Quantum Adversarial Attacks: Developing Quantum FGSM Algorithm

Mst Shapna Akter*, Hossain Shahriar†, Alfredo Cuzzocrea‡§, Fan Wu¶
*Dept. of Intelligent and Robotics Systems, University of West Florida, USA. Email: jannatul.shapna99@gmail.com
†Center for CyberSecurity, University of West Florida, USA. Email: hshahria@kennesaw.edu
‡iDEA Lab, University of Calabria, Rende, Italy.
§Dept. of Computer Science, University of Paris City, Paris, France. Email: alfredo.cuzzocrea@unical.it
¶Dept. of Computer Science, Tuskegee University, Tuskegee, USA. Email: fwu@tuskegee.edu

***Abstract*—Quantum machine learning, noted for its remarkable advancements in enhancing computational speed and augmenting data processing efficacy, is acquiring considerable recognition within the scientific community. Despite the noteworthy advancements of quantum machine learning, it shares with its traditional counterpart a susceptibility to adversarial threats. This presents a significant challenge and underscores the need for robust countermeasures in this emerging field of study. Developing effective adversarial attack algorithms, such as the Quantum Fast Gradient Sign Method (QFGSM), is crucial to opening the path for exploring robust defense mechanisms against these threats. However, unlike traditional machine learning, the quantum field currently lacks the effective techniques to orchestrate adversarial attacks. This study introduces and assesses the QFGSM, a adversarial attack algorithm tailored for QNNs. Additionally, we evaluate the effect of random noise and its quantum version on these models, providing a comprehensive comparison to understand the efficacy of adversarial attack methods. The evaluation is conducted on both traditional Neural Networks (NNs) and QNNs using the ClaMP dataset, a cybersecurity-focused dataset, and involves performance metrics like Accuracy, Precision, Recall, and F1 score. Our findings underscore the differential resilience of NNs and QNNs under adversarial attacks and reveal the contrasting effects of FGSM, QFGSM, and random noise-based methods. Our study reveals that Quantum Neural Networks (QNNs) show significant vulnerability to our proposed Quantum Fast Gradient Sign Method (QFGSM) compared to random noise, indicating a need for quantum-specific defenses in the face of this advanced adversarial attack algorithm.**

Adversarial Attack, Quantum Neural Network (QNN), Neural Network (NN), FGSM, QFGSM

I. Introduction

In recent years, machine learning models become increasingly prevalent across various sensitive sectors, including healthcare [1], cybersecurity [2], autonomous vehicles [3], and other sectors[4–7]. It is crucial to comprehend and combat the potential adversarial vulnerabilities associated with these advancements. Adversarial attacks typically involve the introduction of subtly altered inputs designed to trick the model into erroneous predictions, posing serious threats to the model's reliability and integrity. The training process for quantum machine learning follows a similar approach to classical machine learning, with the distinction that qubits are employed instead of classical bits. However, it is important to note that despite this quantum framework, perturbations can still be leveraged on the input data to induce adversarial attacks. There are several effective techniques exist such as Fast Gradient Sign Method (FGSM), Basic Iterative Method (BIM), DeepFool (DF), and Jacobian-based Saliency Map Attack (JSMA). FGSM is a well-established method used to generate adversarial examples in the classical machine learning realm. This technique leverages the gradients of the model's loss function to formulate perturbations, leading to increased loss, and consequently, a decrease in model performance. In contrast, such a technique remains largely unexplored in the quantum realm. Our research, thus, not only investigates the resilience of NNs and QNNs to adversarial attacks, but also introduces the quantum fast gradient sign method (QFGSM), an adversarial attack algorithm designed specifically for quantum machine learning models. Alongside the conception and deployment of QFGSM, our study involves a comparison of the impacts of random noise and its quantum equivalent on the performance of NNs and QNNs in the cybersecurity field. This effort is aimed at providing a thorough evaluation of the effectiveness of adversarial attack methodologies. Our findings, derived from comparative evaluations using key performance metrics such as accuracy, precision, recall, and F1 score, demonstrate that the QFGSM methodology results in a significant decline in the accuracy of the QNN model following the adversarial attack. Our work contributes in two distinct ways: first, we develop and implement the Quantum Fast Gradient Sign Method (QFGSM), an adversarial attack algorithm designed explicitly for Quantum Neural Networks (QNNs). This represents a significant step forward in understanding the vulnerabilities of quantum machine learning models, bridging the gap in knowledge regarding the susceptibilities of these models to adversarial attacks. Second, our research delivers a comprehensive evaluation of the impact of random and quantum noise on both classical Neural Networks (NNs) and QNNs. This comparative study provides solid information on the differential effects of noise types on these machine learning models, furthering our understanding of their resilience and vulnerabilities.

