrid classical–quantum models represent the most established and experimentally validated quantum learning paradigms, offering a balance between expressive power, robustness to noise, and computational efficiency.

2.2 Dataset

The first dataset used in this research is the "TON_IoT Datasets" (UNSW Sydney 2024), a next-generation dataset collection for Industry 4.0, IoT, and Industrial IoT (IIoT), designed to evaluate the reliability and effectiveness of various artificial intelligence-based cybersecurity applications, and has been used in several works (Gad et al. 2021; Guo et al. 2023; Cao et al. 2024). The "Train_Test_datasets" part consists of four types of datasets: telemetry data from IoT and IIoT sensors, data from Windows 7 and 10 operating systems, Ubuntu 14 and 18 TLS, and network traffic data. The dataset includes a total of 211,043 records in nine categories of cyberattacks as described in Table 1.

NSL-KDD is the second intrusion detection dataset used in this work, and it was introduced by Tavallaee et al. (2009) as a means to overcome the shortcomings of KDD-99, particularly to enhance the reliability of network intrusion detection system evaluations.

The dataset also features a more balanced record selection, where the number of records from each difficulty level is inversely proportional to their representation in the original KDD dataset. Additionally, NSL-KDD is smaller in size compared to KDD-99, consisting of fewer but unique records, which reduces computational costs and enhances efficiency for training machine learning models.

Table 1 Distribution of attacks in the "TON_IoT Datasets"

Type	Number of rows
Normal	50,000
Backdoor	20,000
DDoS	20,000
DoS	20,000
Injection	20,000
Password	20,000
Ransomware	20,000
Scanning	20,000
XSS	20,000
MITM	1,043

The training set contains 125,972 records, while the test set contains 22,542 records. Each set is categorized by attack types, with the distribution described in Table 2.

The NSL-KDD dataset includes 41 features that represent various aspects of network connections, such as basic connection details, content features, and traffic characteristics. These features are categorized as continuous or categorical, and they provide rich data for training machine learning models to detect both normal and malicious network traffic.

These datasets are chosen to jointly evaluate the proposed approach in both realistic, heterogeneous IoT/IIoT environments and well-established benchmark settings for intrusion detection, enabling an assessment of robustness, comparability with prior work, and generalization across different traffic characteristics and attack distributions.

3 Related work

The application of QML for cybersecurity and specifically for intrusion detection continues to advance. We provide a review of relevant studies, showcasing different quantum techniques, models, and their experimental results on various datasets.

A 20-qubit QNN was implemented in Bhattacharya et al. (2024), using three hidden layers for intrusion detection on the CSE-CICIDS2018 dataset, reporting a 95% accuracy and 97% F1 score. The use of a noiseless setup

Table 2 NSL-KDD train and test data distribution

Class	Training Set	Percentage (Training)	Test Set	Percentage (Test)
Normal	67,342	53.46%	9,710	43.08%
DoS (Denial of Service)	45,927	36.46%	7,457	33.08%
Probe	11,656	9.25%	2,421	10.74%
R2L (Remote to Local)	995	0.79%	2,754	12.22%
U2R (User to Root)	52	0.04%	200	0.89%
Total	125,972	100%	22,542	100%

limited insights on noise resilience. A different study (Kukliansky et al. 2024) developed a QNN using single qubits per feature with rotational encoding on the x-axis, utilizing 8 qubits tested on IonQ's quantum computers. Optimized with SGD, this approach achieved an F1 score of 0.86 on the NF-UNSW-NB15 dataset. The performance benefits were attributed to the strategic selection of native gates in an all-to-all quantum setup, ensuring minimal performance degradation attributable to noise.

An hybrid QNN architecture has been introduced in Gautam et al. (2024), combining quantum layers with Dense classical layers for KDD99 and CICIDS-2017 datasets, achieving high accuracies of 99.81% and 98.74% not considering the noise. An evaluation of Botnet DGA classification has been proposed in Suryotrisongko and Musashi (2022), using noise models derived from IBM quantum devices. They leveraged four features from their Botnet DGA dataset and implemented a hybrid quantum-classical model. Their results demonstrated accuracy reaching up to 94.7% with $n = 100$ samples, 93.9% with $n = 1,000$ samples, and 89.7% with $n = 10,000$ samples, showing a performance degradation with an increasing number of samples. Another hybrid model was presented in Liu et al. (2024), consisting of a variational adversarial encoder and fuzzy Gaussian quantile neural network for UNSW-NB15, achieving 95% accuracy and a lower F1 score of 72%.

The CIC-DDoS 2019 dataset has been used in Küçükkara et al. (2024), developing a QNN-based intrusion detection system using with COBYLA optimization on IBM's simulator. The model achieved an accuracy of 92.63% and F1 score of 92.11% on statevector simulation, while local computation accuracy reached 80.69%, indicating a significant impact of noise.

A quantum federated learning approach was implemented in Houda et al. (2024) using QNN with AngleEmbedding for NSL-KDD, reporting a 98% accuracy with 5 to 15 clients. The use of multiple clients enhanced performance, although noise impact was not considered.

A variational QNN-based intrusion detection was proposed in Gong et al. (2022), tested on KDD Cup 99 reached an accuracy of 97.81% and an F1 score of 98.35% on noiseless IBM hardware. Noise reduced F1 to 83.87%, highlighting the importance of noise mitigation for quantum-enhanced models. On the same dataset, this study (Venkatachalam and Liu 2023) implemented QSVM and VQC with minimal features, reporting accuracies between 60-64% and an F1 score of 45%. Limited feature representation suggests scope for optimization in future quantum approaches.

Rahman et al. (2024) employed EfficientSU2 and COBYLA optimizer in a Variational Quantum Classification (VQC) model on NSL-KDD, resulting in 90% accuracy in IBM quantum simulators. Further research by Rahman et al. (2023) introduced a qGAN on NSL-KDD, though no specific metrics or code were provided. Other generative models for anomaly detection were presented in Tezuka et al. (2024), structured into three stages: probability distribution construction, anomaly score calculation, and threshold setting. While effective, the study lacked specific performance metrics, such as accuracy.

The work (Gouveia and Correia 2020) focused on dimensionality reduction, demonstrating that quantum autoencoders outperform PCA on the UNB NSL-KDD and UNSW NB15 datasets. With 150 training examples, the model achieved accuracies of 0.75 and 0.93, respectively. Hdaib et al. (2024) integrated quantum autoencoders with quantum one-class SVM, quantum random forest, and quantum k-nearest neighbors. For KDD99, IoT-23, and CIC-IoT23 datasets, F1 scores of 97%, 96%, and 98% were achieved. Noise effects were not specified, leaving a general uncertain.

The study (Said 2023) utilized a Quantum Support Vector Machine (QSVM) for detecting DDoS attacks on smart micro-grids. Exploiting a sample of 38 features and 2950 data points selected as dataset, authors report an accuracy of 99.94% which, while noteworthy, appears unusually high given the typical challenges in this domain. This Quantum model can be the result of overfitting due to their capacity to capture intricate patterns in the training data, especially when using many features. Without sufficient evidence of rigorous validation on independent datasets, it remains unclear whether the reported performance genuinely reflects the model's ability to generalize beyond the training set.

In the preprint (Abreu et al. 2024), Abreu et al. evaluated QML models, including VQC, QSVM, QKM, and QCNN, on datasets UNSW-NB15, CIC-IDS17, CICIoT23, and TON_IoT. Personalized circuit optimization contributed to improved results, with VQC and QSVM models achieving a 97% F1 score on ToN_IoT. QCNN emerged as the most effective model across scenarios. However, this work does not delve into the use of an increasing number of qubits and is limited to examining only 4 feature maps. Therefore, it has not yet undergone peer review, requiring further validation.

QSVM and QCNN for intrusion detection was explored in Kalinin and Krundyshev (2022), achieving 98% accuracy on custom data streams without a noise model. The absence of a detailed setup on quantum simulation limits reproducibility. The convolutional approach was also explored in Gong et al. (2024), toghether with a Variational Quantum Neural Network (VQNN) on UNSW-NB15 dataset. In noiseless conditions, the model reached a 95.86% F1 score, dropping to 80.62% under noisy conditions.

Table 3 summarizes recent quantum and hybrid quantum-classical models applied to intrusion and anomaly detection, reporting their architectures, datasets, performance metrics, and treatment of noise. From the table, several trends

Table 3 Comparison of quantum machine learning models for intrusion detection

Ref.	Model / Approach	Dataset(s)	Performance Metrics	Noise Consideration
Bhattacharya et al. (2024)	20-qubit QNN with 3 hidden layers	CSE-CICIDS2018	Accuracy: 95%, F1 Score: 97%	Not considered
Kukliansky et al. (2024)	QNN with rotational encoding on 8 qubits	NF-UNSW-NB15	F1 Score: 0.86	IonQ's quantum computers, native gate optimization
Gautam et al. (2024)	Hybrid QNN with Dense classical layers	KDD99, CICIDS-2017	Accuracy: 99.81% (KDD99), 98.74% (CICIDS-2017)	Not considered
Suryotrisongko and Musashi (2022)	Hybrid quantum-classical model	Botnet DGA dataset	Accuracy: 89.7% ($n = 10,000$)	Noise models derived from IBM quantum devices
Liu et al. (2024)	Hybrid model with variational adversarial encoder	UNSW-NB15	Accuracy: 95%, F1 Score: 72%	Not considered
Küçükkara et al. (2024)	QNN with COBYLA optimization	CIC-DDoS 2019	Accuracy: 92.63% (statevector), 80.69% (noisy computation)	Significant impact of noise
Houda et al. (2024)	Quantum Federated Learning (QNN with AngleEmbedding)	NSL-KDD	Accuracy: 98%	Not considered
Gong et al. (2022)	Variational QNN for intrusion detection	KDD Cup 99	F1 Score: 98.35% (noiseless), 83.87% (noisy)	Significant impact of noise
Venkatachalam and Liu (2023)	QSVM and VQC with minimal features	KDD Cup 99	Accuracy: 60-64%, F1 Score: 45%	Not specified
Rahman et al. (2024)	VQC with EfficientSU2 and COBYLA optimizer	NSL-KDD	Accuracy: 90%	Not specified
Rahman et al. (2023)	qGAN model	NSL-KDD	No specific metrics provided	Not specified
Tezuka et al. (2024)	Generative model for anomaly detection	Unspecified	No performance metrics provided	Not specified
Gouveia and Correia (2020)	Quantum autoencoder for dimensionality reduction	UNB NSL-KDD, UNSW NB15	Accuracy: 0.75 (NSL-KDD), 0.93 (UNSW NB15)	Not specified
Hdaib et al. (2024)	Quantum autoencoder with quantum one-class SVM	KDD99, IoT-23, CIC-IoT23	F1 Score: 97%, 96%, 98%	Not specified
Said (2023)	QSVM for DDoS detection on smart micro-grids	Custom	Accuracy: 99.94%	Not considered
Kalinin and Krundyshev (2022)	QSVM, QCNN for intrusion detection	Custom data streams	Accuracy: 98%	No noise model specified
Gong et al. (2024)	VQNN with convolutional approach	UNSW-NB15	F1 Score: 95.86% (noiseless), 80.62% (noisy)	Significant impact of noise

emerge: most models achieve high accuracy or F1 scores on standard datasets, particularly KDD99, NSL-KDD, and UNSW-NB15. Hybrid quantum-classical models generally report slightly higher performance than pure quantum models. However, noise resilience is often limited or not considered; only a few studies explicitly evaluate noisy execution, and these show a significant drop in performance under realistic noise conditions. This summary makes it easier to identify both effective modeling strategies and gaps in the literature, especially the need for more robust noise modeling and evaluation in practical quantum machine learning applications for cybersecurity.

The focus of this work was not placed on reviewing articles that employ classical machine learning methods, as the primary objective of this work was not to achieve a quantum advantage. Rather, the goal was to provide insights into the methods and their configurations to enhance model performance when operating under quantum noise. Moreover, existing literature such as Jayalaxmi et al. (2022) already reports 100% accuracy on the same dataset with classical ML models.

4 Methodology

This section outlines the methodology adopted to prepare the datasets, configure and optimize quantum machine learning models, and evaluate their performance under realistic quantum conditions.

Both the ToN_IoT and NSL-KDD datasets underwent a standardized preprocessing pipeline to ensure data quality and suitability for classification. After cleaning and encoding categorical features, the data were split into stratified training (80%) and testing (20%) sets. To reduce computational cost while preserving representativeness, a balanced 15% sample per class was used, resulting in 11,270 training and 2,818 testing samples for ToN_IoT, and 15,114 training and 3,780 testing instances for NSL-KDD. Feature values were normalized within and reduced via Principal Component Analysis (PCA), with a number based on the qubit count of the each configuration.

