Quantum Machine Learning for Intrusion Detection on Noisy Quantum Computers

Franco Cirillo
University of Salerno
Fisciano (SA), Italy
fracirillo@unisa.it

Christian Esposito
University of Salerno
Fisciano (SA), Italy
esposito@unisa.it

***Abstract*—Intrusion Detection Systems (IDS) play a fundamental role in safeguarding digital infrastructures against cyber threats. Quantum Machine Learning (QML) presents a promising frontier in this domain, offering potential computational speedups and improved management of complex datasets. However, its practical deployment is currently limited by the constraints of Noisy Intermediate-Scale Quantum (NISQ) devices. This work explores the application of various QML approaches, specifically Quantum Support Vector Classifiers (QSVC), Variational Quantum Classifiers (VQC), and hybrid quantum-classical neural networks for anomaly detection tasks. The models were fine-tuned and benchmarked using the ToN_IoT dataset, with performance assessed on IBM's noisy quantum simulators to evaluate their behavior under realistic noise conditions. Results indicate that the Pegasos-QSVC model outperformed others, reaching an accuracy of 93.05% and an F1 score of 93.73%. Through a comparative analysis of different circuit architectures, this study identifies strategies to improve the robustness of QML systems, emphasizing their emerging potential in cybersecurity scenarios.**

***Index Terms*—Anomaly detection, QSVM, VQC, Hybrid Quantum-classical Neural Networks (HQNN), NISQ, Quantum Noisy Simulators.**

With the increasing complexity of modern network infrastructures and the continuous evolution of cyberattacks becoming ever more sophisticated, information security, authentication [1], attestation [2], and malware detection [3] have taken on unprecedented importance [4].

Machine Learning (ML) techniques have shown significant success in enhancing Intrusion Detection Systems (IDS) [5], [6]. In parallel, Quantum Machine Learning (QML) [7] is gaining traction as a next-generation approach, with the potential to outperform classical methods in certain contexts. Leveraging quantum parallelism and state representation capabilities, QML can handle complex, high-dimensional data more efficiently, potentially leading to faster training times [8] and improved detection performance [9]. Among the most actively studied QML paradigms are Quantum Support Vector Machines (QSVM), Variational Quantum Classifiers (VQC), and hybrid quantum-classical neural networks [10].

However, a significant limitation in current research is the reliance on idealized simulators that overlook the challenges introduced by hardware noise [11], [12]. Since NISQ-era quantum devices are affected by errors such as decoherence and imperfect gate operations, evaluations that ignore these aspects may misrepresent the true capabilities of QML models. While testing on real quantum machines is valuable for validation, it is often resource-intensive and costly.

An effective compromise is the adoption of noisy quantum simulators, such as those offered by IBM, which allow for controlled testing environments that emulate real hardware imperfections. These platforms support the refinement of QML architectures under more realistic conditions, enabling researchers to assess model resilience and devise strategies for noise mitigation without incurring high operational costs. In this context, the ToN_IoT dataset [13], specifically designed for cybersecurity in Industry 4.0, IoT, and IIoT ecosystems, provides a robust benchmark for evaluating anomaly detection capabilities in QML-based IDS models.

This work makes the following key contributions:

- An in-depth analysis of the performance of Pegasos-QSVC, VQC, and hybrid quantum-classical neural networks, evaluated using the ToN_IoT dataset across various model configurations.
- Experimental validation of the top-performing configurations using IBM's noisy quantum simulators, providing a meaningful contribution to the field by assessing QML robustness in noisy environments.
- An in-depth analysis of circuit design choices to determine their impact on noise resilience, offering a practical guidelines for selecting quantum circuit structures optimized for specific noise profiles.

This work is organized into these sections: Section I provides the background, explaining QML models used, and Section II reviews the related work. Section III details the methodology, including data preprocessing and the use of quantum computing tools and noisy simulators. Section IV presents the optimization and performance analysis of Pegasos-QSVC, VQC, and hybrid neural network models. Finally, Section V summarizes the findings and discusses the impact of noise and Section VI concludes and outlines future directions.

I. Background

QML is an emerging discipline that combines machine learning theory with the principles of quantum computing. QML has the potential to enhance algorithm execution speed and improve the management of high-dimensional data, offering significant advancements in both efficiency and scalability.

There are some QML algorithms that have shown good performances in literature [14].

One of particular interest is the Quantum Support Vector Machine (QSVM), a quantum version of the classical Support Vector Machine (SVM) algorithm. In classical SVMs, the kernel trick is used to map data into a higher-dimensional space. Similarly, QSVM applies this concept but transforms the data into a quantum feature space [15], [16]. Another algorithm based on SVM that utilizes the quantum kernel method is the Pegasos-QSVC. Inspired by the Pegasos algorithm (Primal Estimated sub-GrAdient SOlver for SVM) [17], it employs a stochastic gradient descent approach to solve the primal optimization problem of SVMs. Unlike standard SVMs, which require processing the entire dataset at each iteration, Pegasos updates model parameters using only a small subset of the data, thereby reducing computational costs and ensuring training complexity is independent of the training set size.

The VQC is a quantum model inspired by classical neural networks, particularly the multilayer perceptron. It employs a parameterized quantum circuit trained with classical optimization methods to perform classification tasks. As a type of variational quantum algorithm (VQA), the VQC depends heavily on the choice of *ansatz*, the structure of the quantum circuit, which directly affects the algorithm's expressiveness and performance. In this work, we explore four specific ansatzes: Two-Local, Pauli Two-Design, Real Amplitudes, and EfficientSU2, each designed with distinct features and trade-offs suited to various quantum computing problems.

Two-Local is a variational quantum circuit composed of repeated layers of rotations and entanglement. Standard gates commonly used in the literature and defined in the documentation have been applied: R_Y gates for rotations and CX gates for entanglement.