II. Related Work

Several studies have delved into quantum machine learning and adversarial attacks. Lu et al. [8] explored adversarial attacks on quantum classifiers but did not develop a quantum version of

2836-3795/24/$31.00 ©2024 IEEE
DOI 10.1109/COMPSAC61105.2024.00145

the FGSM method. Du et al. [9] addressed the vulnerability of quantum classifiers, touching on the iterative-fast gradient sign method (I-FGSM) without creating a quantum-specific FGSM. West et al. [10] benchmarked the robustness of quantum ML networks against adversarial attacks, including FGSM, but did not propose a quantum FGSM version. Suryotrisongko et al. [11] researched the adversarial vulnerability of hybrid quantum deep learning models and showed a 5.9% accuracy gain through adversarial training. They employed FGSM, among other attacks, but did not suggest a quantum-specific FGSM. In essence, while the principles of FGSM have been adapted for quantum settings, a quantum-specific FGSM, potentially termed QFGSM, remains unexplored, presenting a promising avenue for future research.

III. Methodology

A. Dataset

The dataset named "ClaMP" [12] used in this study consists of a total of 5,184 samples, including both malware and benign instances. The malware samples were collected from the virus share dataset. The benign samples were obtained from Windows XP and Windows 7 program files. Additionally, some benign samples were collected from online free software archives. The data set comprises 69 features, which can be categorized into two types: raw features and derived features. The raw features encompass 54 variables that directly extract values from specific fields within the Portable Executable (PE) header of executable files. The PE format is a common file format used in Windows operating systems to store executable programs and libraries. Various fields of the PE header were utilized to discriminate between malware and benign files. Specifically, fields such as NumberOfSections, SizeOfInitializedData, NumberOfSymbols, etc., were observed to exhibit larger deviations between malware and benign files. These field values were found to be informative in distinguishing between the two classes, as they demonstrated significant variations in statistical properties, such as mean and standard deviation. In addition to raw features, derived features were generated by validating the raw values against a set of predefined rules. These rules were derived from guidelines provided for PE header fields. There exists a mild imbalance between the categories, with Category 1 representing 52. 2% and Category 0 comprising 47.8% of the total. Although this skew is not severe, it does imply a moderate imbalance within the dataset. This imbalance, while notable, does not substantially affect the validity or reliability of the experimental analyzes performed in this study. However, while performing the experiments, we ensured balanced sampling by selecting different proportions from the total data set.

B. Experimental Work

In this study, we utilize a Quantum Neural Network (QNN), a derivative of Quantum Machine Learning (QML), and apply it to the ClaMP dataset [12]. The initial step involves preprocessing the raw data to prepare it as input for the QNN model. This preprocessing was done using Python and Scikit-Learn's (sklearn) shuffle function. Additionally, we employed the reset index and drop functions from the Python library, as well as the encoder from the sklearn library for labeling. To ensure prediction accuracy, we took measures to maintain class balance within the dataset, specifically focusing on balanced sections for the experimental phase. After applying the shuffle function, we reset the index to sequentially organize the dataset. The drop function was utilized to eliminate irrelevant columns that do not contribute to the prediction process. Given that quantum machine learning models require numerical input, categorical values were transformed into numerical ones, and all numerical values were normalized to ensure a uniform scale. Given the limited number of available qubits for quantum computations, we faced a challenge in processing high-dimensional data. To resolve this issue, we utilized Principal Component Analysis (PCA), a common dimensionality reduction technique. The original dataset consisted of 69 features, which we successfully reduced to 2 principal components using PCA. Despite the significant reduction in dimensionality, the PCA transformation ensured that the essential patterns and features in the data were preserved. This is because PCA operates by projecting the original data onto a new subspace where the variance (or information content) is maximized. Hence, the two principal components that we ended up with represented the directions in the high-dimensional space along which the original data varied the most. We divided the entire dataset, consisting of 5,210 rows, into 20 distinct sections. Each section started at 5 percent of the total dataset, incrementally increasing by 5 percent up to the full 100 percent. Subsequently, the quantum machine learning model was applied to each divided dataset. Before input into the QML model, features were encoded into quantum states. Furthermore, each portion of the dataset was subjected to perturbation using both random noise and FGSM (Fast Gradient Sign Method) techniques, followed by QFGSM for the QNN model. This served to test the robustness of both the neural network and quantum neural network models. Initially, we introduced clean data to both models, subsequently adding perturbed data to the same proportion to assess the robustness. A comparative performance analysis was carried out between QML and traditional machine learning, based on experimental results with different portions of the dataset. A variational classifier is a type of hybrid quantum-classical machine learning model that combines the power of quantum computation with classical optimization techniques to solve classification tasks. A quantum circuit is a series of quantum gates and operations that manipulate qubits, the fundamental units of quantum computation. Given a dataset $X \in \mathbb{R}^{n \times m}$, where n is the number of data samples and m is the number of features, we first preprocess the data by encoding it into the quantum state of m qubits using angle encoding scheme. The algorithm for building this quantum circuit is outlined in Algorithm 1, and the procedure for the variational classifier is detailed in Algorithm 2. Let x_{ij} denote the scalar value of the j-th feature for the i-th data sample, with $x_{ij} \in \mathbb{R}$. The angle encoding transforms the scalar values x_{ij} into rotation angles $\theta_{ij} = \arccos(\sqrt{x_{ij}})$, which