Following preprocessing, quantum classification models were configured, trained, and optimized under both ideal and noisy simulation conditions to assess their robustness and practical applicability.

4.1 QML models configuration

In this work, we have implemented and evaluated three QML models. As our QSVM model, we used the Pegasos-QSVC algorithm. The methodology for hyperparameter optimization is summarized in Fig. 1. Following the preprocessing stage, we tested different values of the regularization parameter to identify the optimal configuration. Subsequently, various feature maps, numbers of qubits, and repetitions (r) were explored to assess their impact on the model's performance. The best-performing configurations were then evaluated on noisy quantum simulators to validate their robustness, yielding the metrics detailed in Section 2.

For the VQC model, different ansatz designs and optimizers were tested to optimize the performance of the classifier. The methodology applied to this process is outlined in Fig. 2. These evaluations allowed us to identify configurations that effectively balance quantum resources with model accuracy.

The structure of the hybrid quantum-classical neural network is presented in Fig. 3. This architecture integrates classical layers with a quantum layer, leveraging the strengths of both paradigms. The quantum circuit used in the model consists of a feature map and an ansatz. The classical component was developed using fully connected layers, which were employed to project data from one dimension to another. Each linear layer is followed by a non-linear activation function, such as ReLU, which introduces non-linearity into the model, enabling the network to learn complex representations of the data. Specifically, the input data first passes through an initial linear layer, is transformed by the ReLU activation function, and then passes through a second linear layer. Subsequently, the output of the second layer can serve as the input to the quantum circuit. The measurement results of the circuit are then used as the final predictions, indicating either 0 or 1 based on the measurement outcomes.

All the circuit components, such as the feature map and ansatz, were modeled using the `qiskit` package (version `1.1`). The QML models were implemented with `qiskit-machine-learning` (version `0.7`), and the optimizers used for VQC were sourced from the `qiskit_algorithms` package. For the implementation of the hybrid quantum-classical model, the `Torch` library was employed. Further details on these model and their configurations are described in Section 5.

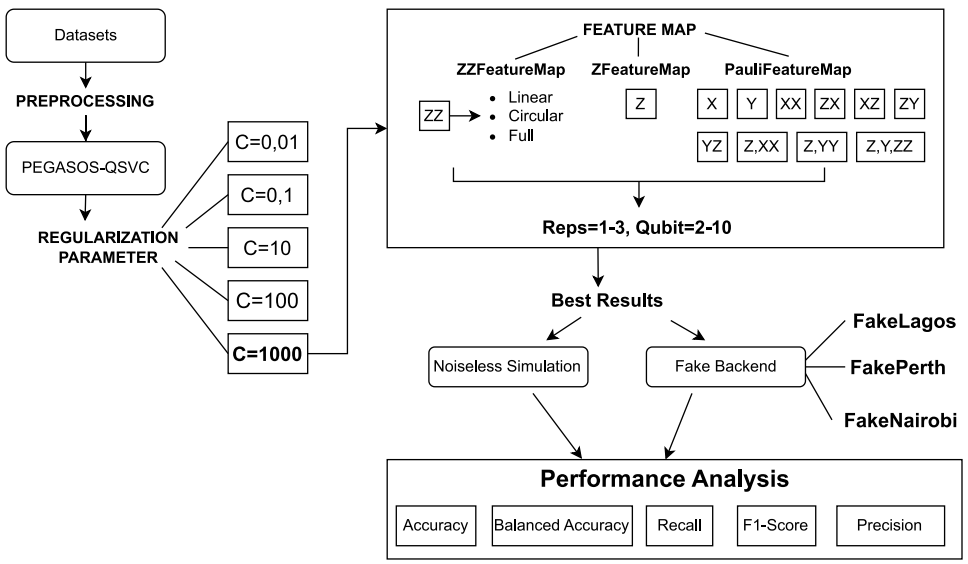

Fig. 1 QSVC configuration optimization methodology

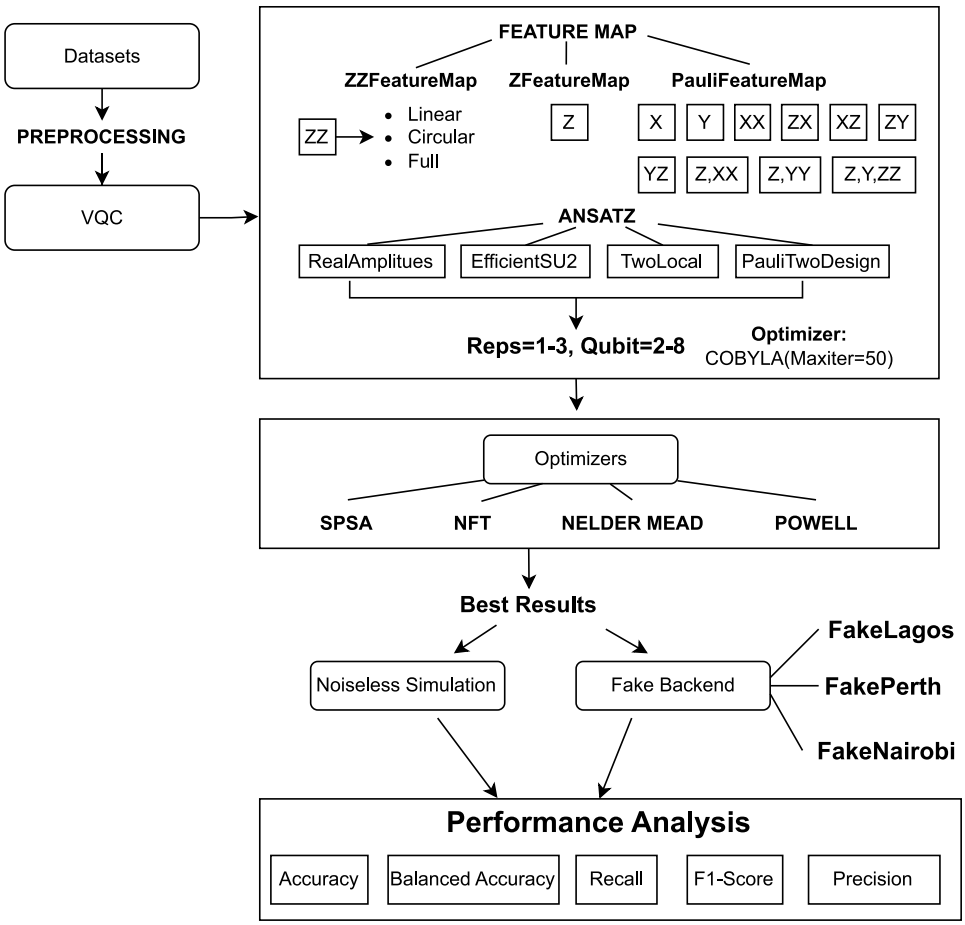

Fig. 2 VQC configuration optimization methodology

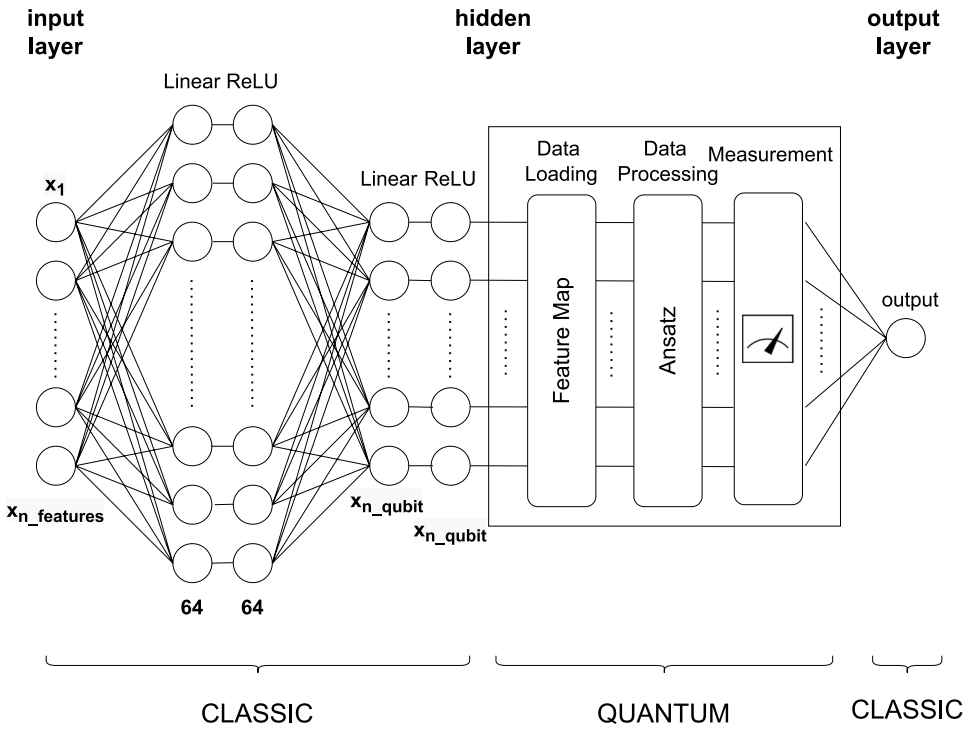

Fig. 3 Hybrid quantum-classical neural network structure

To ensure a statistically robust evaluation among all the models, we conducted multiple runs using different 4 random seeds for each execution and reported the mean performance and the corresponding standard deviation. This approach accounts for the sensitivity of QML models to seeding choices and provides a more reliable assessment of their performance.

4.2 Noisy quantum computation

When analyzing optimal combinations of quantum circuits, it is essential to go beyond ideal scenarios, such as noiseless simulations. Real quantum devices are subject to imperfections caused by various types of errors, including qubit decoherence, gate errors, and environmental noise, which can compromise computational reliability. Several key metrics are critical for evaluating the performance and reliability of quantum systems, including T1, T2, readout error, and gate errors such as rz, sx, x, and cx errors. Each metric reflects a different aspect of the challenges faced in maintaining and manipulating delicate quantum states.

T1, the relaxation time, measures how long a qubit can stay in an excited state before decaying to its ground state. A longer T1 is desirable for extended computation but is limited by environmental noise and material properties. Similarly, T2, the dephasing time, indicates how long a qubit maintains coherence between quantum states, critical for superposition. T2 is often shorter than T1, as it includes both energy relaxation and phase noise. Readout error reflects inaccuracies in measuring qubit states. Even if operations are performed perfectly, high readout error can distort results. Gate errors occur during qubit operations, with rz gates typically having minimal errors due to their virtual implementation, while single-qubit gates like sx and x face challenges from control inaccuracies. The cx (CNOT) gate, essential for entanglement, often exhibits the highest error rates due to its complexity and sensitivity to noise.

The main challenges of testing the proposed quantum ML methods on actual quantum hardware are the high computational cost and the long queue times required for execution. Evaluating all tested configurations is particularly demanding, with several configurations, especially when dealing with deep circuits and large sample sizes. Given these constraints, we opted for noisy simulators, which allowed us to systematically assess model performance under realistic noise conditions while maintaining experimental feasibility.

The selected quantum circuits were also tested under noisy conditions using *fake backends* provided by IBM. *Fake backends* are designed to mimic the behavior of IBM Quantum systems and are built using system snapshots. These snapshots contain information about the simulated quantum device, such as the coupling map, which describes the physical connections between qubits, and the qubit properties. The simulator was built using the `qiskit-aer` package with the Sampler primitive, and all simulations were performed with 1024 shots.

Since the maximum number of qubits used in the selected combinations is 7, fake backends with 7 qubits were considered. Among the available fake backends, three were selected: `FakeLagosV2`, `FakeNairobiV2`, and `FakePerth`. While these backends share the same coupling map Fig. 4, they differ in their associated error levels. The coupling map represents the physical connectivity between qubits, indicating which pairs of qubits can interact directly, typically for two-qubit gates like CNOT. This map is crucial because not all qubits in a quantum processor are directly connected. If an operation requires interaction between non-connected qubits, additional gates must be introduced to route the information, increasing the circuit depth and error probability.

This backend selection aimed to explore various noise conditions, as each fake backend introduces different types and intensities of errors. Specifically, FakeLagosV2, as shown in Tables 4 and 5, was chosen for its good single-gate error probability but high two-qubit gate error and readout error, whereas FakeNairobiV2, Tables 6 and 7, and FakePerth, Tables 8 and 9, are characterized by higher gate error levels but lower readout error and two-qubit gate error. Data represented in these tables has been collected from the latest snapshot of 2024-05-27.