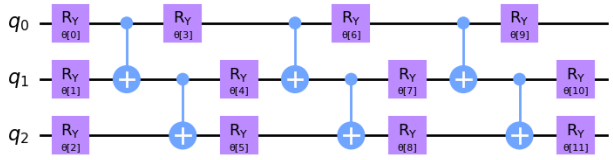

Fig. 1: Example of a Two-Local circuit with 3 qubits and 3 repetitions.

Pauli Two-Design is a variant of the Two-Local. It starts with an initial rotation around the Y-axis with an angle of $\frac{\pi}{4}$. The subsequent layers alternate between rotations around the X, Y, or Z axes and entanglement layers.

Real Amplitudes is also derived from the Two-Local. It generates a trial wavefunction by repeatedly applying rotations around the Y-axis on the qubits and using CX gates for entanglement. As the name suggests, the wavefunction amplitudes are restricted to real values, which can simplify the optimization problem in some cases.

EfficientSU2 is a hardware-efficient variational quantum circuit. The name SU2 refers to the special unitary group

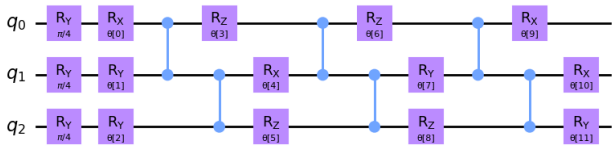

Fig. 2: Example of a Pauli Two-Design circuit with 3 qubits and 3 repetitions.

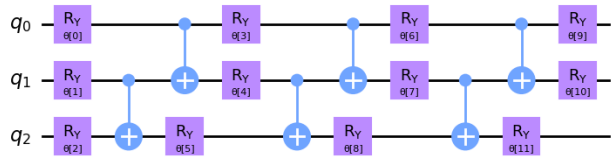

Fig. 3: Example of a Real Amplitudes circuit with 3 qubits and 3 repetitions.

of dimension 2, which includes all 2×2 complex matrices with a determinant equal to 1. Compared to other ansatzes, it uses a larger number of rotation gates, combining R_Y and R_Z rotations in each repetition.

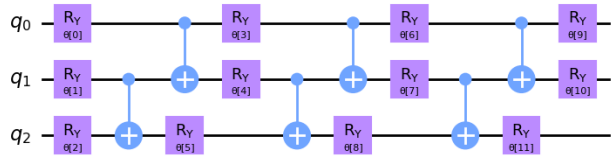

Fig. 4: Example of a EfficientSU2 circuit with 3 qubits and 3 repetitions.

An other model is the hybrid classical-quantum neural network, implemented by integrating a parameterized quantum circuit as a hidden layer.

II. Related Work

As 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 [11], 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 limited insights on noise resilience. A different study [18] 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 CIC-DDoS 2019 dataset has been used in [19], 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 [20] 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 [21], 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 [22] implemented QSVM and VQC with minimal features, reporting accuracies between 60-64% and an F1 score of 45%. The study [23] employed EfficientSU2 and COBYLA optimizer in a VQC model on NSL-KDD, resulting in 90% accuracy in IBM quantum simulators. Further research [24], [25] introduced generative models, though no specific metrics or code were provided.

An hybrid QNN architecture has been introduced in [12], 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 [26], using noise models derived from IBM quantum devices. 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. An other hybrid model was presented in [27], 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 work [28] 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. The study [29] 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 [30] utilized a 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. 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 [31], Abreu et al. evaluated QML models, including VQC, QSVM, QKM, and QCNN, on datasets UNSW-NB15, CIC-IDS17, CICIoT23, and TONIoT. Personalized circuit optimization contributed to improved results, with VQC and QSVM models achieving a 97% F1 score on ToNIoT. 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 [32], 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 [33], together 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 I provides a summary of recent quantum and hybrid quantum-classical models applied to intrusion and anomaly detection. Although these models exhibit strong performance across diverse datasets, their resilience to noise either remains unaddressed or yields limited results.

Therefore, rather than aiming to surpass classical methods, which have already reached peak performance in some cases [34], the goal of this work is to enhance current QML models to be resistant to quantum noise, enabling their future application once hardware technology advances sufficiently and to support more complex data. We systematically tested various quantum circuit configurations to understand which characteristics make them more resilient to noise. By evaluating different quantum circuits across multiple noisy simulators, we identified which configurations are best suited for specific noise profiles. These insights provide valuable guidelines for selecting circuit structures that naturally mitigate noise effects.

III. Methodology

This section outlines the approach taken to prepare the dataset, implement and optimize quantum machine learning models, and evaluate their performance under realistic quantum conditions.

A. Data preprocessing

The dataset was preprocessed by removing irrelevant features, handling missing values, eliminating duplicates, and encoding categorical data. It was split into training and testing sets, min-max scaling normalized values, and PCA reduced dimensionality, ensuring the dataset was optimized for classification and quantum model compatibility.

B. QML models configuration

In this work, we have configured and evaluated three QML models. As our QSVM model, we used the Pegasos-QSVC algorithm. The methodology for hyperparameter optimization is summarized in Figure 5. 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 (reps) were explored to assess their impact on the model's performance. The

