A Privacy-Preserving Framework for Efficient Network Intrusion Detection in Consumer Network Using Quantum Federated Learning

Zakaria Abou El Houda, *Member, IEEE*, Hajar Moudoud, *Member, IEEE*,
Bouziane Brik, *Senior Member, IEEE*, and Muhammad Adil, *Senior Member, IEEE*

***Abstract*—The proliferation of consumer networks has increased vulnerabilities to network intrusions, emphasizing the critical need for robust intrusion detection systems (IDS). The data-driven Artificial Intelligence (AI) approach has gained attention for enhancing IDS capabilities to deal with emerging security threats. However, these AI-based IDS face challenges in scalability and privacy preservation. More importantly, they are time-consuming and may perform poorly on high-dimensional and complex data due to the lack of computational resources. To address these shortcomings, in this paper, we introduce a novel framework, called Quantum Federated Learning IDS (QFL-IDS), that merges Quantum Computing (QC) with Federated Learning (FL) to allow for an efficient, robust, and privacy-preserving approach for detecting network intrusions in consumer networks. Leveraging the decentralized nature of FL, QFL-IDS enables multiple consumer devices to collaboratively train a global intrusion detection model while preserving the privacy of individual user data. Furthermore, we leverage the computational power of quantum computing to improve the efficiency of model training and inference processes. We demonstrate the efficacy of our framework through extensive experiments. The obtained results show significant improvements in detection accuracy and computational efficiency compared to the current traditional centralized and federated learning approaches. This makes QFL-IDS a promising framework to cope with the new emerging security threats in a timely and effective manner.**

***Index Terms*—Intrusion detection systems (IDS), quantum computing (QC), federated learning (FL), quantum federated learning, consumer network.**

I. Introduction

In the digital era, the expansion of consumer networks has been exponential, serving as the backbone for various smart devices and Internet of Things (IoT) applications. These networks, characterized by their vastness and diversity, have become critical infrastructures for personal, commercial, and governmental operations. However, this expansion has led to a surge in network vulnerabilities, making them prime targets for cyber-attacks. The financial losses are anticipated to increase from $6 trillion in 2021 to $10.5 trillion by 2025 [1]. This highlights the need to develop reliable, robust, and effective Intrusion Detection systems (IDSs) to protect consumer networks [2], [3]. Traditional IDS often struggle with scalability issues due to the growing volume and complexity of data. They also face challenges in preserving user privacy while ensuring timely and accurate detection of malicious network traffic. Moreover, the computational demands of processing and analyzing such vast datasets can overwhelm conventional computing resources, leading to inefficiencies and delays in threat detection and mitigation.

Quantum Computing (QC), particularly Quantum Machine Learning (QML), emerge as promising technologies that are capable of significantly strengthening IDS capabilities by addressing their main issues, including scalability, privacy, and computational efficiency [4]. QML leverages the principles of quantum physics to perform computations, leveraging quantum resources like entanglement and superposition of states. This allows quantum computers to handle exponential growth in the dimensions of attack data. Unlike traditional computers that use bits, quantum computers utilize qubits, which can exist in multiple states simultaneously. The intrinsic parallelism capability of quantum computers allows them to handle high-dimensional datasets and execute mathematical operations and linear algebra much faster than classical computers. In 2019, Google demonstrated quantum supremacy by solving a problem in 200 seconds that would take a classical computer 10,000 years, while IBM is actively developing larger quantum computers with, a clear roadmap for scaling up quantum technologies with more than 1000 qubits. These advancements that result from quantum algorithms such as Shor's and Grover's algorithms, quantum approximate optimization algorithm (QAOA), and Boltzmann quantum machines (QBM), are rapidly leading to the large deployment and integration of quantum computer capabilities, in several fields. Quantum Federated Learning (QFL) merges classical Federated Learning (FL) [5], [6], [7], [8] with quantum computing to address the rising complexities in data processing.

Manuscript received 31 March 2024; revised 3 July 2024; accepted 14 August 2024. Date of publication 12 September 2024; date of current version 31 December 2024. This work was supported by INRS-EMT. *(Corresponding author: Zakaria Abou El Houda.)*

Zakaria Abou El Houda is with the Centre Énergie Matériaux Télécommunications, UMR INRS-UQO, Institut National de la Recherche Scientifique, Gatineau, QC H3T 1J4, Canada (e-mail: zakaria.abouelhouda@inrs.ca).

Hajar Moudoud is with the Département d'informatique et d'ingénierie, Université du Québec en Outaouais, Gatineau, QC J1K 2R1, Canada (e-mail: hajar.moudoud@uqo.ca).

Bouziane Brik is with the Computer Science Department, College of Computing and Informatics, University of Sharjah, Sharjah, UAE (e-mail: bbrik@sharjah.ac.ae).