Noisy simulation obviously produces different results compared to the ideal one. However, starting with noise-free simulations represents a useful approach, as adding

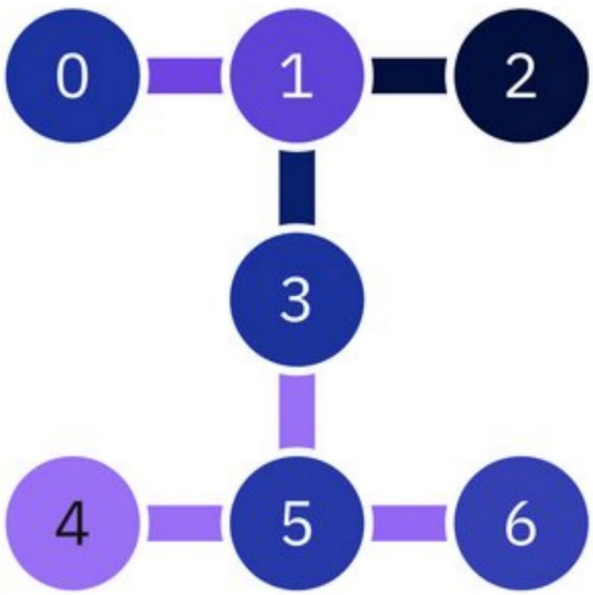

Fig. 4 Coupling map of the fake backends used

Table 4 T1, T2, and readout error values for each qubit in FakeLagos

Qubit	T1 (μs)	T2 (μs)	Readout Error
1	146.27	33.30	0.1690
2	92.22	93.08	0.1362
3	105.44	83.08	0.4638
4	120.91	57.86	0.0167
5	100.22	24.44	0.0292
6	88.07	72.40	0.2619
7	181.83	125.43	0.3480

Table 5 RX, SX, X, CX gate errors for each qubit or couple of qubits in FakeLagos

Qubits	Gate	Name	Gate Error
0-6	rz	rz0-6	0
0	sx	sx0	0.00016
1	sx	sx1	0.00025
2	sx	sx2	0.00029
3	sx	sx3	0.00029
4	sx	sx4	0.00025
5	sx	sx5	0.00030
6	sx	sx6	0.00021
0	x	x0	0.00016
1	x	x1	0.00025
2	x	x2	0.00029
3	x	x3	0.00029
4	x	x4	0.00025
5	x	x5	0.00030
6	x	x6	0.00021
5, 6	cx	cx5_6	0.0202
5, 4	cx	cx5_4	0.0083
3, 1	cx	cx3_1	0.0107
3, 5	cx	cx3_5	0.0290
2, 1	cx	cx2_1	0.0103
0, 1	cx	cx0_1	0.0094

Table 6 T1, T2, and readout error values for each qubit in FakeNairobi

Qubit	T1 (μs)	T2 (μs)	Readout Error
1	89.12	15.79	0.0580
2	87.27	126.40	0.0199
3	133.24	127.14	0.0193
4	74.94	76.00	0.0223
5	131.22	106.13	0.0183
6	80.65	12.04	0.0225
7	76.64	126.21	0.0258

noise is computationally expensive. Ideal simulations thus serve as an initial baseline from which to explore various aspects further.

5 Optimization and results

In this section, we outline the search parameters used for the optimization process and describe the steps taken to fine-tune the models. For each of the three models, the results of the best evaluations are presented, both under noiseless and noisy simulation conditions. This analysis highlights the performance differences and robustness of the models across various simulation environments. As summarized in Table 10, the main hyperparameter configurations adopted for each model are reported, including the general settings used throughout all experiments. Further details and motivations for the selected configurations are provided in the dedicated subsections describing each model evaluation.

Table 7 RX, SX, X, CX gate errors for each qubit or couple of qubits in FakeNairobi

Qubits	Gate	Name	Gate Error
[0-6]	rz	rz0-6	0
[0]	sx	sx0	0.0004
[1]	sx	sx1	0.0003
[2]	sx	sx2	0.0002
[3]	sx	sx3	0.0004
[4]	sx	sx4	0.0005
[5]	sx	sx5	0.0008
[6]	sx	sx6	0.0003
[0]	x	x0	0.0004
[1]	x	x1	0.0003
[2]	x	x2	0.0002
[3]	x	x3	0.0004
[4]	x	x4	0.0005
[5]	x	x5	0.0008
[6]	x	x6	0.0003
[6, 5]	cx	cx6_5	0.0107
[5, 4]	cx	cx5_4	0.0070
[5, 3]	cx	cx5_3	0.0126
[1, 3]	cx	cx1_3	0.0068
[2, 1]	cx	cx2_1	0.0070
[0, 1]	cx	cx0_1	0.0086

Table 8 T1, T2, and Readout Error values for each qubit in FakePerth

Qubit	T1 (μs)	T2 (μs)	Readout Error
1	55.93	95.07	0.0287
2	123.02	49.93	0.0254
3	195.38	57.13	0.0326
4	160.72	271.22	0.0290
5	51.43	56.56	0.0308
6	93.48	123.55	0.0431
7	154.41	213.50	0.0195

5.1 Pegasos-QSVC

The Pegasos-QSVC model has been used for this research instead of QSVC because it allows achieving good results in relatively short times, as it is independent of the size of the training set.

Table 9 RX, SX, X, CX gate errors for each qubit or couple of qubits in FakePerth

Qubits	Gate	Name	Gate Error
0-6	rz	rz0-6	0
0	sx	sx0	0.00024
1	sx	sx1	0.00037
2	sx	sx2	0.00040
3	sx	sx3	0.00033
4	sx	sx4	0.00043
5	sx	sx5	0.00053
6	sx	sx6	0.00040
0	x	x0	0.00024
1	x	x1	0.00037
2	x	x2	0.00040
3	x	x3	0.00033
4	x	x4	0.00043
5	x	x5	0.00053
6	x	x6	0.00040
6, 5	cx	cx6_5	0.0130
4, 5	cx	cx4_5	0.0170
3, 5	cx	cx3_5	0.0086
3, 1	cx	cx3_1	0.0048
2, 1	cx	cx2_1	0.0079
0, 1	cx	cx0_1	0.0069

5.1.1 Regularization parameter *C*

An initial element explored is the regularization parameter C of the PegasosQSVC function, which controls the trade-off between maximizing the margin and minimizing classification errors in the SGD update rule. As discussed in Bhavsar et al. (2023), a higher C reduces the regularization effect, penalizing misclassified points more strongly and thus fitting the training data more closely, which can improve accuracy but also increase the risk of overfitting. Conversely, a lower C increases regularization, allowing a softer margin and potentially improving generalization by tolerating some misclassifications.

The performance of the Pegasos-QSVC algorithm was analyzed for different C values, as shown in Table 11. Empirically, higher$C$ values (1000 and 100) yielded the best performance, with metrics exceeding 91% and a balanced F1-Score. This suggests that, for this specific intrusion detection task, stronger fitting to the data provided by higher C values was beneficial. Therefore,$C = 1000$ was selected as the optimal parameter.

In addition to setting the parameter C, it is also necessary to specify the number of steps τ to execute during the training procedure. An analysis was conducted to determine the optimal number of steps by testing values of 20, 50, 100, 200, and 500. The result that balances execution time and accuracy was achieved with 100 steps.

Table 10 Hyperparameter configurations for reproducibility

General Settings	
Dataset split (Train/test)	80–20
Seeds	12345–12348
QSVC	
Algorithm	PegasosQSVC
PCA component count	2–10
C	0.01–1000
N_{steps}	20, 50, 100, 200, 500
VQC	
PCA component count	2–10
Max iterations	200
HQNN	
Epochs	20
Optimizer	Adam
Learning rate	0.001
Batch size	64

5.1.2 Feature map

The encoding of classical data into quantum data using angle encoding employs rotations of quantum gates. In Qiskit, the PauliFeatureMap, ZFeatureMap, and ZZFeatureMap are predefined circuits for this mapping. The PauliFeatureMap (based on the expansion of Pauli operators) is the most general, using arbitrary combinations of Pauli operators (X, Y, Z); the ZFeatureMap (a first-order circuit with evolution around Z) applies rotations along the Z -axis; and the ZZFeatureMap (a second-order circuit with ZZ evolution) applies two rotations around the Z-axis for each qubit, creating interactions between them. While all three feature maps are analyzed in the literature, the PauliFeatureMap is often studied with a single combination of rotations. In this work, the following aspects were explored. For the ZZFeatureMap, three levels of entanglement were considered:

- Full Entanglement: Each qubit is entangled with all other qubits.
- Linear Entanglement: Each qubit is entangled only with its adjacent qubit.
- Circular Entanglement: Qubits are entangled in a closed loop, where the last qubit is entangled with the first.

Table 11 Pegasos-QSVC performances for several C values, the best one is highlighted

Pegasos-QSVC	Accuracy	Precision	Recall	F1-Score
C= 1000	**92,25%**	**92,77%**	**97,18%**	**94,92%**
C= 100	91,61%	92,02%	97,18%	94,53%
C= 10	79,16%	89,48%	81,65%	85,39%
C= 1	74,56%	74,56%	100%	85,42%
C= 0.1	74,56%	74,56%	100%	85,42%
C= 0.01	74,56%	74,56%	100%	85,42%

For the PauliFeatureMap, the following combinations of rotations were explored: "X", "Y", "XX", "XZ", "ZX", "YZ", "ZY", "Z,YY", "Z,XX", and "Z,Y,ZZ".

Another important element is the number of repetitions that refers to the number of times a subcircuit of encoding operations is repeated within the quantum circuit. Each repetition adds a layer of operations, increasing the circuit depth and enhancing its ability to capture complex patterns in the data. However, this can also introduce computational complexity and additional noise when executed on real hardware. Since the number of repetitions can significantly affect model performance, the feature maps were analyzed using configurations with 1, 2, and 3 repetitions. A variable number of qubits between 2 and 10 was analyzed to provide a comprehensive overview of the model's behavior as a function of the number of qubits.

5.1.3 Execution results

A total of 378 experimental configurations were evaluated on the ToN_IoT dataset, exploring variations in the number of qubits (ranging from 2 to 10), 14 different feature maps, and 1 to 3 repetitions per feature map. After identifying the most promising configurations, a subset of the best-performing tests was also conducted on the NSL-KDD dataset to assess their generalization across different datasets. To optimize the number of tests, experiments were first conducted on a noiseless simulator and then on a noisy simulator.

The Table 12 shows the configurations of the models which have obtained the best performances on the ToN_IoT dataset in terms of Accuracy, Precision, Recall, F1 Score and Balanced Accuracy. For each metric, we also report the corresponding standard deviation, computed across the runs with 4 seeds, to provide an estimate of the variability and reliability of the results.

Observing the results as a function of the number of qubits (q), it is immediately evident that the use of two or three qubits does not yield good results, except for some isolated cases. The model's accuracy improves with an increase in the number of qubits; however, beyond a certain threshold (around 6–7 qubits), the improvement tends to stabilize. This behavior indicates that the models no longer benefit from adding more qubits, suggesting that the system is reaching a saturation point.

Regarding the number of repetitions (r), it has been observed that increasing it tends to enhance the stability of the results. However, this does not lead to significantly better performance, while it increases the complexity of the circuit.

Regarding the FeatureMaps, the ZZFeatureMap with Full and Circular entanglement generally performs better

Table 12 Pegasos-QSVC configurations with best performances on ToN_IoT / NSL-KDD, the best one is highlighted

Best	Accuracy (±	Precision (±	Recall (± std)	F1 Score (±	Balanced
FeatureMap	std)	std)		std)	Accuracy (± std)
q=6, ZZFeatureMap	94.12% ± 0.93 /	96.10% ± 0.82 /	96.01% ± 0.71 /	96.05% ± 0.76 /	92.29% ± 0.89 /
(Linear), r=1	90.21% ± 0.75	88.63% ± 0.77	90.29% ± 0.85	89.43% ± 0.83	90.21% ± 0.78
q=6, PauliFeatureMap	94.16% ± 0.87 /	96.04% ± 0.75 /	96.13% ± 0.83 /	96.09% ± 0.78 /	92.26% ± 0.90 /
[ZX], r=1	87.57% ± 0.80	85.79% ± 0.84	87.51% ± 0.73	86.64% ± 0.88	87.56% ± 0.79
q=4, PauliFeatureMap	93.17% ± 1.01 /	93.91% ± 0.98 /	97.15% ± 1.02 /	95.50% ± 0.95 /	89.34% ± 0.99 /
[Z,YY], r=1	92.18% ± 0.99	94.68% ± 0.87	87.68% ± 0.97	91.04% ± 0.85	91.88% ± 0.89
q=7, ZFeatureMap	**95.44% ± 0.72/**	**96.93% ± 0.69/**	**96.96% ± 0.81/**	**96.94% ± 0.75/**	**93.98% ± 0.84/**
, r=2	**95.80% ± 0.76**	**96.68% ± 0.73**	**93.82% ± 0.85**	**95.23% ± 0.79**	**95.67% ± 0.77**
q=5, PauliFeatureMap	94.14% ± 0.88 /	96.22% ± 0.81 /	95.91% ± 0.77 /	96.06% ± 0.74 /	92.43% ± 0.79 /
[XX], r=2	88.03% ± 0.83	83.54% ± 0.91	92.37% ± 0.80	87.74% ± 0.88	88.30% ± 0.85
q=6, PauliFeatureMap	95.61% ± 0.71 /	97.11% ± 0.66 /	96.99% ± 0.78 /	97.05% ± 0.72 /	94.27% ± 0.83 /
[Z,XX], r=2	86.32% ± 0.87	84.49% ± 0.93	86.17% ± 0.88	85.34% ± 0.84	86.31% ± 0.90
q=4, PauliFeatureMap	94.57% ± 0.82 /	95.89% ± 0.78 /	96.86% ± 0.75 /	96.37% ± 0.81 /	92.35% ± 0.79 /
[Z,Y,ZZ], r=2	84.95% ± 0.91	81.78% ± 0.93	84.97% ± 0.86	83.85% ± 0.90	84.95% ± 0.88
q=5, ZZFeatureMap	94.35% ± 0.79 /	95.65% ± 0.83 /	96.83% ± 0.77 /	96.24% ± 0.75 /	91.96% ± 0.88 /
(Linear), r=3	87.34% ± 0.86	87.61% ± 0.82	84.37% ± 0.89	85.96% ± 0.84	87.16% ± 0.79

than the Linear configuration, particularly with one or two repetitions. However, with three repetitions, the Linear configuration surpasses the others, demonstrating improved effectiveness with increased repetitions. The "Z,YY" feature map consistently ranks among the top-performing configurations, showing robust effectiveness across varying numbers of qubits and repetitions. Feature maps without entanglement ("X", "Y", and "Z") are less stable compared to entangled configurations but can still achieve high performance in some cases.