Reference	Model / Approach	Dataset(s)	Performance Metrics	Noise Consideration
[11]	20-qubit QNN with 3 hidden layers	CSE-CICIDS2018	Accuracy: 95%, F1 Score: 97%	Not considered
[18]	QNN with rotational encoding on 8 qubits	NF-UNSW-NB15	F1 Score: 0.86	IonQ's quantum computers, native gate optimization
[12]	Hybrid QNN with Dense classical layers	KDD99, CICIDS-2017	Accuracy: 99.81% (KDD99), 98.74% (CICIDS-2017)	Not considered
[26]	Hybrid quantum-classical model	Botnet DGA dataset	Accuracy: 89.7% ($n = 10,000$)	Noise models derived from IBM quantum devices
[27]	Hybrid model with variational adversarial encoder	UNSW-NB15	Accuracy: 95%, F1 Score: 72%	Not considered
[19]	QNN with COBYLA optimization	CIC-DDoS 2019	Accuracy: 92.63% (statevector), 80.69% (noisy computation)	Significant impact of noise
[20]	Quantum Federated Learning (QNN with AngleEmbedding)	NSL-KDD	Accuracy: 98%	Not considered
[21]	Variational QNN for intrusion detection	KDD Cup 99	F1 Score: 98.35% (noiseless), 83.87% (noisy)	Significant impact of noise
[22]	QSVM and VQC with minimal features	KDD Cup 99	Accuracy: 60-64%, F1 Score: 45%	Not specified
[23]	VQC with EfficientSU2 and COBYLA optimizer	NSL-KDD	Accuracy: 90%	Not specified
[24]	qGAN model	NSL-KDD	No specific metrics provided	Not specified
[25]	Generative model for anomaly detection	Unspecified	No performance metrics provided	Not specified
[28]	Quantum autoencoder for dimensionality reduction	UNB NSL-KDD, UNSW NB15	Accuracy: 0.75 (NSL-KDD), 0.93 (UNSW NB15)	Not specified
[29]	Quantum autoencoder with quantum one-class SVM	KDD99, IoT-23, CIC-IoT23	F1 Score: 97%, 96%, 98%	Not specified
[30]	QSVM for DDoS detection on smart micro-grids	Custom	Accuracy: 99.94%	Not considered
[32]	QSVM, QCNN for intrusion detection	Custom data streams	Accuracy: 98%	No noise model specified
[33]	VQNN with convolutional approach	UNSW-NB15	F1 Score: 95.86% (noiseless), 80.62% (noisy)	Significant impact of noise

TABLE I: Comparison of Quantum Machine Learning Models for Intrusion Detection

best-performing configurations were then evaluated on noisy quantum simulators to validate their robustness. 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 Figure 6. The structure of the hybrid quantum-classical neural network is presented in Figure 7. 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.

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.

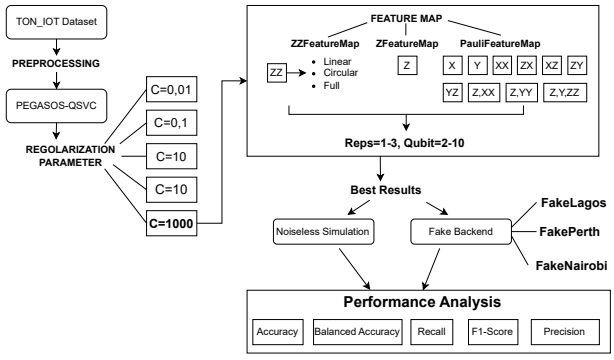

Fig. 5: QSVC configuration optimization methodology

C. Noisy quantum computation

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

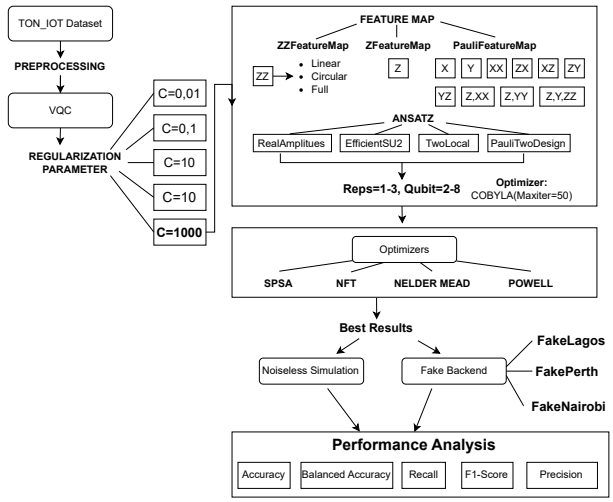

Fig. 6: VQC configuration optimization methodology

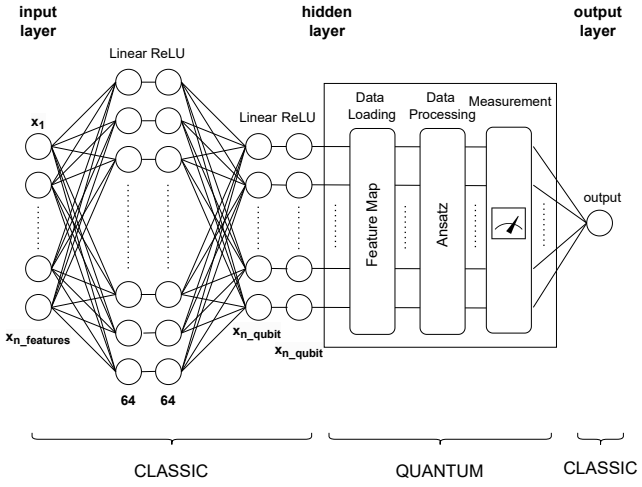

Fig. 7: Hybrid quantum-classical neural network structure

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. These metrics are deeply interconnected. Shorter T1 and T2 times constrain computation time, while high gate and readout errors compound inaccuracies, including relaxation times (T_1), coherence times (T_2), and error rates for operations like quantum gates.

Due to the high computational cost and long queue times associated with running large numbers of configurations, we did not test our proposed quantum machine learning methods on actual devices. Instead, we relied on noisy simulators, which provided a more practical alternative for evaluating several configurations, especially for deep circuits and larger datasets. The selected quantum circuits were tested under noisy conditions using *fake backends* provided by IBM, which 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. Noisy simulation obviously produces different results compared to the ideal one. However, starting with noise-free simulations represents a useful approach, as adding noise is computationally expensive. Ideal simulations thus serve as an initial baseline from which to explore various aspects further.

IV. 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.

A. 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.

1) Regularization Parameter C: An initial element explored is the regularization parameter C of the PegasosQSVC function, which influences the balance between the regularization term and the classification error term in the SGD update rule. As mentioned in [35], a higher C reduces penalties for misclassified points, resulting in wider margins and better generalization but increasing the risk of bias or underfitting. Conversely, a lower C forces the algorithm to fit misclassified points more closely, leading to narrower margins and a risk of overfitting while capturing more complex patterns.