are then used to apply single-qubit rotations to the quantum state. The encoded quantum state is then processed through a series of parameterized quantum gates, typically organized into layers, to create a quantum variational circuit. The choice of quantum gates and their arrangement can vary depending on the problem, but common choices include Hadamard, Pauli-X, Pauli-Y, Pauli-Z, CNOT, and other entangling gates [13]. In our project, we specifically chose to utilize the Hadamard gate. We opted for the Hadamard gate over other parameterized quantum gates because it's capacity to generate superposition states allowed us to fully explore PCA-derived data space, despite its reduced dimensionality. We could efficiently scan through the entire transformed solution space and capture the patterns differentiating malware from benign instances. This is due to the Hadamard gate's ability to transform computational basis states into superpositions, thereby allowing our quantum algorithm to address the complexities inherent in this binary classification problem. The output of the quantum circuit is obtained by measuring the quantum state of the qubits, usually in the computational basis, which provides an expectation value that can be used for classification tasks. Classical optimization methods, like gradient descent and the Adam optimizer, are utilized to optimize the parameters of quantum gates. These techniques work in conjunction with evaluations conducted on a quantum computer or simulator. The objective is to minimize a predefined cost function - in this case, cross-entropy, which is used for classification purposes. During the optimization process, the scalar training data is used to compute the cost function and its gradients with respect to the parameters of the quantum gates, guiding the optimization towards better solutions.

Algorithm 1 Quantum Circuit for Quantum Variational Classifier

Function `quantum_circuit(`*parameters, data*`)`
  `// Create a quantum circuit using Pennylane`
  for *i in range(num_qubits)* **do**
    Apply Hadamard gate to the i-th qubit
  end
  `// Apply AngleEmbedding with data and rotation`
  AngleEmbedding(features = data, wires = range(num_qubits), rotation = 'Y')
  `// Apply StronglyEntanglingLayers with parameters`
  StronglyEntanglingLayers(weights = parameters, wires = range(num_qubits))
  return *Expectation value of PauliZ on the first qubit*
EndFunction

Algorithm 2 Variational Classifier Function

Function `variational_classifier(`*weights, bias, x*`)`
  `// Call the quantum_circuit function with the given parameters and input data`
  circuit_output ← quantum_circuit(weights, x)
  `// Add bias to the circuit output`
  classifier_output ← circuit_output + bias
  return *classifier_output*
EndFunction