On the NSL-KDD dataset, the performances are lower, not maintaining the generalization observed in ToN_IoT, except for the specific configuration of q=7, ZFeatureMap, and r=2, which achieved a similar accuracy of 95.80%. This result suggests that while some configurations can generalize well, most do not retain the same level of effectiveness across different datasets.

One potential reason for this behavior could be differences in the statistical distributions and feature representations between the two datasets. The ToN_IoT dataset is characterized by a diverse set of attack scenarios with a richer feature space, while NSL-KDD, despite being a widely used benchmark, contains different structural and temporal dependencies. These variations may challenge the ability of quantum models to maintain consistent performance across datasets.

Moreover, the lower performance on NSL-KDD suggests that the feature encoding strategy and the chosen quantum feature maps play a crucial role in determining generalization capabilities. The ZFeatureMap with q=7 and r=2 appears to strike an optimal balance between expressivity and robustness, allowing it to retain high accuracy even when transitioning to a different dataset. This further supports the idea that specific configurations may have a stronger capacity for generalization, while others may be overly tailored to the dataset they were trained on.

5.1.4 Noisy simulation

The QSVC models have been tested on these noisy simulators, Table 13. The results show a decline in performance when noise is introduced into the simulations. However, the ZFeatureMap shows good performance even in the presence of noise, maintaining high results across all three noisy models. For the other feature maps, the results vary depending on the type of Fake Backend used, highlighting how the errors specific to each backend impact performance differently. In particular, FakeLagos is more sensitive to feature maps with a higher number of CX gates because it experiences greater errors for this type of gate compared to the other two backends. For example, it performs well when using the ZFeatureMap, which does not involve entanglement between qubits, compared to cases with greater entanglement, such as the ZZFeatureMap, obtaining lower values in terms of accuracy with respect to the other simulator.

On the NSL-KDD dataset, the performances are lower as expected, failing to maintain the generalization observed in ToN_IoT. However, an exception is observed for the specific configuration with $q = 7$, ZFeatureMap, and $r = 2$, which is highlighted in the table. Even though some configurations achieve better values in specific cases, this particular setup proves to be the most stable and general across different conditions.

5.2 VQC

The second model analyzed is the VQC. This analysis considers the same 14 Feature Maps as in the Pegasos-QSVC model, combined with the four most commonly used parameterized circuits (ansatz) provided by Qiskit: *RealAmplitudes*, *EfficientSU2*, *TwoLocal*, and *PauliTwoDesign*. Therefore, the number of repetitions for the Feature Map was set to 1, while the repetitions for the variational circuit were analyzed for values of 1, 2, and 3 and a number of qubits between 2 and 8.

5.2.1 Ansatz and FeatureMap analysis

EfficientSU2 is the ansatz that performs best on average and achieves the highest maximum values across all repetition settings, with the average accuracy increasing from 75.50% to 79.05% as the repetitions increase from 1 to 3. RealAmplitudes is another ansatz that performs well, with a noticeable improvement in average performance from 73.02% (r=1) to 77.99% (r=3). TwoLocal shows lower performance compared to the other two ansatzes but remains consistent. PauliTwoDesign has the worst performance, even though it improves with the number of repetitions.

The average performance for each ansatz improves as the number of repetitions increases. This is evident in both average and maximum values, suggesting that increasing enhances the overall performance of the ansatzes. However, while performance increases with r for all ansatzes, the improvement is not uniform: EfficientSU2 and RealAmplitudes show steady growth, while TwoLocal and PauliTwoDesign exhibit more modest gains. The Fig. 5 summerizes the results on the ansatzes used for the repetitions applied.

After confirming that r=3 is the best of the tested configurations for the ansatz, we analyze the results obtained with this value and show the best configurations in Table 14.

The results show that *PauliFeatureMap [X]* and *PauliFeatureMap [XX]* cannot classify the data effectively, regardless of the feature map used or the number of qubits; for this reason it has not been put in the Table 14. Conversely,

Table 13 Pegasos-QSVC performance of best configurations on Fake Nairobi, Fake Lagos and Fake Perth with ToN_IoT / NSL-KDD, the best one is highlighted.

Configuration	Fake Nairobi	Fake Lagos	Fake Perth
q=6, ZZFeatureMap (Linear), r=1	Acc: 93.10% / 87.99% ± std: 0.82 / 0.77 F1-Score: 95.34% / 86.87% ± std: 0.74 / 0.81	Acc: 93.13% / 88.47% ± std: 0.79 / 0.82 F1-Score: 95.36% / 87.90% ± std: 0.77 / 0.85	Acc: 93.10% / 87.96% ± std: 0.81 / 0.78 F1-Score: 95.34% / 86.81% ± std: 0.76 / 0.83
q=6, PauliFeatureMap [ZX], r=1	Acc: 91.31% / 86.14% ± std: 0.86 / 0.88 F1-Score: 94.24% / 84.66% ± std: 0.79 / 0.91	Acc: 91.24% / 85.79% ± std: 0.83 / 0.85 F1-Score: 94.12% / 84.51% ± std: 0.77 / 0.87	Acc: 91.85% / 87.01% ± std: 0.88 / 0.82 F1-Score: 94.62% / 85.92% ± std: 0.80 / 0.86
q=4, PauliFeatureMap [Z,YY], r=1	Acc: 95.35% / 90.63% ± std: 0.78 / 0.83 F1-Score: 96.88% / 89.62% ± std: 0.70 / 0.84	Acc: 95.35% / 90.37% ± std: 0.79 / 0.82 F1-Score: 96.88% / 89.38% ± std: 0.72 / 0.83	Acc: 95.35% / 91.16% ± std: 0.77 / 0.79 F1-Score: 96.88% / 90.19% ± std: 0.69 / 0.81
q=7, ZFeatureMap , r=2	Acc: 93.24% / 94.60% ± std: 0.73 / 0.75 F1-Score: 95.54% / 94.13% ± std: 0.69 / 0.71	Acc: 93.24% / 94.66% ± std: 0.74 / 0.76 F1-Score: 95.54% / 94.19% ± std: 0.68 / 0.70	Acc: 94.31% / 94.74% ± std: 0.70 / 0.73 F1-Score: 96.21% / 94.28% ± std: 0.67 / 0.72
q=5, PauliFeatureMap [XX], r=2	Acc: 93.76% / 87.12% ± std: 0.81 / 0.85 F1-Score: 95.81% / 85.92% ± std: 0.76 / 0.88	Acc: 93.86% / 87.04% ± std: 0.82 / 0.84 F1-Score: 95.88% / 86.33% ± std: 0.78 / 0.87	Acc: 93.88% / 86.56% ± std: 0.79 / 0.86 F1-Score: 95.89% / 85.53% ± std: 0.74 / 0.88
q=6, PauliFeatureMap [Z,XX], r=2	Acc: 95.35% / 85.66% ± std: 0.74 / 0.87 F1-Score: 96.87% / 82.72% ± std: 0.70 / 0.92	Acc: 95.49% / 86.85% ± std: 0.73 / 0.86 F1-Score: 96.97% / 83.91% ± std: 0.68 / 0.91	Acc: 94.52% / 85.32% ± std: 0.76 / 0.84 F1-Score: 96.35% / 82.40% ± std: 0.71 / 0.89
q=4, PauliFeatureMap [Z,Y,ZZ], r=2	Acc: 95.35% / 83.84% ± std: 0.77 / 0.90 F1-Score: 96.87% / 83.45% ± std: 0.72 / 0.92	Acc: 95.51% / 84.52% ± std: 0.75 / 0.88 F1-Score: 96.98% / 83.04% ± std: 0.70 / 0.90	Acc: 94.42% / 83.44% ± std: 0.80 / 0.86 F1-Score: 96.22% / 81.37% ± std: 0.74 / 0.91
q=5, ZZFeatureMap (Linear), r=3	Acc: 94.31% / 85.93% ± std: 0.78 / 0.85 F1-Score: 96.21% / 86.10% ± std: 0.73 / 0.83	Acc: 92.11% / 86.01% ± std: 0.84 / 0.83 F1-Score: 94.66% / 85.02% ± std: 0.78 / 0.88	Acc: 94.40% / 84.71% ± std: 0.76 / 0.86 F1-Score: 96.27% / 84.88% ± std: 0.70 / 0.89

the *ZZFeatureMap*, in all its entanglement variants, exhibits superior performance, with higher average and maximum values compared to other configurations. However, the highest value is achieved by *PauliFeatureMap [Y]*. Applying the VQC to this dataset reveals that configurations involving an X-rotation tend to yield lower results compared to those using Z- and Y-rotations. Additionally, it can be observed that *PauliFeatureMap [Y]* achieves very high values across all ansatz combinations.

5.2.2 Optimizer analysis

The choice of optimizer plays a crucial role in determining the efficiency of a variational quantum algorithm, both in terms of accuracy and convergence speed. Qiskit provides a wide variety of optimizers, categorized into three types: Gradient-based optimizers, Gradient-free optimizers, and Hardware-specific optimizers, where gradient evaluation depends on quantum architecture. All analyses conducted on the VQC used *COBYLA* with a maximum of 50 iterations, as commonly reported in the literature. In addition to COBYLA, other optimizers were considered in this work. As shown in Table 15, the analysis focused on the best results obtained with COBYLA, *SPSA*, *NFT*, *NELDER_MEAD*, and *POWELL*, all executed with a maximum of 200 iterations.

The results demonstrate the performance of various optimizers applied to different configurations of quantum classifiers.

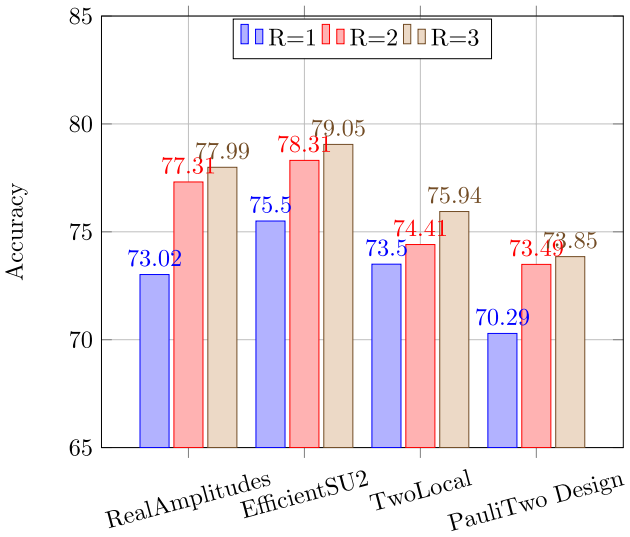

Fig. 5 Ansatz average performance for r=1-3

Among the optimizers, COBYLA generally achieves the highest accuracies, SPSA and NFT also perform well, but are less stable. Interestingly, NELDER MEAD and POWELL achieve competitive results in specific cases, such as q=3, ZZFeatureMap (Circular), EfficientSU2, r=2, where NELDER MEAD achieves 94.21% accuracy. The findings suggest that the choice of optimizer significantly impacts performance and should be carefully selected based on the configuration and circuit design.