The performance of the Pegasos-QSVC algorithm was analyzed for different values of the regularization parameter C, Table II. Higher C values (1000 and 100) result in strong performance, with metrics exceeding 91% and a balanced F1-Score. 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.

2) Feature Map: Classical data was encoded into quantum data using angle encoding with predefined circuits in Qiskit, including PauliFeatureMap, ZFeatureMap, and ZZFeatureMap. The ZZFeatureMap employed three entanglement types: full, linear, and circular. For the PauliFeatureMap, various rotation combinations (`"X"`, `"Y"`, `"XX"`, `"XZ"`, `"ZX"`,

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%

TABLE II: Pegasos-QSVC performances for several C values

"YZ", "ZY", "Z,YY", "Z,XX", "Z,Y,ZZ") were explored. The number of repetitions, representing how many times encoding operations are repeated in the circuit, was analyzed at 1, 2, and 3 repetitions to balance pattern complexity and hardware noise. Additionally, experiments varied the number of qubits (2 to 10) to examine the model's performance across different configurations.

3) Execution results: The total number of experimental results amounts to 378, derived from testing configurations with 2 to 10 qubits, 14 feature maps, and 1 to 3 repetitions for each feature map. The tests were conducted initially on a noiseless simulator and subsequently on a noisy simulator to optimize the number of tests. The Table III shows the configurations of the models which have obtained the best performances.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 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.

4) Noisy simulation: 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 qubits architecture, they differ in their associated error levels. The selection aimed to explore various noise conditions, as each fake backend introduces different types and intensities of errors. Specifically, FakeLagosV2, as shown in Tables VI, IX, was chosen for its good single-gate error probability but high two-qubit gate error and readout error, whereas FakeNairobiV2, Tables V , VIII, and FakePerth, Tables IV, VII, 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.

The QSVC models have been tested on these noisy simulators, Table X. The results show a decline in performance when noise is introduced. Specifically, the highest value in the ideal simulation, obtained with PauliFeatureMap[Z, XX] using 2 repetitions and 6 qubits, becomes the worst when noise is applied. In contrast, the ZFeatureMap shows good performance even in the presence of noise, maintaining high results across all three noisy models. 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 PauliFeatureMap [ZX].

B. VQC

The 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*. 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, for a total of 1176 tests.

From the obtained results, the best performance is achieved with 3–4 qubits, unlike the Pegasos-QSVC model, where, as previously mentioned, 5–6 qubits are required. This outcome was expected since the VQC is significantly more complex, combining a feature map, a parameterized circuit, and varying repetitions. However, in many cases, good results are also obtained with 5 or 6 qubits, highlighting how different combinations and configurations can yield varied behaviors.

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.

Best FeatureMap	Accuracy	Precision	Recall	F1 Score	Balanced Accuracy
q=6, ZZFeatureMap (Linear), r=1	94,12%	96,10%	96,01%	96,05%	92,29%
q=6, PauliFeatureMap [ZX], r=1	94,16%	96,04%	96,13%	96,09%	92,26%
q=4, PauliFeatureMap [Z,YY], r=1	93,17%	93,91%	97,15%	95,50%	89,34%
q=7, ZFeatureMap, r=2	95,44%	96,93%	96,96%	96,94%	93,98%
q=5, PauliFeatureMap [XX], r=2	94,14%	96,22%	95,91%	96,06%	92,43%
q=6, PauliFeatureMap [Z,XX], r=2	95,61%	97,11%	96,99%	97,05%	94,27%
q=4, PauliFeatureMap [Z,Y,ZZ], r=2	94,57%	95,89%	96,86%	96,37%	92,35%
q=5, ZZFeatureMap (Linear), r=3	94,35%	95,65%	96,83%	96,24%	91,96%

TABLE III: Pegasos-QSVC configurations with best performances

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

TABLE IV: T1, T2, and Readout Error values for each qubit in FakePerth.

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

TABLE V: T1, T2, and Readout Error values for each qubit in FakeNairobi.

The average performance for each ansatz improves as the number of repetitions increases. However, while performance increases with r for all ansatzes, the improvement is not uniform. Figure 8 summerizes the results on the ansatzes used for the reps 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 XI.

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 XI. Conversely, the *ZZFeatureMap*, in all its entanglement variants, exhibits superior performance, with higher average and maximum values compared to other configurations. 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.

2) Optimizer analysis: The choice of optimizer plays a crucial role in determining the efficiency of a variational quantum algorithm.In addition to *COBYLA*, other optimizers were considered in this work. As shown in Table XII, the

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 VI: T1, T2, and Readout Error values for each qubit in FakeLagos.

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

TABLE VII: RX, SX, X, CX gate errors for each qubit or couple of qubits in FakePerth.

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. 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.

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 VIII: 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	0.00016
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 IX: RX, SX, X, CX gate errors for each qubit or couple of qubits in FakeLagos.

3) Noisy simulation: Table XIII summarizes the performance of VQC on Fake Nairobi, Fake Lagos, and Fake Perth. The configuration 3 QUBIT, PauliFeatureMap [Y], RealAmplitudes(REPS=2), SPSA consistently delivers the best overall results, achieving high accuracy, precision, recall, and F1-scores across all noise models, with accuracy above 91% and F1-Score up to 94.19%. In contrast, the configuration 5 QUBIT, PauliFeatureMap [Z, YY], EfficientSU2 (REPS=2), SPSA shows significant performance drops, especially on Fake Lagos and Fake Perth, with balanced accuracies falling to 50.00%. 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.