In this study, we devoted to develop a quantum counterpart for random noise and quantum adaptation of the Fast Gradient Sign Method (FGSM). However, for the classical neural network model, we opted to employ existing methodologies for both random noise and classical FGSM [14]. This approach allowed us to concentrate on our primary objective while ensuring robustness in the conventional aspects of our model. We add random perturbations to the training dataset by generating a noise matrix. This synthetic noise can be considered a controlled simulation of quantum noise, given its random and unpredictable nature. The procedure for introducing random perturbation is outlined in Algorithm 3. We can add random perturbations by generating a noise matrix $N \in \mathbb{R}^{n \times m}$ with elements N_{ij} sampled from a standard normal distribution ($N_{ij} \sim \mathcal{N}(0,1)$ for $i = 1, \ldots, n$ and $j = 1, \ldots, m$). We then scale the noise matrix N by a factor ϵ to create a new matrix $S \in \mathbb{R}^{n \times m}$, where $S_{ij} = \epsilon N_{ij}$ for $i = 1, \ldots, n$ and $j = 1, \ldots, m$. The factor ϵ controls the magnitude of the random perturbations added to the training data set. Adjusting the value of ϵ, we can effectively control the level of noise introduced into the data during the training process. A lower value of ϵ would result in minor perturbations, while a higher value would lead to more significant perturbations.

Finally, we add the scaled noise matrix S to the original training data set X_{train} to create a perturbed dataset $X_{\text{perturbed}} \in \mathbb{R}^{n \times m}$, with $X_{\text{perturbed},ij} = X_{\text{train},ij} + S_{ij}$ for $i = 1, \ldots, n$ and $j = 1, \ldots, m$.

Algorithm 3 Function to add random perturbations to input data.

Function `add_perturbation(`*data, epsilon*`)`
  noise ← np.random.randn(*data.shape) scaled_noise ← epsilon * noise perturbed_data ← data + scaled_noise
  return *perturbed_data*
EndFunction

C. QFGSM Algorithm

The Fast Gradient Sign Method (FGSM) is an alternative and robust methodology to assess the vulnerability of our quantum machine learning model. This approach revolves around the concept of adversarial examples, which are essentially manipulated inputs created to mislead the machine learning model. In quantum machine learning, we employ the FGSM algorithm to generate adversarial examples by introducing small pertur-

bations to the parameters of a quantum circuit (i.e., weights and bias). The FGSM algorithm operates by first calculating the gradient of the loss function with respect to the input data. Subsequently, it forms a perturbation proportional to the sign of this gradient. The original input data are then altered by adding this perturbation, thereby creating the adversarial example. Mathematically, given a quantum model parameterized by weights w and bias b, the FGSM algorithm proceeds as follows. First, it computes the gradient of the loss function $L(w, b, x, y)$ with respect to w and b given an input x and a target output y. The detailed steps of this procedure are described in Algorithm 4. This is achieved using the qml.grad function:

$$\nabla_w L, \nabla_b L = \text{qml.grad}(L(w, b, x, y), \text{argnum} = [0, 1]) \quad (1)$$

The sign function is then applied to these gradients to generate a binary vector that only considers the direction of the steepest increase of the function:

$$\text{sign}(\nabla_w L), \text{sign}(\nabla_b L) = \text{np.sign}(\nabla_w L), \text{np.sign}(\nabla_b L) \quad (2)$$

Here, the distinction between "sign" and "np.sign" is mostly a difference of notation and context: "sign" is the general mathematical function, and "np.sign" is a specific implementation of it that operates on arrays.

Next, the algorithm computes the perturbation by multiplying the sign of the gradient by a small constant ϵ:

$$\Delta w, \Delta b = \epsilon \cdot \text{sign}(\nabla_w L), \epsilon \cdot \text{sign}(\nabla_b L) \quad (3)$$

Finally, adversarial examples are created by adding the perturbation to the original weights and bias:

$$w_{\text{perturbed}}, b_{\text{perturbed}} = w + \Delta w, b + \Delta b \quad (4)$$

The overall effect of the FGSM algorithm is to generate inputs that result in incorrect predictions, providing a means to investigate the robustness of quantum machine learning models.

Algorithm 4 Quantum FGSM Function.

Function `QuantumFGSM(`**weights**, **bias**, **x**, **y**, ϵ)
  `/* Initialize gradient function */`
  gradient_fn ← qml.grad(cost_fn, argnum=[0, 1])
  `/* Compute gradients */`
  gradient_weights, **gradient_bias** ← **gradient_fn**(**weights**, **bias**, **x**, **y**)
  `/* Compute perturbations */`
  perturbation_weights ← ϵ × np.sign(**gradient_weights**) **perturbation_bias** ← ϵ × np.sign(**gradient_bias**)
  `/* Generate perturbed weights and biases */`
  weights_perturbed ← **weights** + **perturbation_weights** **bias_perturbed** ← **bias** + **perturbation_bias**
  return **weights_perturbed**, **bias_perturbed**
EndFunction