5.2.3 Noisy simulation

Table 16 summarizes the performance of VQC across the same noise backends of the previous model on ToN_IoT and NSL-KDD datasets. The configuration 3 QUBIT, PauliFeatureMap [Y], RealAmplitudes(r=2), COBYLA consistently delivers the best overall results, achieving high accuracy and F1-scores across all noise models, with accuracy above 91.57% and F1-Score up to 94.50%.

In contrast, configuration like 4 QUBIT, ZZFeatureMap(Full), RealAmplitudes (r=3) and SPSA shows significant performance drops, with F1-scores falling close to 50.00%. These results highlight the impact of configuration and optimizer selection on VQC performance under different noise conditions. Specifically, FakeNairobi and FakePerth perform better than FakeLagos with circuits that require more entanglement, such as the ZZFeatureMap, and with EfficientSU2, which is more complex compared to Real Amplitudes.

5.3 Hybrid classical-quantum neural network

Finally, the last model addressed in this work is the hybrid neural network. Its significance lies in its ability to leverage the distinctive features of both the classical

Table 14 VQC performance for each ansatz and featuremap

Ansatz	ZZFeatureMap			PauliFeatureMap							
	Linear	Full	Circular	Z	Y	XZ	ZX	YZ	ZY	Z,YY	Z, XX	Z,Y,ZZ
RealAmplitudes	85.04	92.48	84.19	88.75	91.54	82.18	75.40	89.62	89.74	86.13	78.90	83.93
EfficientSU2	90.57	90.81	90.81	88.44	91.82	82.28	83.91	90.55	86.32	93.17	89.13	90.26
TwoLocal	86.03	84.78	88.23	81.45	91.54	84.10	80.86	84.03	84.62	86.20	75.99	81.76
PauliTwoDesign	87.24	82.99	87.36	81.99	83.32	86.08	76.84	79.14	79.77	84.64	70.65	77.65

Table 15 VQC best configurations with several optimizers

Best configurations	COBYLA	SPSA	NFT	NELDER MEAD	POWELL
q=3, ZZ (Linear), EfficientSU2, r=2	91.23% ± 0.74	89.10% ± 0.69	90.10% ± 0.83	85.18% ± 0.91	87.43% ± 0.77
q=3, Pauli [Y], RealAmplitudes, r=2	92.13% ± 0.71	91.47% ± 0.66	91.54% ± 0.70	88.42% ± 0.82	83.06% ± 0.89
q=5, Pauli [Z, YY], EfficientSU2, r=2	91.11% ± 0.78	92.79% ± 0.69	90.17% ± 0.83	81.73% ± 0.95	86.29% ± 0.87
q=3, ZZ (Circular), EfficientSU2, r=2	90.81% ± 0.73	92.13% ± 0.77	94.18% ± 0.64	94.21% ± 0.67	89.84% ± 0.79
q=3, Pauli [Z,Y,ZZ], EfficientSU2, r=3	90.26% ± 0.69	93.00% ± 0.65	93.60% ± 0.68	92.98% ± 0.72	93.57% ± 0.70
q=4, ZZ (Full), RealAmplitudes, r=3	92.48% ± 0.71	81.43% ± 0.92	83.25% ± 0.88	85.99% ± 0.85	82.37% ± 0.94

and quantum domain. Unlike the previous models, the fine tuning of the hybrid model does not analyze the best combinations of feature maps and ansatzes for its quantum component. However, the knowledge acquired in earlier phases was utilized to inform the implementation of this model and a modified version of the hybrid model was implemented.

5.3.1 Structure

The generic structure is described in Fig. 3. The quantum circuit used in the model consists of a feature map and an ansatz. Since the context involves neural networks, the analysis from the VQC model was utilized as a starting point to select the best combinations of feature maps and ansatzes. For the quantum component, a Quantum Neural Network (QNN) was developed using Qiskit's `EstimatorQNN`. It allows for the construction of a QNN using parameterized quantum circuits. Unlike VQC, which is specifically designed for classification and includes several predefined elements for that purpose, EstimatorQNN requires more manual configuration. EstimatorQNN works by computing the expected value of quantum observables within the circuit and using these values to optimize the circuit's parameters during training. The *TorchConnector* class is then used to integrate the Quantum Neural Network into PyTorch, enabling its use as a PyTorch module.The hybrid network was optimized using the Adam optimizer.

5.3.2 Results

From the results obtained with a noiseless simulator in Table 17, it emerges that 3-qubit configurations perform well, achieving high accuracy and F1-scores, though performance varies based on the chosen feature map and ansatz. A 5-qubit setups improve results, with the best configuration reaching 97.06% accuracy. The highest performance is obtained with 6 qubits, where the ZZFeatureMap (Full) and EfficientSU2 (r=2) achieve 97.12% accuracy. However, this improvement comes at the cost of increased circuit complexity and computational overhead, as higher-qubit configurations require more resources for training and simulation.

Analyzing this optimal configuration under noisy conditions, reported in Table 18, reveals that the results are highly dependent on the backend used. For instance, in the case of FakeLagos, there is a noticeable drop in performance, likely due to greater sensitivity to errors in two-qubit gates given PauliFeatureMap ["ZY"] and EfficientSU2, while FakeNairobi maintains a consistent level of performance, achieving an accuracy of 94.86% and an F1-score of 94.83%. This variation highlights the impact of backend architecture and noise levels on the reliability of quantum-assisted models. For this reason, further testing is essential to thoroughly evaluate the robustness of the models across different backends.

6 Time evaluation

When assessing the performance of machine learning models, accuracy and generalization capabilities are often the primary focus. However, computational efficiency plays a crucial role, particularly in quantum and hybrid quantum-classical models, where resource constraints and execution time can significantly impact practical usability.

Computing the computational efficiency is especially relevant in quantum simulations, where execution time correlates with the quantum resources utilized rather than reflecting the real physical runtime on quantum hardware. The time metrics reported in this section are therefore measured on a CPU-based simulator and should be interpreted as indicators of simulation cost, not as direct estimates of execution time on an actual quantum processor. A longer simulation time typically indicates higher circuit complexity, either due to an increased number of qubits processed in parallel or deeper parameterized quantum circuits, thus providing a useful alternative for assessing scalability trends and model feasibility.

The simulations were executed on a CPU server with the following specifications: Processor Intel Xeon (56 cores, base/boost frequency 1.9–4.8 GHz); Cache memory: 105

Table 16 VQC performance of best configurations on Fake Nairobi, Fake Lagos and Fake Perth with ToN_IoT / NSL-KDD, the best one is highlighted

Configuration	Fake Nairobi	Fake Lagos	Fake Perth
3 QUBIT, ZZFeatureMap(Linear), EfficientSU2 (r=2), COBYLA	Acc: 90.22% / 89.60% ± std: 0.71 / 0.68 F1-Score: 93.59% / 88.95% ± std: 0.65 / 0.79	Acc: 89.77% / 89.05% ± std: 0.69 / 0.72 F1-Score: 93.26% / 88.40% ± std: 0.67 / 0.81	Acc: 90.90% / 87.99% ± std: 0.63 / 0.84 F1-Score: 93.98% / 74.22% ± std: 0.61 / 0.92
3 QUBIT, PauliFeatureMap [Y], RealAmplitudes(r=2), COBYLA	**Acc: 91.57% / 87.54%** **± std: 0.66 / 0.79** **F1-Score: 94.50% / 85.40%** **± std: 0.59 / 0.83**	**Acc: 91.19% / 86.72%** **± std: 0.64 / 0.82** **F1-Score: 94.27% / 84.66%** **± std: 0.57 / 0.86**	**Acc: 91.28% / 87.49%** **± std: 0.63 / 0.80** **F1-Score: 94.32% / 85.28%** **± std: 0.56 / 0.84**
3 QUBIT, PauliFeatureMap [Y], RealAmplitudes(r=2), SPSA	Acc: 88.78% / 77.96% ± std: 0.74 / 0.88 F1-Score: 92.37% / 74.22% ± std: 0.68 / 0.90	Acc: 35.86% / 63.04% ± std: 0.93 / 0.81 F1-Score: 30.04% / 62.27% ± std: 0.94 / 0.79	Acc: 58.14% / 64.21% ± std: 0.86 / 0.78 F1-Score: 66.29% / 62.59% ± std: 0.80 / 0.77
5 QUBIT, PauliFeatureMap [Z, YY], EfficientSU2 (r=2), SPSA	Acc: 81.90% / 79.13% ± std: 0.83 / 0.81 F1-Score: 89.11% / 77.94% ± std: 0.70 / 0.85	Acc: 74.79% / 75.48% ± std: 0.86 / 0.84 F1-Score: 85.53% / 73.18% ± std: 0.73 / 0.88	Acc: 82.21% / 84.55% ± std: 0.79 / 0.77 F1-Score: 89.25% / 83.65% ± std: 0.69 / 0.75
3 QUBIT, ZZFeatureMap(Circular), EfficientSU2(r=3), NELDER_MEAD	Acc: 89.52% / 88.73% ± std: 0.72 / 0.74 F1-Score: 91.46% / 87.85% ± std: 0.66 / 0.79	Acc: 74.63% / 78.29% ± std: 0.87 / 0.82 F1-Score: 85.46% / 75.76% ± std: 0.76 / 0.88	Acc: 74.60% / 78.62% ± std: 0.85 / 0.83 F1-Score: 85.45% / 76.30% ± std: 0.77 / 0.86
q=4, ZZFeatureMap(Full), RealAmplitudes, (r=3), SPSA	Acc: 56.17% / 55.03% ± std: 0.90 / 0.91 F1-Score: 44.59% / 41.94% ± std: 0.94 / 0.95	Acc: 27.50% / 55.11% ± std: 0.97 / 0.91 F1-Score: 18.62% / 40.93% ± std: 0.98 / 0.94	Acc: 32.34% / 54.23% ± std: 0.96 / 0.92 F1-Score: 31.55% / 43.76% ± std: 0.95 / 0.93

Table 17 Hybrid quantum-classical network noiseless performance on best configurations with ToN_IoT / NSL-KDD, the best one is highlighted

Configuration	Noiseless simulation
3 QUBIT, PauliFeatureMap [“Z”, “YY”], RealAmplitudes (r=3)	Acc: 93.07% ± 0.62 / 94.60% ± 0.55 Precision: 92.28% ± 0.65 / 94.80% ± 0.57 Recall: 89.09% ± 0.72 / 94.42% ± 0.59 F1-Score: 90.53% ± 0.68 / 94.56% ± 0.58
3 QUBIT, PauliFeatureMap [“YZ”], EfficientSU2 (r=2)	Acc: 93.10% ± 0.61 / 94.95% ± 0.54 Precision: 92.30% ± 0.64 / 94.94% ± 0.55 Recall: 89.10% ± 0.71 / 94.90% ± 0.57 F1-Score: 90.50% ± 0.67 / 94.92% ± 0.56
3 QUBIT, PauliFeatureMap [“Z”, “YY”], EfficientSU2 (r=2)	Acc: 95.46% ± 0.55 / 93.39% ± 0.63 Precision: 94.09% ± 0.57 / 93.54% ± 0.62 Recall: 93.93% ± 0.58 / 93.21% ± 0.65 F1-Score: 94.01% ± 0.56 / 93.33% ± 0.64
5 QUBIT, PauliFeatureMap [“Y”], RealAmplitudes (r=3)	Acc: 95.48% ± 0.53 / 95.48% ± 0.53 Precision: 94.06% ± 0.56 / 95.63% ± 0.52 Recall: 94.03% ± 0.56 / 95.32% ± 0.54 F1-Score: 94.05% ± 0.55 / 95.44% ± 0.53
3 QUBIT, PauliFeatureMap [“ZY”], EfficientSU2 (r=2)	Acc: 95.56% ± 0.52 / 90.98% ± 0.72 Precision: 94.09% ± 0.54 / 92.66% ± 0.68 Recall: 93.93% ± 0.55 / 90.33% ± 0.73 F1-Score: 94.01% ± 0.54 / 90.75% ± 0.71
5 QUBIT, PauliFeatureMap [“ZY”], EfficientSU2 (r=1)	Acc: 95.48% ± 0.53 / 97.06% ± 0.49 Precision: 94.13% ± 0.55 / 97.21% ± 0.47 Recall: 93.95% ± 0.56 / 96.93% ± 0.48 F1-Score: 94.04% ± 0.54 / 97.04% ± 0.47
5 QUBIT, PauliFeatureMap[“Z”, “YY”], EfficientSU2 (r=2)	Acc: 88.50% ± 0.77 / 96.96% ± 0.50 Precision: 88.50% ± 0.77 / 97.09% ± 0.49 Recall: 88.10% ± 0.78 / 96.83% ± 0.51 F1-Score: 83.10% ± 0.85 / 96.94% ± 0.50
6 QUBIT, ZZFeatureMap(Full), EfficientSU2 (r=2)	Acc: 95.43% ± 0.54 / 97.12% ± 0.48 Precision: 96.95% ± 0.50 / 97.23% ± 0.47 Recall: 96.98% ± 0.50 / 97.00% ± 0.48 F1-Score: 96.97% ± 0.51 / 97.10% ± 0.47

Table 18 Hybrid quantum-classical network performance of the best configuration on Fake Nairobi, Fake Lagos and Fake Perth

Configuration	Fake Nairobi	Fake Lagos	Fake Perth
6 QUBIT, ZZFeatureMap(Full), EfficientSU2 (r=2)	Acc: 94.86% ± 0.62 Prec.: 94.88% ± 0.61 Rec.: 94.80% ± 0.63 F1-Sc.: 94.83% ± 0.62	Acc: 90.89% ± 0.75 Prec.: 91.90% ± 0.70 Rec.: 90.61% ± 0.77 F1-Sc.: 91.25% ± 0.73	Acc: 91.05% ± 0.72 Prec.: 92.77% ± 0.68 Rec.: 90.40% ± 0.74 F1-Sc.: 91.57% ± 0.71

MB; RAM: 256 GB; Operating System Ubuntu 24.04.3 LTS. These details are provided to ensure reproducibility and contextualize the reported absolute times.