Muhammad Adil is with the Department of Computer Science and Engineering, University at Buffalo, Buffalo, NY 14215 USA (e-mail: Muhammad.adil@ieee.org).

Digital Object Identifier 10.1109/TCE.2024.3458985

QFL, will not only preserve data privacy and initiate more institutions to share their knowledge to detect zero-day attacks but also, allow for local training given the fragile nature of computing qubits and the difficulty of transferring them through the networks. QFL leverages the decentralized data processing of FL with the unparalleled computational power of QC, offering a transformative solution to enhance IDS capabilities to deal with zero-day attacks. This ensures that the initial deployment of QFL can leverage the robustness and widespread availability of current classical systems, enabling a smoother transition and more efficient implementation. By utilizing these pre-optimized parameters, QFL can effectively bridge the gap between quantum and classical computing environments, promoting a more practical and scalable adoption of quantum technologies in the immediate future. By harnessing QC's potential, QFL significantly enhances the efficiency of model training and inference processes [9]. This integration promises to overcome the computational bottlenecks of traditional IDS, reducing the training time and an improvement in detection accuracy over traditional methods [9], [10]. This approach not only preserves the privacy of individual users but also reduces the bandwidth requirements typically associated with centralized learning approaches [10], [11].

In this context, we introduce a novel framework, called Quantum Federated Learning IDS (QFL-IDS), that merges Quantum Computing (QC) with Federated Learning (FL) to allow for an efficient, robust, and privacy-preserving approach for detecting network intrusions in consumer networks. Leveraging the decentralized nature of FL, QFL-IDS enables multiple consumer devices to collaboratively train a global intrusion detection model while preserving the privacy of individual user data. Furthermore, we leverage the computational power of quantum computing to improve the efficiency of model training and inference processes. We demonstrate the efficacy of our framework through extensive experiments, which show significant improvements in detection accuracy and computational efficiency compared to the current traditional centralized and federated learning approaches. This makes QFL-IDS a promising framework to cope with the new emerging security threats in a timely and effective manner while preserving the privacy of consumer networks.

The main contributions of this paper are summarized as follows:

- We design a novel distributed architecture that enables multiple consumer devices to collaboratively train a shared model while preserving their privacy and enhancing efficiency via quantum training mechanisms.
- We propose a novel framework (QFL-IDS) that leverages FL and QC to ensure an efficient, robust, and privacy-preserving approach for detecting network intrusions in consumer networks.
- We demonstrate the efficacy of our framework through extensive experiments using the NSL-KDD public dataset, which shows significant improvements in detection accuracy and computational efficiency compared to traditional AI-based solutions.

The remainder of this paper is structured as follows. Section II presents the background of QML and reviews the state-of-the-art solutions. Section III describes our system model. The performance evaluation is presented in Section IV. Section V concludes the paper.

II. Background and Related Work

The convergence of quantum computing and machine learning offers the potential to solve complex computational problems more efficiently than classical methods, potentially advancing the detection and mitigation of emerging cyber threats, such as zero-day attacks. This section provides an overview of QML and reviews key studies that utilize QML for attack detection.

A. Quantum Computing

Quantum computing represents a fundamental shift from classical computing paradigms, harnessing the principles of quantum mechanics to process information in a profoundly different way. At its core, quantum computing utilizes qubits (quantum bits) instead of the traditional bits seen in classical computing. Quantum computation involves initializing qubits in a certain state, applying a series of quantum gates (quantum circuit) to perform operations, and finally measuring the qubits to collapse their state to a classical output. The power of quantum computing lies in the ability to perform operations on all possible states simultaneously and to exploit interference and entanglement phenomena to solve certain problems more efficiently than classical computers.

For a system of n qubits, the general state can be described as:

$$|\Psi\rangle = \sum_{x=0}^{2^n-1} c_x |x\rangle$$

where $|x\rangle$ represents the binary representation of x in the n-qubit system, and c_x are complex coefficients satisfying $\sum |c_x|^2 = 1$.

Qubits: A qubit is the basic unit of quantum information, analogous to the bit in classical computing. Unlike a classical bit, which can be in one of two states (0 or 1), a qubit can be in a state of superposition, where it represents both 0 and 1 simultaneously. The state of a qubit can be represented as:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$$

where $|\psi\rangle$ is the quantum state of the qubit, $|0\rangle$ and $|1\rangle$ are the basis states (analogous to the classical states 0 and 1), and α and β are complex numbers that satisfy the normalization condition $|\alpha|^2 + |\beta|^2 = 1$. This condition ensures that the total probability (the probability of finding the qubit in either state upon measurement) is 1.