Configuration	Fake Nairobi	Fake Lagos	Fake Perth
6 QUBIT, PauliFeatureMap [ZX], reps=1	Acc: 93,03% Precision: 95,42% Recall: 97,17% F1-Score: 93,71% Acc B: 89,05%	Acc: 88,10% Precision: 93,31% Recall: 90,62% F1-Score: 91,95% Acc B: 85,79%	Acc: 88,10% Precision: 93,31% Recall: 90,62% F1-Score: 91,95% Acc B: 85,79%
4 QUBIT, PauliFeatureMap [Z,YY], reps=1	Acc: 83,25% Precision: 86,42% Recall: 91,98% F1-Score: 89,12% Acc B: 74,82%	Acc: 91,07% Precision: 91,44% Recall: 97,12% F1-Score: 94,19% Acc B: 85,23%	Acc: 91,54% Precision: 91,99% Recall: 97,15% F1-Score: 94,50% Acc B: 86,18%
5 QUBIT, ZFeatureMap, reps=1	Acc: 93,05% Precision: 95,43% Recall: 97,18% F1-Score: 93,73% Acc B: 89,07%	Acc: 93,05% Precision: 95,43% Recall: 97,18% F1-Score: 93,73% Acc B: 89,07%	Acc: 93,05% Precision: 95,43% Recall: 97,18% F1-Score: 93,73% Acc B: 89,07%
7 QUBIT, ZFeatureMap, reps=2	Acc: 82,37% Precision: 92,28% Recall: 83,33% F1-Score: 87,58% Acc B: 81,45%	Acc: 90,43% Precision: 90,23% Recall: 97,75% F1-Score: 93,84% Acc B: 83,37%	Acc: 82,37% Precision: 92,28% Recall: 83,33% F1-Score: 87,58% Acc B: 81,45%
5 QUBIT, PauliFeatureMap [XX], reps=2	Acc: 91,28% Precision: 91,69% Recall: 97,15% F1-Score: 94,34% Acc B: 85,67%	Acc: 89,84% Precision: 96,23% Recall: 89,89% F1-Score: 92,96% Acc B: 89,79%	Acc: 82,42% Precision: 91,02% Recall: 84,79% F1-Score: 87.80% Acc B: 80,14%
6 QUBIT, PauliFeatureMap [Z,XX], reps=2	Acc: 74,56% Precision: 74,56% Recall: 100% F1-Score: 85,42% Acc B: 50,00%	Acc: 74,56% Precision: 74,56% Recall: 100% F1-Score: 85,42% Acc B: 50,00%	Acc: 74,56% Precision: 74,56% Recall: 100% F1-Score: 85,42% Acc B: 50,00%
5 QUBIT, ZZFeatureMap (Linear), reps=3	Acc: 83,25% Precision: 86,42% Recall: 91,98% F1-Score: 89,12% Acc B: 74,82%	Acc: 92,11% Precision: 94,83% Recall: 97,08% F1-Score: 92,68% Acc B: 87,31%	Acc: 83,25% Precision: 86,42% Recall: 91,98% F1-Score: 89,12% Acc B: 74,82%
4 QUBIT, PauliFeatureMap [Z,Y,ZZ], reps=3	Acc: 83,25% Precision: 86,42% Recall: 91,98% F1-Score: 89,12% Acc B: 74,82%	Acc: 83,25% Precision: 86,42% Recall: 91,98% F1-Score: 89,12% Acc B: 74,82%	Acc: 83,25% Precision: 86,42% Recall: 91,98% F1-Score: 89,12% Acc B: 74,82%

TABLE X: Pegasos-QSVC performance of best configurations on Fake Nairobi, Fake Lagos and Fake Perth

C. Hybrid Classical-Quantum Neural Network

1) Structure: The generic structure is described in Figure 7. 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. The hybrid network was optimized using the Adam optimizer.

2) Results: From the results obtained with a noiseless simulator in Table XIV, it emerges that the use of simpler circuits contributes to optimizing the performance of the hybrid network. With 3 qubits, a good performance value was achieved was with an accuracy of 93.1%. However, transitioning from 3 to 5 qubits, combined with a lower number of repetitions in the circuit, led to the overall best performance of the model, with 95.4% of accuracy and 96.9% of F1-score. Analyzing this optimal configuration under noisy conditions, reported in Table XV, 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

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 XI: Best accuracy for each ansatz

Best configurations	COBYLA	SPSA	NFT	NELDER MEAD	POWELL
q=3, ZZ (Linear), EfficientSU2, r=2	91,23%	89,10%	90,10%	85,18%	87,43%
q=3, Pauli [Y], RealAmplitudes, r=2	91,54%	91,47%	91,54%	88,42%	83,06%
q=5, Pauli [Z, YY], EfficientSU2, r=2	91,11%	92,79%	90,17%	81,73%	86,29%
q=3, ZZ (Circular), EfficientSU2, r=2	90,81%	92,13%	94,18%	94,21%	89,84%
q=3, Pauli [Z,Y,ZZ], EfficientSU2, r=3	90,26%	93,00%	93,60%	92,98%	93,57%
q=4, ZZ (Full), RealAmplitudes, r=3	92,48%	81,43%	83,25%	85,99%	82,37%

TABLE XII: Best VQC configuration with several optimizers

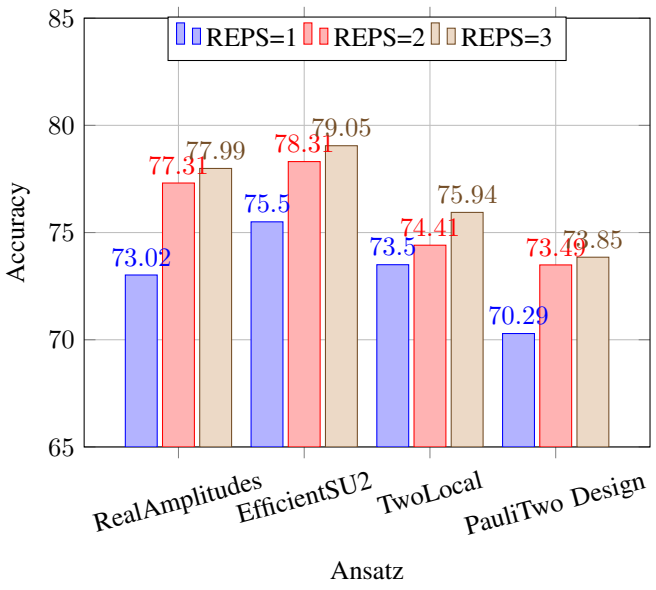

Fig. 8: Ansatz average performance for r=1, r=2, r=3.