It is important to emphasize that these results are not intended for a direct performance comparison with classical models. Since the experiments were conducted entirely on a quantum simulator, the reported times primarily reflect the computational load of emulating quantum circuits rather than their physical execution latency. Consequently, a realistic comparison with classical training and inference times cannot be made without running the same models on real quantum hardware, which was beyond the scope of this study.

Within this framework, the execution times provide a quantitative comparison across different configurations of the three QML models evaluated: Pegasos-QSVC, VQC, and Hybrid Classical–Quantum Neural Network. Training time reflects how configuration choices influence the cost of model development, while prediction time offers insights into inference efficiency. This analysis allows us to identify architectures that best balance accuracy and computational efficiency, providing qualitative guidance on scalability without implying absolute performance on real quantum devices.

6.1 Pegasos-QSVC

The training time for Pegasos-QSVC varies significantly across different configurations tested in Table 12, as shown in Fig. 6. The lowest training times are observed for configurations with fewer qubits and simpler feature maps, such as with 4 qubits, PauliFeatureMap [Z,YY] repeted once, which takes 18.41s. Conversely, the longest training time (85.36s) corresponds to the configuration with $q = 6$, a PauliFeatureMap [Z,XX], and $r = 2$, indicating that increasing feature map complexity and repetitions can significantly impact computational costs.

Interestingly, configurations with comparable accuracy present significant differences in time requirements, with

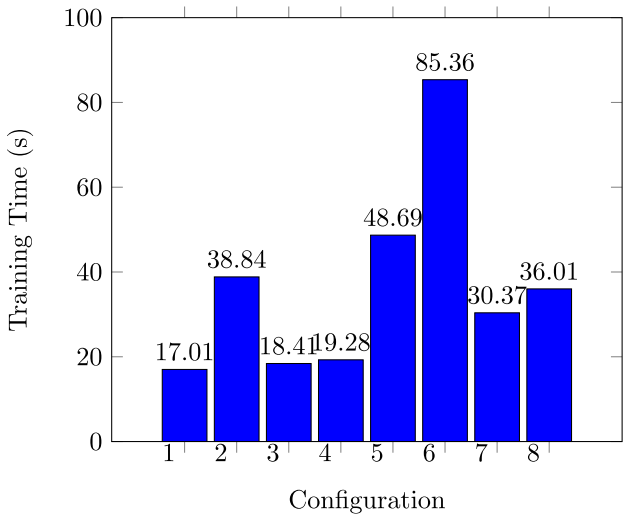

Fig. 6 Training time for each Pegasos-QSVC configuration of Table 12

one taking up to three times longer than another, highlighting the impact of hyperparameter choices on computational efficiency.

One distinctive characteristic of Pegasos-QSVC is that it achieves relatively low training times compared to other quantum models but exhibits significantly higher prediction times. This trade-off is particularly evident in Fig. 7, where inference time per sample varies from 0.31s to 2.4s, meaning that some configurations take nearly eight times longer for prediction. This considerable inference cost motivated us to compute prediction time in addition to training time, as it plays a crucial role in real-world applications where fast inference is essential.

It is important to note that in a noisy simulation environment, these proportions remain consistent, but overall execution times increase due to the additional overhead introduced by quantum simulators. This highlights the need to optimize quantum kernel-based models to balance accuracy and computational feasibility.

6.2 VQC

The training time of the VQC is strongly influenced by the choice of ansatz. Among the ansatzes tested, RealAmplitudes is typically the fastest due to its minimal number of parameters and simple structure. EfficientSU2 follows closely, as it strikes a balance between efficiency and expressivity. TwoLocal, while potentially efficient, depends heavily on its specific configuration and the choice of entanglement structure. On the other end of the spectrum, PauliTwoDesign is generally the slowest, as its greater depth and complexity lead to increased computational costs. These differences highlight the importance of selecting an appropriate ansatz, especially when training speed is a critical factor.

Another major factor impacting VQC training time is the choice of optimizer. As shown in Fig. 8, COBYLA consistently exhibits the fastest execution time across all configurations described in Table 15, with values ranging between 1,773s and 5,041s. SPSA, Nelder-Mead, and NFT show moderate execution times, with SPSA and NFT being slightly slower than Nelder-Mead in most cases. Meanwhile, Powell is by far the slowest optimizer, often taking more than double the time of the other methods. For example, in Config 3, Powell reaches 24,552s, while COBYLA completes the same task in just 5,041s.

These results emphasize that both ansatz selection and optimization strategy are key considerations for reducing computational costs in VQC training. Choosing an efficient

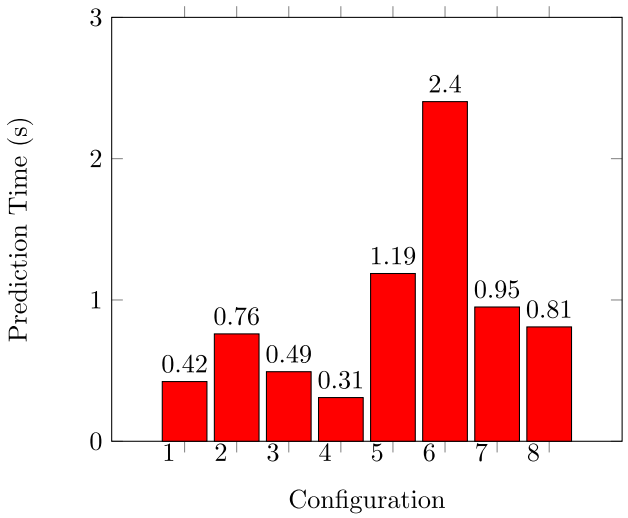

Fig. 7 Prediction time per sample for Pegasos-QSVC configuration of Table 12

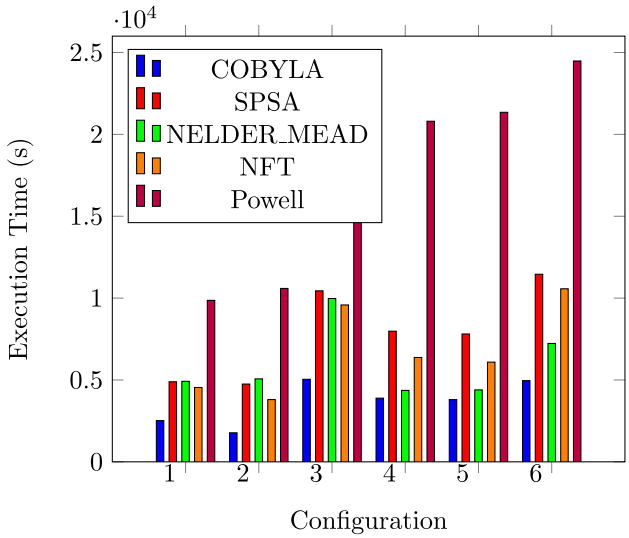

Fig. 8 Execution time per optimizer and configuration of Table 15

ansatz and optimizer can significantly shorten training time, making VQC more viable for practical applications.

6.3 Hybrid classical-quantum neural network

The execution time of the Hybrid Classical-Quantum Neural Network, across different configurations described in Table 17, is illustrated in Fig. 9. A clear trend emerges, indicating that an increase in the number of qubits and the number of repetitions leads to a substantial rise in computational cost. For instance, while Configuration 1 completes in 36,970 seconds, Configuration 8, characterized by a higher qubit count and a simpler structure, requires 286,353 seconds, representing nearly an eightfold increase. This trend highlights the exponential growth in execution time as the complexity of the quantum component of the model increases.

Additionally, as previously discussed in the context of the VQC, the choice of ansatz has a notable impact on training efficiency. The data suggests that configurations utilizing the RealAmplitudes ansatz generally achieve faster execution times compared to those employing EfficientSU2. This is consistent with the fact that RealAmplitudes has a simpler structure with fewer parameters, resulting in reduced computational overhead. The difference becomes particularly pronounced in configurations with a larger number of qubits and repetitions, where the EfficientSU2 ansatz exhibits a steeper increase in execution time.

These observations underscore the necessity of careful model design when implementing hybrid quantum-classical neural networks. While deeper and more expressive circuits may enhance model performance, they also entail significantly higher computational costs. Therefore, selecting an appropriate ansatz and optimizing circuit complexity is essential for balancing accuracy and efficiency in practical applications.

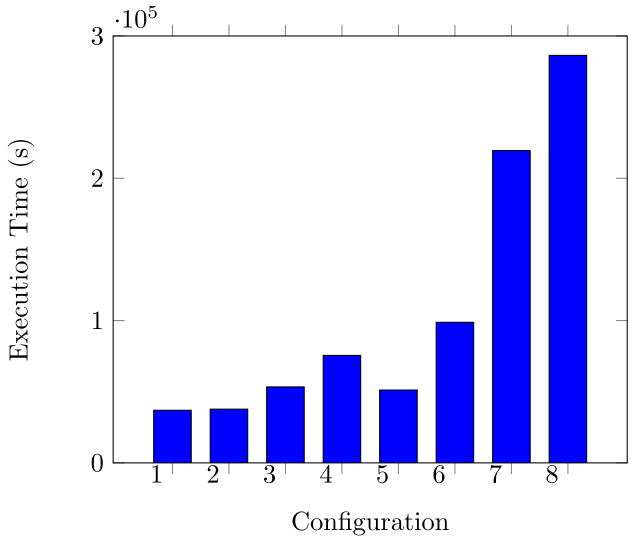

Fig. 9 Execution time per configuration of Table 17

7 Comparison with classical models

Although the primary objective of this study was to investigate the effects of distinct noise sources originating from different quantum backends on various quantum circuit architectures, focusing on the influence of key design parameters on QML performance under both noiseless and noisy conditions, we additionally performed a comparative analysis with established ML algorithms. This complementary evaluation provides a meaningful benchmark to contextualize the current maturity and practical potential of QML techniques.

To ensure a fair and consistent comparison, both classical and quantum models were trained and tested on identical data subsets and subjected to the same preprocessing pipeline, including normalization and dimensionality reduction through PCA. The only controlled variation concerned the number of PCA components, which was aligned with the number of qubits employed in each quantum configuration, ranging from 2 to 10. This setup guaranteed that classical and quantum models operated on feature spaces of equivalent dimensionality and statistical properties.

The selected classical algorithms (SVM, XGBoost, Random Forest, and Neural Network) were chosen based on their prevalence and competitiveness in contemporary literature addressing comparable classification tasks. The corresponding performance metrics are reported in Table 19.

Table 19 Accuracy of classical models for varying numbers of features with ToN_IoT / NSL-KDD datasets

Features	SVM	XGBoost	Random forest	Neural networks
2	90.10% / 88.40%	91.53% / 89.85%	91.21% / 89.10%	89.92% / 88.10%
3	91.84% / 90.30%	93.62% / 92.10%	93.13% / 91.40%	91.52% / 89.80%
4	93.05% / 91.80%	95.17% / 93.60%	94.43% / 92.30%	92.36% / 90.70%
5	96.03% / 91.75%	98.96% / 95.13%	98.65% / 94.35%	95.54% / 92.45%
6	96.08% / 92.03%	98.87% / 95.24%	98.70% / 94.22%	95.54% / 91.98%
7	96.22% / 92.55%	**99.10%** / 96.02%	98.70% / 95.63%	95.70% / 92.15%
8	96.24% / 92.15%	99.01% / 96.18%	98.70% / 95.07%	95.91% / 92.33%
9	96.27% / 92.23%	98.89% / 95.57%	98.20% / 95.56%	96.03% / 92.78%
10	96.24% / 92.22%	97.99% / 96.01%	98.61% / 95.73%	96.03% / 92.75%

From the observed outcomes, it is evident that the model accuracy generally improves with an increasing number of features, up to a certain point beyond which some models tend to plateau or decrease. Among the models tested, XGBoost emerges as the top-performing algorithm, achieving the highest accuracy with seven features. Random Forest demonstrates notable stability and consistently high accuracy. Both SVM and Neural Networks yield solid performance, showing gradual improvements as the number of features increases.