IV. Results And Discussion

Table 1 and Table 2 provide comprehensive comparative evaluations of two types of machine learning models – traditional Neural Networks (NN) and Quantum Neural Networks (QNN). Each model's performance is assessed using four key metrics: Accuracy, Precision, Recall, and F1 score, before and after adversarial attacks. Two distinct types of adversarial attacks are investigated: an algorithm introducing random noise (Table 1), and Fast Gradient Sign Method (FGSM) combined with Quantum Fast Gradient Sign Method (QFGSM) (Table 2). In Table 1, a distinct trend can be observed in the performance of the NN and QNN models under the influence of random noise. For both models, performance tends to decrease with increasing noise, reflecting the degradation in the model's ability to make accurate predictions as noise contaminates the data. However, there are several instances where the performance remains the same or slightly improves after the attack. This could potentially be due to the model overfitting to the noise, misinterpreting it as useful information or there might also be instances where the noise does not significantly impact the model's performance. This could occur if the noise introduced does not substantially distort the relevant features the model uses to make its predictions. The model could still identify and use the meaningful patterns in the data to make relatively accurate predictions, despite the presence of noise. This would indicate a certain level of robustness in the model to noise, but it could also reflect the fact that the noise added was not substantial or impactful enough to meaningfully disrupt the model's performance. Random noise introduces uncertainty and disrupts the inherent structure within the dataset. The impact of this disruption manifests as a decrease in the model's performance as it becomes more challenging to identify the underlying patterns in the noisy data. Therefore, in the case of random noise, the adversarial attack might not be effective enough to decrease the model's performance. Random noise, unlike more sophisticated adversarial attacks like FGSM, doesn't necessarily exploit model's weaknesses. Hence, the impact of random noise can be less damaging. From the table 1 (NN model), at a 0.05 proportion, the accuracy before the attack is 0.54, and after the attack is 0.54 as well. This suggests that the noise introduced during the adversarial attack did not affect the model's ability to correctly classify instances. It's important to keep in mind that this doesn't necessarily mean the model is robust against all adversarial attacks but rather in this instance, the introduced random noise didn't significantly change the input data in a way that affected the model's decision boundary. For the QNN model at a 0.15 proportion, the accuracy actually improved slightly from 0.50 to 0.52 after the adversarial attack. The analysis of data presented in Table 2 provides an insightful evaluation of the performance of both Neural Networks (NN) and Quantum Neural Networks (QNN) in relation to adversarial attacks, specifically using the Fast Gradient Sign Method (FGSM) and its quantum counterpart QFGSM. Before the adversarial attack, both models display a fluctuating but relatively strong performance with respect to these metrics,

across a range of proportions from 0.05 to 1.00. The highest accuracy achieved by the NN and QNN prior to attack is 0.62 (at proportion 0.30) and 0.68 (at proportion 0.25), respectively. However, upon the introduction of adversarial noise through FGSM, a decline in the performance metrics is observed for both models. FGSM, a white-box adversarial attack, operates by utilizing the gradients of the neural network loss function with respect to the input data to create perturbations that maximize the loss. This effectively crafts adversarial samples that mislead the model, resulting in erroneous predictions and consequently, reduced performance. In the context of this study, the impact of FGSM appears more profoundly on the QNN as compared to the NN. For example, the minimum accuracy of the QNN post-attack dips drastically to 0.21 (at proportion 0.15), significantly lower than that of the NN, which maintains a lowest post-attack accuracy of 0.40 (at proportion 0.35). The precision, recall, and F1 scores of the QNN also experience substantial reductions, echoing the trend observed in the accuracy. Even though the NN also exhibits a decrease in performance post-attack, its metrics remain relatively higher across all proportions compared to the QNN. This might be indicative of the NN's comparatively higher resilience to adversarial attacks introduced by FGSM. Both NN and QNN models show performance degradation after the adversarial attack using FGSM, with the QNN showing a higher degree of vulnerability. The QFGSM, being the quantum variant of FGSM, was designed to provide perturbations in a way that would increase the likelihood of misclassification by the model. The fact that the QNN shows a higher degree of vulnerability to these adversarial attacks indicates that the QFGSM is indeed fulfilling its intended purpose effectively.