Configuration	Fake Nairobi	Fake Lagos	Fake Perth
3 QUBIT, ZZFeatureMap(Linear), EfficientSU2 (REPS=2), COBYLA	Acc: 82,42% Precision: 91,02% Recall: 84,79% F1-Score: 87,80% Acc B: 80,14%	Acc: 78,76% Precision: 90,73% Recall:79,66% F1-Score: 84,83% Acc B: 77,90%	Acc: 90,95% Precision: 96,54% Recall: 91,13% F1-Score: 93,76% Acc B: 90,78%
3 QUBIT, PauliFeatureMap [Y], RealAmplitudes(REPS=2), COBYLA	Acc: 90,86% Precision: 90,97% Recall: 97,40% F1-Score: 94,08% Acc B: 84,54%	Acc: 91,57% Precision: 91,99% Recall: 97,15% F1-Score: 94,50% Acc B: 86,18%	Acc: 90,43% Precision: 90,23% Recall: 97,75% F1-Score: 93,84% Acc B: 83,37%
3 QUBIT, PauliFeatureMap [Y], RealAmplitudes(REPS=2), SPSA	Acc: 91,05% Precision: 91,44% Recall: 97,08% F1-Score: 94,18% Acc B: 85,22%	Acc: 91,07% Precision: 91,44% Recall: 97,12% F1-Score: 94,19% Acc B: 85,23%	Acc: 91,05% Precision: 91,44% Recall: 97,08% F1-Score: 94,18% Acc B: 85,22%
5 QUBIT, PauliFeatureMap [Z, YY], EfficientSU2 (REPS=2), SPSA	Acc: 84,17% Precision: 87,87% Recall: 91,38% F1-Score: 89,59% Acc B: 77,21%	Acc: 74,56% Precision: 74,56% Recall: 100% F1-Score: 85,42% Acc B: 50,00%	Acc: 74,56% Precision: 74,56% Recall: 100% F1-Score: 85,42% Acc B: 50,00%
3 QUBIT, ZZFeatureMap(Circular), EfficientSU2(REPS=3), NELDER_MEAD	Acc: 88,16% Precision: 93,31% Recall: 90,62% F1-Score: 91,95% Acc B: 85,79%	Acc: 74,56% Precision: 74,56% Recall: 100% F1-Score: 85,42% Acc B: 50,00%	Acc: 74,56% Precision: 74,56% Recall: 100% F1-Score: 85,42% Acc B: 50,00%

TABLE XIII: VQC performance of best configurations on Fake Nairobi, Fake Lagos and Fake Perth

due to greater sensitivity to errors in two-qubit gates given PauliFeatureMap ["ZY"] and EfficientSU2, while FakePerth maintains a consistent level of performance, achieving an accuracy of 94.2% and an F1-score of 92.4%.

V. Final Analysis and Discussion

Best results for each QML algorithm are summarized in Table XVI and Table XVII respectively for noiseless and noisy simulations. Pegasos-QSVC stands out for achieving the best overall performance than VQC. This result highlights the efficiency of simpler, optimization-focused approaches when compared to more complex variational models. VQC also delivers strong results in both noiseless and noisy simulations, with competitive accuracy and F1-scores. However, their performance is highly sensitive to parameters like qubit count, circuit depth, and optimization methods, requiring systematic testing to identify optimal configurations. The hybrid neural network performs well in noiseless simulations, showcasing the benefits of combining classical and quantum paradigms for robust and efficient models. However, these results do not always translate to noisy environments, where performance can drop significantly for certain backends. Despite these challenges, the model achieved 95.44% accuracy and an F1 Score of 96.94% on the FakePerth backend, showcasing its robustness even under some noisy conditions.

In general, the performance of QSVC, VQC, and hybrid quantum-classical networks is heavily influenced by the noise characteristics of the simulated quantum backends. For FakeLagos exhibits greater sensitivity to errors in two-qubit gates, leading to poorer performance with highly entangled feature maps like some PauliFeatureMap and complex variational forms like EfficientSU2, but good performance with simpler feature maps and ansatz. In contrast, FakeNairobi

Configuration	Noiseless simulation
3 QUBIT, PauliFeatureMap [”Z”, ”YY”], RealAmplitudes (reps=3)	Acc: 83,7% Precision: 90,3% Recall: 68,1% F1-Score: 71,6%
3 QUBIT, PauliFeatureMap [”YZ”], EfficientSU2 (reps=2)	Acc: 93,1% Precision: 92,3% Recall: 89,1% F1-Score: 90,5%
3 QUBIT, PauliFeatureMap [”Z”, ”YY”], EfficientSU2 (reps=2)	Acc: 93,1% Precision: 92,3% Recall: 89,1% F1-Score: 90,5%
3 QUBIT, PauliFeatureMap [”ZY”], TwoLocal (reps=3)	Acc: 74,6% Precision: 37,3% Recall: 50,0% F1-Score: 42.7%
5 QUBIT, PauliFeatureMap [”Y”], RealAmplitudes (reps=3)	Acc: 93,0% Precision: 92,2% Recall: 89,1% F1-Score: 90,5%
3 QUBIT, PauliFeatureMap [”ZY”], EfficientSU2 (reps=2)	Acc: 93.1% Precision: 92.2% Recall: 89.1% F1-Score: 90.5%
5 QUBIT, PauliFeatureMap [”ZY”], EfficientSU2 (reps=1)	Acc: 95,4% Precision: 96,9% Recall: 96,9% F1-Score: 96,9%
5 QUBIT, PauliFeatureMap[”Z”, ”YY”], EfficientSU2 (reps=2)	Acc: 88,5% Precision: 88,5% Recall: 88,1% F1-Score: 83,1%
6 QUBIT, ZZFeatureMap(Full), EfficientSU2 (reps=2)	Acc: 74,6% Precision: 37,3% Recall: 50,0% F1-Score: 42.7%