By comparing these values with the results obtained from the quantum algorithms executed under noisy conditions, it becomes evident that classical models still exhibit higher peak performance in terms of accuracy, for instance, XGBoost achieves a maximum of 99.1%. Nevertheless, quantum models demonstrate competitive behavior in low-dimensional feature spaces, highlighting their potential efficiency in scenarios with limited data representations. In particular, the Pegasos-QSVC reaches an accuracy of 95.35% when trained with only four features (corresponding to four qubits), a result not attained by the classical models under the same configuration.

This observation suggests that quantum learning architectures may achieve faster convergence and maintain strong generalization properties even with a reduced number of input features. Such characteristics are particularly relevant in applications where feature extraction or data acquisition is costly, or where model complexity must remain constrained. Although quantum models currently remain limited by noise and qubit fidelity, these findings indicate that, with continued hardware improvements and optimized circuit designs, QML approaches could offer meaningful advantages in compact and resource-constrained learning settings.

8 Final analysis and discussion

8.1 Models and data

The best results for each QML algorithm are summarized in Tables 20 and 21, corresponding to noiseless and noisy simulations, respectively. Among the evaluated models, Pegasos-QSVC emerges as the most efficient, achieving the best overall performance while also being the fastest in terms of training time. This highlights the advantage of kernel-based approaches, which avoid the optimization challenges of variational quantum circuits while maintaining high accuracy. The model's relatively low training times across different configurations make it a promising candidate for real-world applications, particularly in scenarios where scalability and computational efficiency are key constraints. However, one notable drawback is its high prediction time, which motivated us to include inference time as an additional evaluation metric.

Table 20 Best configuration for each model without noise with ToN_IoT / NSL-KDD datasets

Configuration	Performance
PegasosQSVC - 7 QUBIT, ZFeatureMap, r=2	Accuracy: 95.44% / 95.80% F1 Score: 96.94% / 95.23%
VQC - 3 QUBIT, ZZFeatureMap(Circular), EfficientSU2, r=3, NELDER_MEAD	Accuracy: 94.21% / 93.21% F1 Score: 96.10% / 93.10%
Hybrid Quantum-Classical Neural Network - 6 QUBIT, ZFeatureMap(Full), r=1, ADAM	Accuracy: 95.43% / 97.12% F1 Score: 96.97% / 97.10%

Table 21 Best configuration for each model with noise with ToN_IoT / NSL-KDD datasets

Configuration	Performance
PegasosQSVC - 7 QUBIT, ZFeatureMap, r=2	Accuracy: 93.24% / 94.60% F1 Score: 95.54% / 94.13%
VQC - 3 QUBIT, PauliFeatureMap [Y], RealAmplitudes(r=2), COBYLA	Accuracy: 91.57% / 87.54% F1 Score: 94.50% / 85.40%
Hybrid Quantum-Classical Neural Network - PauliFeatureMap [Y], 5 QUBIT, r=1, ADAM	Accuracy: 94.86% / 93.78% F1-Score: 94.83% / 93.90%

The VQC also delivers competitive results in both noiseless and noisy conditions, demonstrating the effectiveness of variational quantum models in classification tasks. However, its performance is highly sensitive to hyperparameters such as qubit count, circuit depth, ansatz structure, and optimizer choice. Our analysis indicates that RealAmplitudes is generally the fastest ansatz due to its simplicity, while PauliTwoDesign is the slowest because of its greater depth and number of parameters. Regarding optimizers, COBYLA consistently achieves the best and more stable accuracy and shortest execution times, whereas Powell exhibits the longest training durations, often requiring more than twice the time of other methods. This strong dependence on optimization strategies makes VQC less predictable in terms of computational cost, requiring systematic tuning to identify the best-performing configurations.

The Hybrid Classical-Quantum Neural Network demonstrates strong performance in noiseless simulations, leveraging the strengths of both classical and quantum paradigms. However, execution time increases significantly with higher qubit counts and deeper circuits, as evidenced in our analysis. The results further confirm that RealAmplitudes is computationally more efficient than EfficientSU2, reinforcing our findings from VQC. While this model achieved notable accuracy, particularly in certain noisy environments such as FakeNairobi, where it reached an accuracy of 94.86% and

an F1-score of 94.83%, its performance is more susceptible to noise than other models, leading to variability across different quantum backends.

The observed performance disparity between the ToN_IoT and NSL-KDD datasets underscores the challenges associated with generalization in quantum machine learning models. While models generally achieve high and consistent performance on ToN_IoT, their effectiveness diminishes on NSL-KDD, indicating that model behavior is sensitive to dataset characteristics and design choices.

Several factors may contribute to this performance gap. First, the two datasets differ in their statistical distributions, feature spaces, and complexity of attack scenarios. ToN_IoT encompasses a broader and more heterogeneous set of attacks, with richer and more informative features, which may facilitate learning and pattern recognition. NSL-KDD, in contrast, presents a different structure, with temporal and categorical dependencies that may be less aligned with the inductive biases of the models.

Second, the encoding of classical data into quantum circuits and the choice of feature maps play a critical role in determining expressivity and robustness. Encodings that effectively capture the structure of one dataset may be suboptimal for another, leading to reduced generalization.

Overall, these observations highlight that consistent cross-dataset performance requires careful consideration of both dataset properties and circuit design, suggesting that quantum models must balance expressivity with robustness to achieve reliable generalization.

8.2 Noise

The quantitative assessment of noise influence across different quantum models is presented in Fig. 10a–c, which depict the frequency distributions of F_1-scores as a function of circuit complexity, measured by the total number of CNOT gates of the tested circuits, averaging the 2 datasets tested.

Here, the F_1-score frequency represents the number of occurrences of a given F_1 value obtained across the multiple experimental configurations. Rather than reporting a single averaged score, the heatmaps capture the empirical distribution. High-frequency regions indicate not only high accuracy, but also robustness and reproducibility of the model across varying noise conditions.

These heatmaps reveal distinct patterns of sensitivity to noise and circuit depth among the three model families (QSVC, VQC, and HQNN). In the QSVC configuration (Fig. 10a), the F_1-score distribution exhibits a broad and almost symmetric bell-shaped region centered between 25 and 35 CNOTs, where F_1-score stabilizes around 92%. This indicates that the quantum kernel-based approach maintains a favourable balance between expressivity and noise accumulation up to moderate circuit sizes, with performance gradually degrading beyond 60 CNOTs as two-qubit errors become dominant.

The VQC model (Fig. 10b) shows a markedly different trend, with a much narrower high-frequency region concentrated between 8 and 15 CNOTs, corresponding to peak F_1-scores of approximately 90%. Performance declines sharply beyond this threshold, reflecting the higher susceptibility of variational circuits to noise amplification during deeper parameterized evolutions. In contrast, the HQNN model (Fig. 10c), which combines quantum feature extraction with classical post-processing, displays a smoother and more extended plateau of stability. F_1-scores remain consistently above 85% across the 20–35 CNOT interval, and the degradation beyond this region is gradual, confirming that hybrid quantum–classical designs effectively mitigate the propagation of quantum noise.

A particularly interesting observation arises from the leftmost region of all three heatmaps, corresponding to circuits with zero or nearly zero CNOT operations. In this case, the distributions become highly irregular and dispersed, indicating unpredictable performance and unstable behavior. The corresponding column in the heatmaps shows a dense

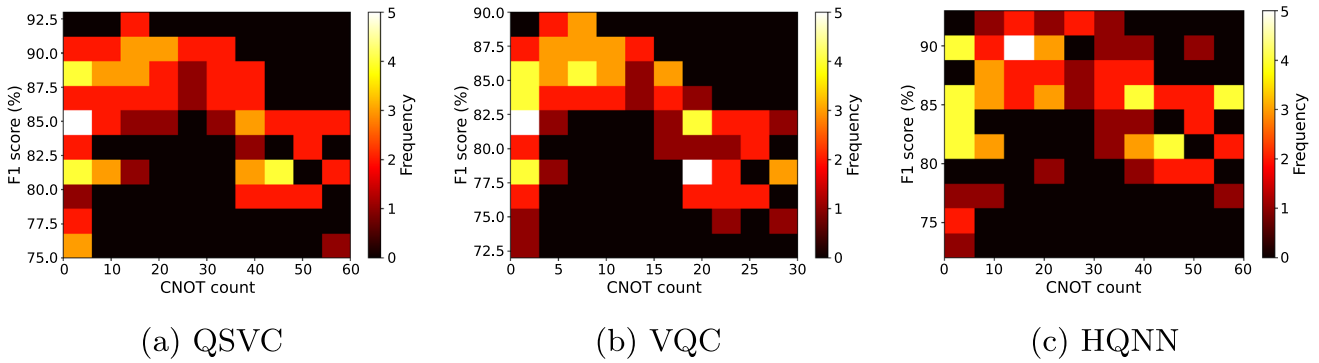

Fig. 10 Frequency heatmaps of F_1-scores versus CNOT count for the three quantum models under noisy simulation. Each subfigure highlights the region of optimal performance, where model expressivity and noise accumulation are balanced

concentration of occurrences spanning almost the entire range of F_1 values, from the lowest to the highest. This suggests that in the complete absence of entangling operations, the models rely purely on single-qubit transformations, leading to non-deterministic and dataset-dependent behaviors that do not generalize across backends or feature maps. Consequently, a minimum level of controlled entanglement appears to be essential to achieve consistent and noise-resilient learning performance.

Complementary insights are obtained from the comparison of fake backend simulations, which emulate specific IBM Quantum devices. *FakeLagos*, characterized by higher two-qubit gate error rates, consistently exhibits performance degradation when employing highly entangled feature maps (e.g., *PauliFeatureMap*) or complex variational forms such as *EfficientSU2*. Conversely, *FakeNairobi* and *FakePerth*, which exhibit lower gate error probabilities and better calibration fidelity, support more entangled and deeper circuit structures without substantial accuracy loss. These observations reinforce that once a target backend is selected for training, its tolerance to two-qubit errors should be carefully profiled to determine the most appropriate circuit design. Specifically, for noise-sensitive processors like *FakeLagos*, adopting simpler and shallower architectures is advisable, while more robust backends such as *FakeNairobi* or *FakePerth* can effectively sustain higher entanglement levels and circuit depth with minimal performance degradation.

Building upon these observations, the quantitative analyses allow us to extract explicit design principles for noise-resilient QML architectures, directly linking circuit complexity to performance under realistic noise conditions. The following guidelines summarize the main insights from our heatmap analyses and hardware comparison:

- **Circuit depth and entanglement:** For near-term devices, shallow and moderately entangled circuits generally maximize robustness. In our experiments, total CNOT counts in the 20–35 range provide the best trade-off between expressivity and error accumulation, keeping F_1 -scores consistently above 85%.
- **Feature map choice:** Simple feature maps such as *ZFeatureMap* or *ZZFeatureMap* with r = 1–2 outperform deeper, highly entangled maps on noisy backends, limiting propagation of two-qubit gate errors without sacrificing classification capability.
- **Variational vs. hybrid models:** Fully variational circuits (VQC) perform well up to ∼25 CNOTs, but hybrid quantum-classical networks (HQNN) provide more stable results for intermediate-depth circuits, compensating for stochastic gate errors and reducing variance across multiple runs.
- **Backend-specific tolerance:** Noise characteristics of the hardware strongly influence performance. For example, *FakeLagos* consistently shows lower noise tolerance, requiring shallower circuits, whereas *FakeNairobi* and *FakePerth* allow moderately deeper architectures with limited degradation. Understanding the backend's noise profile is crucial for mapping QML model design to hardware constraints.

In the specific context of quantum machine learning for intrusion detection under noisy conditions, the Pegasos-QSVC model implemented with 7 qubits, a *ZFeatureMap*, and $r = 2$ demonstrated good performance. This configuration achieved accuracies of 93.24% and 94.60%, and F1-scores of 95.54% and 94.13% on the ToN_IoT and NSL-KDD datasets, respectively. These results indicate an improvement over previously reported quantum baselines (accuracy $\approx$ 89.7%, F1-score $\approx$ 86%), highlighting the effectiveness of this setup for QML-based anomaly detection under realistic noise scenarios.