In addition to accuracy, precision, recall, and F1 score metrics, we also utilized Receiver Operating Characteristic (ROC) curves to evaluate the performance of our models. Due to page constraints, these graphs are not included in this manuscript. However, they can be accessed here.

V. Conclusion

This paper investigates the capacity of both traditional and quantum neural networks to withstand disruptive influences. Performance of these networks was evaluated based on four standard metrics: accuracy, precision, recall, and the F1 score. These networks were then subjected to two types of disruptions: one generated through random noise, and the other utilizing a more sophisticated approach known as the Quantum Fast Gradient Sign Method (QFGSM). As anticipated, the introduction of additional random noise resulted in a decline in the network performance. However, a significantly steeper drop in performance was observed when FGSM and QFGSM methods. This decline was particularly prominent in quantum neural networks, especially during the application of a quantum-enhanced FGSM attack. Our findings underscore the urgent need for improved defensive measures for these networks against disruptive influences. The development of such protections is particularly crucial for quantum neural networks, given the expanding interest and research in this field.

Acknowledgement

The work is supported by the National Science Foundation under NSF Award #2433800, #1946442, and #2100134. Any opinions, findings, recommendations, expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

[1] U. Ahmed, J. C.-W. Lin, and G. Srivastava, "Mitigating adversarial evasion attacks by deep active learning for medical image classification," *Multimedia Tools and Applications*, vol. 81, no. 29, pp. 41899–41910, 2022.

[2] S. Yan, J. Ren, W. Wang, L. Sun, W. Zhang, and Q. Yu, "A survey of adversarial attack and defense methods for malware classification in cyber security," *IEEE Communications Surveys & Tutorials*, 2022.

[3] K. H. Shibly, M. D. Hossain, H. Inoue, Y. Taenaka, and Y. Kadobayashi, "Autonomous driving model defense study on hijacking adversarial attack," in *Artificial Neural Networks and Machine Learning–ICANN 2022: 31st International Conference on Artificial Neural Networks, Bristol, UK, September 6–9, 2022, Proceedings; Part IV*, pp. 546–557, Springer, 2022.

[4] A.-R. A. Audu, A. Cuzzocrea, C. K. Leung, K. A. MacLeod, N. I. Ohin, and N. C. Pulgar-Vidal, "An intelligent predictive analytics system for transportation analytics on open data towards the development of a smart city," in *Complex, Intelligent, and Software Intensive Systems: Proceedings of the 13th International Conference on Complex, Intelligent, and Software Intensive Systems (CISIS-2019)*, pp. 224–236, Springer, 2020.

[5] P. P. F. Balbin, J. C. Barker, C. K. Leung, M. Tran, R. P. Wall, and A. Cuzzocrea, "Predictive analytics on open big data for supporting smart transportation services," *Procedia computer science*, vol. 176, pp. 3009–3018, 2020.

[6] A. Cuzzocrea, "Combining multidimensional user models and knowledge representation and management techniques for making web services knowledge-aware," *Web Intelligence and Agent Systems: An international journal*, vol. 4, no. 3, pp. 289–312, 2006.

[7] Z. Wu, W. Yin, J. Cao, G. Xu, and A. Cuzzocrea, "Community detection in multi-relational social networks," in *Web Information Systems Engineering–WISE 2013: 14th International Conference, Nanjing, China, October 13-15, 2013, Proceedings, Part II 14*, pp. 43–56, Springer, 2013.

[8] S. Lu, L.-M. Duan, and D.-L. Deng, "Quantum adversarial machine learning," *Physical Review Research*, vol. 2, no. 3, p. 033212, 2020.

[9] Y. Du, M.-H. Hsieh, T. Liu, D. Tao, and N. Liu, "Quantum noise protects quantum classifiers against adversaries," *Physical Review Research*, vol. 3, no. 2, p. 023153, 2021.