TABLE XIV: Hybrid quantum-classical network noiseless performance on best configurations

Configuration	Fake Nairobi	Fake Lagos	Fake Perth
5 QUBIT, PauliFeatureMap [”ZY”], EfficientSU2 (reps=1)	Acc: 90.0% Prec.: 89.9% Rec.: 83.4% F1-Sc.: 85.9%	Acc: 86.8% Prec.: 87.2% Rec.:77.0% F1-Sc.: 80.3%	Acc: 94.2% Prec.: 93.1% Rec.: 91.7% F1-Sc.: 92.4%

TABLE XV: Hybrid quantum-classical network performance of the best configuration on Fake Nairobi, Fake Lagos and Fake Perth

and FakePerth handle noise more effectively, showing better performance on circuits requiring significant entanglement, such as ZZFeatureMap. The results suggest that once the real quantum machine for training is selected, it is useful to assess the machine's resistance to two-qubit gate errors in order to decide the circuit structure to execute. Specifically, machines more sensitive to these errors may require circuits with less entanglement, while machines less prone to errors can handle more complex circuits with greater entanglement.

There is also a significant advancement in the state of the art for purely quantum machine learning in anomaly detection under noisy conditions using Pegasos-QSVC with 5 qubits, a ZFeatureMap, and reps=1. This configuration achieved an accuracy of 93.05% and an F1-score of 93.73%, surpassing the current state of the art, which reports an accuracy of 89.7% and an F1-score of 86%.

Configuration	Performance
PegasosQSVC - 6 QUBIT, PauliFeatureMap [Z,XX], reps=2)	Accuracy: 95,61% Precision: 97,11% Recall: 96,99% F1 Score: 97,05%
VQC - 3 QUBIT, ZZFeatureMap(Circular), EfficientSU2(REPS=3), NELDER_MEAD	Accuracy: 94,21% Precision: 96,46% Recall: 95,75% F1 Score: 96,10%
Hybrid Quantum-Classical Neural Network - PauliFeatureMap [Y], 5 QUBIT, reps=1, ADAM	Accuracy: 95.44% Precision: 96.96% Recall: 96.93% F1 Score: 96.94%

TABLE XVI: Best configuration for each model without noise

Configuration	Performance
PegasosQSVC - 5 QUBIT, ZFeatureMap, (reps=1)	Accuracy: 93,05% Precision: 95,43% Recall: 97,18% F1 Score: 93,73%
VQC - 3 QUBIT, PauliFeatureMap [Y], RealAmplitudes(reps=2), COBYLA	Accuracy: 91,57% Precision: 91,99% Recall: 97,15% F1 Score: 94,50%
Hybrid Quantum-Classical Neural Network - PauliFeatureMap [Y], 5 QUBIT, reps=1, ADAM	Accuracy: 94.2% Precision: 93.1% Recall: 91.7% F1-Score: 92.4%

TABLE XVII: Best configuration for each model with noise

VI. Conclusion and Future Work

In this work, three QML models, QSVM, VQC, and a custom hybrid quantum-classical neural network, have been analyzed for their application in intrusion detection in networks. The approach presented includes fine-tuning various hyperparameters, such as the number of qubits, repetitions, feature maps, ansatz structures, and optimizers. The best configurations were further tested on IBM's noisy simulators to evaluate the models' effectiveness under realistic conditions. Pegasos-QSVC reaches an accuracy of 93.05% and the hybrid quantum-classical model reaches an accuracy of 94.2%. These results highlight the potential of QML in tackling real-world challenges in intrusion detection, even in environments where noise significantly impacts performance. Future research could focus on expanding the range of datasets, enhancing the understanding of QML's scalability and applicability. Moreover, 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.

Acknowledgment

This research was funded by the NGIsargasso project (Europe Horizon Grant No. 101092887), Open Call 4 FRQ-GAN4AD project.

References

[1] F. Cirillo and C. Esposito, “Practical evaluation of a quantum physical unclonable function and design of an authentication scheme,” in *2024 IEEE International Conference on Quantum Computing and Engineering (QCE)*, vol. 1. IEEE, 2024, pp. 1354–1363.

[2] ——, "A qpuf-based scheme for secure and adaptable quantum device attestation in nisq devices," in *2025 International Conference on Quantum Communications, Networking, and Computing (QCNC)*. IEEE, 2025, pp. 117–121.

[3] V. H. Kothavade, S. J. G. Passo, and J. J. Prevost, "Quantum-inspired clustering techniques for malware detection in supply chain networks," in *2025 Congress in Computer Science, Computer Engineering, & Applied Computing (CSCE)*. IEEE, 2025.

[4] A. M. AL-Hawamleh, "Predictions of cybersecurity experts on future cyber-attacks and related cybersecurity measures," *momentum*, vol. 3, no. 14, p. 15, 2023.

[5] A. Thakkar and R. Lohiya, "A review on challenges and future research directions for machine learning-based intrusion detection system," *Archives of Computational Methods in Engineering*, vol. 30, no. 7, pp. 4245–4269, 2023.

[6] S. Muneer, U. Farooq, A. Athar, M. Ahsan Raza, T. M. Ghazal, and S. Sakib, "A critical review of artificial intelligence based approaches in intrusion detection: A comprehensive analysis," *Journal of Engineering*, vol. 2024, no. 1, p. 3909173, 2024.

[7] D. Peral-García, J. Cruz-Benito, and F. J. García-Peñalvo, "Systematic literature review: Quantum machine learning and its applications," *Computer Science Review*, vol. 51, p. 100619, 2024.

[8] F. Cirillo and C. Esposito, "Intrusion detection system based on quantum generative adversarial network," in *Proceedings of the 17th International Conference on Agents and Artificial Intelligence - Volume 1: QAIO*, INSTICC. SciTePress, 2025, pp. 830–838.

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

[10] S. Misra and P. Rani, "Quantum machine learning: A comprehensive overview and analysis," in *2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT)*. IEEE, 2024, pp. 1–5.