8.3 Assumptions and limitations

This study is conducted under assumptions consistent with the current capabilities of near-term quantum hardware, with experimental results obtained using noisy simulations and fake backend models that emulate the noise characteristics of real quantum processors. While these backends provide a realistic approximation of noise, they cannot fully capture all sources of hardware variability, and the reported performance should therefore be interpreted as indicative of expected behavior rather than definitive for execution on physical devices. The observed performance improvements are achieved for specific circuit configurations and feature encodings and do not imply universal optimality across different datasets, model architectures, or hardware platforms. Achieving quantum advantage in QML-based intrusion detection will require continued progress in quantum hardware reliability and noise mitigation, as well as the development of quantum-native learning algorithms capable of more fully exploiting superposition and entanglement to surpass classical approaches as the technology matures.

9 Conclusion and future work

This work presented a systematic evaluation of three quantum machine learning models for intrusion detection, exploring a wide range of architectural and training configurations under both ideal and noisy conditions. The analysis revealed how circuit complexity, entanglement,

and backend noise characteristics jointly affect model performance and robustness.

Among the tested configurations, Pegasos-QSVC achieved an accuracy of 94.60%, VQC reached 91.57%, and the hybrid quantum–classical model attained 94.86%. Beyond accuracy, the study quantitatively assessed noise impact through the F1-score distribution as a function of circuit complexity, identifying optimal design ranges and instability regions. Execution time was also analyzed as an alternative for model complexity, offering insight into accuracy–efficiency trade-offs.

Overall, this work provides a comprehensive and noise-aware perspective on QML for cybersecurity, establishing practical design guidelines for enhancing robustness in realistic quantum environments and contributing to a deeper understanding of how QML models can be effectively optimized for near-term quantum hardware.

Future research could focus on three key areas. First, expanding the range of datasets to include more complex and dynamic network scenarios can enhance the understanding of QML's scalability and applicability. Second, exploring advanced quantum techniques, such as hybrid models with federated learning and Quantum Generative Adversarial Networks (QGAN) (Cirillo and Esposito 2025b, a) for synthetic anomaly generation, could drive innovation and improve system performance. Lastly, when the use of real quantum machine will be more affordable, integrating QML models with real quantum hardware could be an opportunity to have a real result of the models tested.

Acknowledgements This work was partially supported by the Korea Institute of Energy Technology Evaluation and Planning (KETEP) grant funded by the Korea government (MOTIE) (RS-2023-00303559, Study on developing cyber-physical attack response system and security management system to maximize real-time distributed resource availability) and by the NGIsargasso project (Europe Horizon Grant No. 101092887), Open Call 4 FRQGAN4AD project. We acknowledge the use of IBM Quantum services for this work. The views expressed are those of the authors, and do not reflect the official policy or position of IBM or the IBM Quantum team.

Author Contributions Author F.C. conducted the entire literature review, designed the methodology, and performed all the experiments. Author C.E. contributed to the design of the methodology. Author J.T.S provided supervision throughout the project. All authors participated in the final analysis and interpretation of the results, and contributed to the writing and revision of the manuscript.

Funding Open access funding provided by Università degli Studi di Salerno within the CRUI-CARE Agreement. This work was partially supported by the Korea Institute of Energy Technology Evaluation and Planning (KETEP) grant funded by the Korea government (MOTIE) (RS-2023-00303559, Study on developing cyber-physical attack response system and security management system to maximize real-time distributed resource availability) and by the NGIsargasso project (Europe Horizon Grant No. 101092887), Open Call 4 FRQGAN4AD project.

Data Availability No datasets were generated or analysed during the current study.

Declarations

Competing Interests The authors declare no competing interests.

References

Abreu D, Moura D, Rothenberg C, Abelém A (2024). Quantum machine learning for network security, QuantumNetSec

Akrom M (2024) Quantum support vector machine for classification task: A review. J Multiscale Mater Inf 1(2):1–8

Anschuetz ER, Teo M, Yang W, Sud J, Kang C, Tomesh T, Chong FT (2024) Quantifying the limits of classical machine learning models using contextuality. In: 2024 IEEE international conference on quantum computing and engineering (QCE) Vol 2, pp 628–630

Bhattacharya P, Kumari A, Tanwar S, Budhiraja I, Patel S, Rodrigues JJ (2024) Quant-jack: quantum machine learning to detect cryptojacking attacks in IIoT Networks. 2024 IEEE international conference on communications workshops (ICC Workshops) pp 865–870. (ISSN: 2694–2941)

Bhavsar R, Jadav NK, Bodkhe U, Gupta R, Tanwar S, Sharma G, Sharma R (2023) Classification of potentially hazardous asteroids using supervised quantum machine learning. IEEE Access 11

Cao Z, Zhao Z, Shang W, Ai S, Shen S (2024). Using the ton-iot dataset to develop a new intrusion detection system for industrial iot devices. Multimed Tools Appl 1–29

Cerezo M, Verdon G, Huang H-Y, Cincio L, Coles PJ (2022) Challenges and opportunities in quantum machine learning. Nat Comput Sci 2(9):567–576

Cheng B, Deng X-H, Gu X, He Y, Hu G, Huang P et al (2023) Noisy intermediate-scale quantum computers. Front Phys 18(2):21308

Cirillo F, Esposito C (2025a) Federated quantum generative adversarial network for intrusion detection. In: 2025 IEEE 45th international conference on distributed computing systems workshops (icdcsw) pp 69–74

Cirillo F, Esposito C (2025b) Intrusion detection system based on quantum generative adversarial network. Proceedings of the 17th international conference on agents and artificial intelligence - volume 1: Qaio pp 830–838. SciTePress

Cirillo F, Esposito C (2025c) Intrusion detection using quantum generative adversarial networks: a federated approach with noisy simulators. Iet conference proceedings cp925, Vol 2025, pp 31–35

Dao HL (2025) Exploring new variational quantum circuit ansatzes for solving su (2) matrix models. Eur Phys J C 85(6):705

Das A, Chakrabarti A (2024) A comprehensive study of noise models simulation using various quantum simulators. In: 2024

International conference on trends in quantum computing and emerging business technologies, pp 1–6
Gad AR, Nashat AA, Barkat TM (2021) Intrusion detection system using machine learning for vehicular ad hoc networks based on ton-iot dataset. IEEE Access 9:142206–142217
Gautam K, Usha G, Nagar A, Patel S, Jain A (2024) Quantum assisted machine learning for intrusion detection systems. Kattankalathur, India, p 020280
Gong C, Guan W, Gani A, Qi H (2022) Network attack detection scheme based on variational quantum neural network. J Supercomput 78(15):16876–16897. https://doi.org/10.1007/s11227-022-04542-z
Gong C, Guan W, Zhu H, Gani A, Qi H (2024) Network intrusion detection based on variational quantum convolution neural network. J Supercomput 80(9):12743–12770. https://doi.org/10.1007/s11227-024-05919-y
Gouveia A, Correia M (2020) Towards quantum-enhanced machine learning for network intrusion detection. In: 2020 IEEE 19th international symposium on network computing and applications (NCA) pp 1–8. ISSN: 2643–7929
Guo G, Pan X, Liu H, Li F, Pei L, Hu K (2023) An iot intrusion detection system based on ton iot network dataset. In: 2023 IEEE 13th annual computing and communication workshop and conference (CCWC) pp 0333–0338
Hdaib M, Rajasegarar S, Pan L (2024) Quantum deep learningbased anomaly detection for enhanced network security. Quantum Mach Intell 6(1):26. https://doi.org/10.1007/s42484-024-00163-2
Houda ZAE, Moudoud H, Brik B, Adil M (2024) A privacy-preserving framework for efficient network intrusion detection in consumer network using quantum federated learning. IEEE Trans Consum Electron 1–1. https://doi.org/10.1109/TCE.2024.3458985
IBM Quantum (2021). https://quantum.ibm.com/
Jayalaxmi P, Kumar G, Saha R, Conti M, Kim T-H, Thomas R (2022) DeBot: A deep learning-based model for bot detection in industrial internet-of-things. Comput Electr Eng 102:108214. https://doi.org/10.1016/j.compeleceng.2022.108214
Kalinin M, Krundyshev V (2022) Security intrusion detection using quantum machine learning techniques. J Comput Virology Hacking Techn 19(1):125–136. https://doi.org/10.1007/s11416-022-00435-0
Killoran N, Bromley TR, Arrazola JM, Schuld M, Quesada N, Lloyd S (2019) Continuous-variable quantum neural networks. Phys Rev Res 1(3):033063
Kingma DP (2014) Adam: A method for stochastic optimization. arXiv:1412.6980
Küçükkara MY, Atban F, Bayılmış C (2024) Quantum-neural network model for platform independent ddos attack classification in cyber security. Adv Quantum Technol 2400084
Kukliansky A, Orescanin M, Bollmann C, Huffmire T (2024) Network anomaly detection using quantum neural networks on noisy quantum computers. IEEE Trans Quantum Eng 5:1–11. https://doi.org/10.1109/TQE.2024.3359574. Conference Name: IEEE Transactions on Quantum Engineering
Liu Z, Jia X, Li B (2024) E-healthcare application cyber security analysis using quantum machine learning in malicious user detection. Opt Quant Electron 56(3):476. https://doi.org/10.1007/s11082-023-05854-x
Maheshwari D, Sierra-Sosa D, Garcia-Zapirain B (2021) Variational quantum classifier for binary classification: Real vs synthetic dataset. IEEE Access 10:3705–3715
Misra S, Rani P (2024) Quantum machine learning: A comprehensive overview and analysis. In: 2024 15th International conference on computing communication and networking technologies (ICCCNT) pp 1–5
Muneer S, Farooq U, Athar A, Ahsan Raza M, Ghazal TM, Sakib S (2024) A critical review of artificial intelligence based approaches in intrusion detection: A comprehensive analysis. J Eng 2024(1):3909173
Nakanishi KM, Fujii K, Todo S (2020) Sequential minimal optimization for quantum-classical hybrid algorithms. Phys Rev Res 2(4):043158
Peral-García D, Cruz-Benito J, García-Peñalvo FJ (2024) Systematic literature review: Quantum machine learning and its applications. Comput Sci Rev 51:100619
Powell MJ (1994) A direct search optimization method that models the objective and constraint functions by linear interpolation. Springer
Rahman MA, Akter MS, Miller E, Timofti B, Shahriar H, Masum M, Wu F (2024) Fine-tuned variational quantum classifiers for cyber attacks detection based on parameterized quantum circuits and optimizers. 2024 IEEE 48th annual computers, software, and applications conference (COMPSAC), pp 1067–1072. ISSN: 2836–3795
Rahman MA, Shahriar H, Clincy V, Hossain MF, Rahman M, (2023). A quantum generative adversarial network-based intrusion detection system. In: 2023 IEEE 47th annual computers, software, and applications conference (COMPSAC). IEEE, Torino, Italy, pp 1810–1815
Raubitzek S, Mallinger K (2023) On the applicability of quantum machine learning. Entropy 25 (7). https://doi.org/10.3390/e25070992
Said D (2023) Quantum computing and machine learning for cybersecurity: distributed denial of service (DDoS) attack detection on smart micro-grid. Energies 16(8):3572. https://doi.org/10.3390/en16083572
Shalev-Shwartz S, Singer Y, Srebro N (2007) Pegasos: Primal estimated subgradient solver for svm. In: Proceedings of the 24th international conference on machine learning pp 807–814. New York, NY, USA: Association for Computing Machinery
Suryotrisongko H, Musashi Y (2022) Evaluating hybrid quantum-classical deep learning for cybersecurity botnet DGA detection. Proc Comput Sci 197:223–229. https://doi.org/10.1016/j.procs.2021.12.135
Tavallaee M, Bagheri E, Lu W, Ghorbani AA (2009) A detailed analysis of the kdd cup 99 data set. 2009 ieee symposium on computational intelligence for security and defense applications, pp 1–6
Tezuka H, Uno S, Yamamoto N (2024) Generative model for learning quantum ensemble with optimal transport loss. Quantum Mach Intell 6(1):6. https://doi.org/10.1007/s42484-024-00142-7
Thakkar A, Lohiya R (2023) A review on challenges and future research directions for machine learning-based intrusion detection system. Arch Comput Methods Eng 30(7):4245–4269
UNSW Sydney (2024). TONIOT datasets. https://research.unsw.edu.au/projects/toniot-datasets. Accessed: 12 Nov 2024
Venkatachalam P, Liu DQ (2023) On hybrid artificial neural networks and variational quantum classifier for network intrusion detection. In: 2023 International conference on cyber-enabled distributed computing and knowledge discovery (CyberC), pp 410–416. ISSN: 2833–8898
Yamasaki H, Isogai N, Murao M (2023) Advantage of quantum machine learning from general computational advantages