[10] M. T. West, S. M. Erfani, C. Leckie, M. Sevior, L. C. Hollenberg, and M. Usman, "Benchmarking adversarially

TABLE I: Performance metrics for NN and QNN against adversarial attack with random noise algorithm

Model	Proportion	Before Adversarial Attack				After Adversarial Attack			
		Accuracy	Precision	Recall	F1 score	Accuracy	Precision	Recall	F1 score
NN	0.05	0.54	1.00	0.54	0.70	0.50	0.62	0.45	0.62
	0.10	0.54	1.00	0.54	0.70	0.54	1.00	0.54	0.70
	0.15	0.57	0.94	0.57	0.67	0.50	0.50	1.00	0.67
	0.20	0.52	0.81	0.53	0.60	0.45	0.00	0.00	0.00
	0.25	0.48	0.92	0.47	0.56	0.48	1.00	0.16	0.28
	0.30	0.62	1.00	0.62	0.77	0.45	0.41	1.00	0.58
	0.35	0.42	0.71	0.42	0.52	0.42	0.46	0.80	0.58
	0.40	0.57	0.98	0.57	0.71	0.55	0.00	0.00	0.00
	0.45	0.65	0.87	0.65	0.71	0.42	0.42	1.00	0.60
	0.50	0.56	1.00	0.48	0.65	0.51	1.00	0.51	0.68
	0.55	0.53	1.00	0.53	0.70	0.46	1.00	0.46	0.63
	0.60	0.50	1.00	0.50	0.66	0.53	1.00	0.53	0.69
	0.65	0.57	1.00	0.57	0.72	0.48	1.00	0.48	0.65
	0.70	0.52	1.00	0.52	0.68	0.51	1.00	0.51	0.68
	0.75	0.49	1.00	0.49	0.66	0.47	1.00	0.47	0.64
	0.80	0.50	1.00	0.50	0.67	0.60	1.00	0.60	0.75
	0.85	0.53	1.00	0.53	0.69	0.45	1.00	0.45	0.62
	0.90	0.52	1.00	0.52	0.69	0.58	1.00	0.58	0.74
	0.95	0.57	0.55	0.55	0.55	0.60	0.62	0.62	0.62
	1.00	0.56	0.55	0.55	0.55	0.65	0.64	0.66	0.64
QNN	0.05	0.52	0.84	0.52	0.64	0.43	0.53	0.24	0.33
	0.10	0.52	0.98	0.53	0.66	0.50	0.50	1.00	0.67
	0.15	0.57	0.92	0.57	0.65	0. 28	0.17	0.04	0.06
	0.20	0.38	0.53	0.38	0.43	0.72	0.71	0.87	0.78
	0.25	0.68	0.87	0.68	0.70	0.62	0.66	0.84	0.74
	0.30	0.40	0.92	0.40	0.56	0.48	0.47	0.89	0.62
	0.35	0.53	0.76	0.53	0.62	0.62	0.61	0.96	0.75
	0.40	0.55	0.57	0.68	0.62	0.56	0.56	0.59	0.56
	0.45	0.57	0.87	0.57	0.66	0.45	0.47	0.90	0.62
	0.50	0.42	0.51	0.42	0.45	0.45	0.47	0.85	0.61
	0.55	0.40	0.62	0.40	0.49	0.55	1.00	0.25	0.40
	0.60	0.57	0.87	0.57	0.66	0.56	0.00	0.00	0.00
	0.65	0.53	0.81	0.53	0.64	0.62	0.62	1.00	0.77
	0.70	0.48	0.79	0.47	0.51	0.35	0.38	0.82	0.52
	0.75	0.62	0.82	0.62	0.66	0.45	0.00	0.00	0.00
	0.80	0.42	0.79	0.42	0.55	0.35	0.46	0.54	0.50
	0.85	0.68	0.89	0.68	0.71	0.52	0.53	1.00	0.69
	0.90	0.60	0.79	0.60	0.65	0.42	0.45	0.89	0.60
	0.95	0.53	0.52	0.54	0.52	0.50	0.53	0.54	0.53
	1.00	0.55	0.53	0.54	0.53	0.50	0.52	0.52	0.52

robust quantum machine learning at scale," *Physical Review Research*, vol. 5, no. 2, p. 023186, 2023.

[11] H. Suryotrisongko, Y. Musashi, A. Tsuneda, and K. Sugitani, "Adversarial robustness in hybrid quantum-classical deep learning for botnet dga detection," *Journal of Information Processing*, vol. 30, pp. 636–644, 2022.

[12] A. Kumar, K. Kuppusamy, and G. Aghila, "A learning model to detect maliciousness of portable executable using integrated feature set," *Journal of King Saud University-Computer and Information Sciences*, vol. 31, no. 2, pp. 252–265, 2019.

[13] P. De, S. Ranwa, and S. Mukhopadhyay, "Intensity and phase encoding for realization of integrated pauli x, y and z gates using 2d photonic crystal," *Optics & Laser Technology*, vol. 152, p. 108141, 2022.