[11] P. Bhattacharya, A. Kumari, S. Tanwar, I. Budhiraja, S. Patel, and J. J. Rodrigues, "Quant-Jack: Quantum Machine Learning to Detect Cryptojacking Attacks in IIoT Networks," in *2024 IEEE International Conference on Communications Workshops (ICC Workshops)*, Jun. 2024, pp. 865–870, iSSN: 2694-2941.

[12] K. Gautam, G. Usha, A. Nagar, S. Patel, and A. Jain, "Quantum assisted machine learning for intrusion detection systems," Kattankalathur, India, 2024, p. 020280.

[13] UNSW Sydney, "TONIOT Datasets," 2024.

[14] S. Raubitzek and K. Mallinger, "On the applicability of quantum machine learning," *Entropy*, vol. 25, no. 7, 2023.

[15] S. J. G. Passo and J. J. Prevost, "Characterization of quantum computers for optimal quantum machine learning in brain tumor classification," in *Quantum Engineering and Technology Conference (QET 2025)*. IET, 2025.

[16] S. J. G. Passo, V. H. Kothavade, and J. J. Prevost, "Supply chain malware detection via classical and quantum kernel methods in embedded systems," in *2025 55th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)*. IEEE, 2025.

[17] S. Shalev-Shwartz, Y. Singer, and N. Srebro, "Pegasos: Primal estimated sub-gradient solver for svm," in *Proceedings of the 24th International Conference on Machine Learning*, ser. ICML '07. New York, NY, USA: Association for Computing Machinery, 2007, p. 807–814.

[18] A. Kukliansky, M. Orescanin, C. Bollmann, and T. Huffmire, "Network Anomaly Detection Using Quantum Neural Networks on Noisy Quantum Computers," *IEEE Transactions on Quantum Engineering*, vol. 5, pp. 1–11, 2024, conference Name: IEEE Transactions on Quantum Engineering.

[19] M. Y. Küçükkara, F. Atban, and C. Bayılmış, "Quantum-neural network model for platform independent ddos attack classification in cyber security," *Advanced Quantum Technologies*, p. 2400084, 2024.

[20] Z. A. E. Houda, H. Moudoud, B. Brik, and M. Adil, "A Privacy-Preserving Framework for Efficient Network Intrusion Detection in Consumer Network Using Quantum Federated Learning," *IEEE Transactions on Consumer Electronics*, pp. 1–1, 2024, conference Name: IEEE Transactions on Consumer Electronics.

[21] C. Gong, W. Guan, A. Gani, and H. Qi, "Network attack detection scheme based on variational quantum neural network," *The Journal of Supercomputing*, vol. 78, no. 15, pp. 16 876–16 897, Oct. 2022.

[22] P. Venkatachalam and D. Q. Liu, "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)*, Nov. 2023, pp. 410–416, iSSN: 2833-8898.

[23] M. A. Rahman, M. S. Akter, E. Miller, B. Timofti, H. Shahriar, M. Masum, and F. Wu, "Fine-Tuned Variational Quantum Classifiers for Cyber Attacks Detection Based on Parameterized Quantum Circuits and Optimizers," in *2024 IEEE 48th Annual Computers, Software, and Applications Conference (COMPSAC)*, 2024, pp. 1067–1072.

[24] M. A. Rahman, H. Shahriar, V. Clincy, M. F. Hossain, and M. Rahman, "A Quantum Generative Adversarial Network-based Intrusion Detection System," in *2023 IEEE 47th Annual Computers, Software, and Applications Conference (COMPSAC)*. Torino, Italy: IEEE, Jun. 2023, pp. 1810–1815.

[25] H. Tezuka, S. Uno, and N. Yamamoto, "Generative model for learning quantum ensemble with optimal transport loss," *Quantum Machine Intelligence*, vol. 6, no. 1, p. 6, Jun. 2024.

[26] H. Suryotrisongko and Y. Musashi, "Evaluating hybrid quantum-classical deep learning for cybersecurity botnet DGA detection," *Procedia Computer Science*, vol. 197, pp. 223–229, 2022.

[27] Z. Liu, X. Jia, and B. Li, "E-healthcare application cyber security analysis using quantum machine learning in malicious user detection," *Optical and Quantum Electronics*, vol. 56, no. 3, Mar. 2024.

[28] A. Gouveia and M. Correia, "Towards Quantum-Enhanced Machine Learning for Network Intrusion Detection," in *2020 IEEE 19th International Symposium on Network Computing and Applications (NCA)*, Nov. 2020, pp. 1–8, iSSN: 2643-7929.

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

[30] D. Said, "Quantum Computing and Machine Learning for Cybersecurity: Distributed Denial of Service (DDoS) Attack Detection on Smart Micro-Grid," *Energies*, vol. 16, no. 8, p. 3572, Apr. 2023.

[31] D. Abreu, D. Moura, C. Rothenberg, and A. Abelém, "QuantumNetSec: Quantum Machine Learning for Network Security," Sep. 2024.

[32] M. Kalinin and V. Krundyshev, "Security intrusion detection using quantum machine learning techniques," *Journal of Computer Virology and Hacking Techniques*, vol. 19, no. 1, pp. 125–136, Jun. 2022.

[33] C. Gong, W. Guan, H. Zhu, A. Gani, and H. Qi, "Network intrusion detection based on variational quantum convolution neural network," *The Journal of Supercomputing*, vol. 80, no. 9, pp. 12 743–12 770, Jun. 2024.

[34] P. Jayalaxmi, G. Kumar, R. Saha, M. Conti, T.-h. Kim, and R. Thomas, "DeBot: A deep learning-based model for bot detection in industrial internet-of-things," *Computers and Electrical Engineering*, vol. 102, p. 108214, 2022.

[35] R. Bhavsar, N. K. Jadav, U. Bodkhe, R. Gupta, S. Tanwar, G. Sharma, P. N. Bokoro, and R. Sharma, "Classification of potentially hazardous asteroids using supervised quantum machine learning," *IEEE Access*, vol. 11, 2023.