[14] T. Muncsan and A. Kiss, "Transferability of fast gradient sign method," in *Intelligent Systems and Applications: Proceedings of the 2020 Intelligent Systems Conference (IntelliSys) Volume 2*, pp. 23–34, Springer, 2021.

TABLE II: Performance metrics for NN and QNN against adversarial attack with FGSM and QFGSM noise algorithm

Model	Proportion	Before Adversarial Attack				After Adversarial Attack			
		Accuracy	Precision	Recall	F1 score	Accuracy	Precision	Recall	F1 score
NN	0.05	0.54	1.00	0.54	0.70	0.53	1.00	0.53	0.70
	0.10	0.54	1.00	0.54	0.70	0.45	0.65	0.51	0.60
	0.15	0.57	0.94	0.57	0.67	0.45	0.45	0.55	0.50
	0.20	0.52	0.81	0.53	0.60	0.48	0.50	0.52	0.54
	0.25	0.48	0.92	0.47	0.56	0.44	0.53	0.55	0.52
	0.30	0.62	1.00	0.62	0.77	0.42	0.42	0.42	0.53
	0.35	0.42	0.71	0.42	0.52	0.40	0.41	0.41	0.50
	0.40	0.57	0.98	0.57	0.71	0.45	0.45	0.45	0.45
	0.45	0.65	0.87	0.65	0.71	0.47	0.45	0.43	0.46
	0.50	0.56	1.00	0.48	0.65	0.43	0.43	0.44	0.45
	0.55	0.53	1.00	0.53	0.70	0.41	0.47	0.46	0.48
	0.60	0.50	1.00	0.50	0.66	0.51	0.55	0.55	0.55
	0.65	0.57	1.00	0.57	0.72	0.46	0.46	0.48	0.49
	0.70	0.52	1.00	0.52	0.68	0.45	0.45	0.45	0.55
	0.75	0.49	1.00	0.49	0.66	0.43	0.44	0.41	0.42
	0.80	0.50	1.00	0.50	0.67	0.43	0.45	0.45	0.45
	0.85	0.53	1.00	0.53	0.69	0.45	0.56	0.54	0.55
	0.90	0.52	1.00	0.52	0.69	0.46	0.55	0.56	0.58
	0.95	0.57	0.55	0.55	0.55	0.55	0.58	0.58	0.58
	1.00	0.56	0.55	0.55	0.55	0.54	0.55	0.55	0.55
QNN	0.05	0.52	0.84	0.52	0.64	0.39	0.42	0.25	0.30
	0.10	0.52	0.98	0.53	0.66	0.41	0.42	0.42	0.42
	0.15	0.57	0.92	0.57	0.65	0. 21	0.00	0.00	0.00
	0.20	0.38	0.53	0.38	0.43	0.31	0.45	0.45	0.45
	0.25	0.68	0.87	0.68	0.70	0.41	0.39	0.46	0.44
	0.30	0.40	0.92	0.40	0.56	0.41	0.43	0.37	0.41
	0.35	0.53	0.76	0.53	0.62	0.46	0.37	0.35	0.38
	0.40	0.55	0.57	0.68	0.71	0.48	0.55	0.56	0.55
	0.45	0.57	0.87	0.57	0.66	0.30	0.34	0.37	0.36
	0.50	0.42	0.51	0.42	0.45	0.33	0.36	0.39	0.40
	0.55	0.40	0.62	0.40	0.49	0.36	35	0.37	0.40
	0.60	0.57	0.87	0.57	0.51	0.47	0.45	0.43	0.41
	0.65	0.53	0.81	0.53	0.64	0.32	0.38	0.27	0.31
	0.70	0.48	0.79	0.47	0.51	0.25	0.27	0.32	0.32
	0.75	0.62	0.82	0.62	0.66	0.21	0.20	0.20	0.20
	0.80	0.42	0.79	0.42	0.55	0.21	0.23	0.22	0.20
	0.85	0.68	0.89	0.68	0.71	0.41	0.43	0.44	0.42
	0.90	0.60	0.79	0.60	0.65	0.35	0.36	0.36	0.37
	0.95	0.53	0.52	0.54	0.52	0.45	0.48	0.48	0.48
	1.00	0.55	0.53	0.54	0.53	0.44	0.45	0.45	0.45