Quantum Gates: Quantum gates manipulate the state of qubits and are the building blocks of quantum circuits, similar to logic gates in classical circuits. Quantum gates are represented mathematically by unitary matrices. A unitary matrix U has the property that its conjugate transpose $U^\dagger$ is also its inverse, i.e., $UU^\dagger = U^\dagger U = I$, where I is the identity matrix. This property ensures that quantum operations are reversible and preserve the total probability.

A simple example is the Hadamard gate (H), which puts a qubit into a state of superposition:

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

Applying H to $|0\rangle$ results in:

$$H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$

This operation puts the qubit into an equal superposition of $|0\rangle$ and $|1\rangle$.

Quantum Encoding: Encoding classical information into quantum states involves mapping classical data to qubits or groups of qubits. There are several methods for quantum encoding, including:

1) Basis Encoding: Basis encoding directly maps classical bits to qubits, with each qubit representing a single classical bit. For a classical bit string $x = x_1 x_2 \ldots x_n$, the corresponding quantum state is:

$$|x\rangle = |x_1\rangle \otimes |x_2\rangle \otimes \cdots \otimes |x_n\rangle, \tag{1}$$

where $\otimes$ denotes the tensor product, combining individual qubit states into a multi-qubit state.

2) Amplitude Encoding: Amplitude encoding maps classical data to the amplitudes of a quantum state, allowing for the compact representation of 2^n classical values using n qubits. For a normalized classical vector $\vec{x} \in \mathbb{R}^N$ with $N = 2^n$, the quantum state is:

$$|x\rangle = \sum_{i=0}^{N-1} x_i |i\rangle, \tag{2}$$

where $|i\rangle$ represents the i-th basis state in an n-qubit system.

3) Angle Encoding: Angle encoding uses the angles of rotation gates to encode classical data into qubits. A common approach is to use the rotation around the Y-axis (R_y) for a single qubit:

$$|x\rangle = R_y(\theta)|0\rangle = \cos\left(\frac{\theta}{2}\right)|0\rangle + \sin\left(\frac{\theta}{2}\right)|1\rangle, \tag{3}$$

where θ is a parameter encoding the classical information.

B. Related Work

The classically optimized nature of QML parameters allows for seamless integration into existing classical communication infrastructures. Several hybrid quantum-classical AI algorithms have demonstrated superior performance compared to classical AI algorithms. These hybrid models utilize quantum gates and circuits that emulate the operational principles of classical machine learning models. The most popular ones include Quantum support vector machine (QSVM) and variational quantum circuits (VQCs), which have shown great performance in detecting network attacks. VQC architectures include Quantum neural networks (QNNs), Quantum convolutional neural networks (QCNNs) [12], and also Quantum recurrent neural networks (QRNNs) [13]. Fully quantum ML models also have been well investigated to handle the complex nature of quantum systems, known as Parametrized Quantum Circuits (PQC) [14]. Blekos et al. [15] have begun exploring quantum algorithms for feature selection and network intrusion detection. Quantum algorithms, such as the Quantum Approximate Optimization Algorithm (QAOA) and Variational Quantum Eigensolver (VQE), have been proposed to optimize the feature selection process, potentially improving the accuracy and efficiency of intrusion detection models. Salek et al. [16] integrated classical machine learning models with quantum processing units (QPUs) to perform data encoding and quantum parallelism. Kukliansky et al. [17] developed QNNs within the constraints of current quantum computing capabilities specifically focusing on Noisy Intermediate-Scale Quantum (NISQ) devices, to detect network attacks. More specifically, the authors have developed an optimized multilayered QNN architecture that leverages quantum computational power that is available today for network attack detection. The authors Implemented a compact version of this architecture on IonQ's Aria-1 quantum computer. They achieved an F1 score of 0.86 using the NF-UNSW-NB15 dataset, demonstrating the feasibility and effectiveness of their proposed approach. Additionally, the authors introduced a new metric, the certainty factor, which quantifies the model's inherent sharpness and degree of separation between the predicted class probability distribution versus other classes. To address the degradation in performance of the current ML models when dealing with big data (*i.e.*, 10^6 samples or more), Kalinin and Krundyshev [18] proposed QSVM and QCNN to detect network attacks and they have compared their obtained results with traditional AI-based IDS in terms of accuracy and efficiency. They have developed a novel technique to encode data into quantum states. Their experimental results demonstrated that QML-based intrusion detection systems, such as QSVM and QCNN, can efficiently process large datasets with 98% accuracy, performing at twice the speed of traditional ML algorithms used for similar tasks. In [19], Suryotrisongko and Musashi proposed a hybrid quantum-classical deep learning models for detecting Domain Generation Algorithms (DGA)-based botnet threats. They proposed a novel approach that integrates quantum computations within a classical deep learning framework. The model incorporates Pennylane's quantum circuit embeddings and layer circuits to process features from the Botnet DGA dataset, including MinREBotnets, CharLength, TreeNewFeature, and nGramReputation_Alexa. They have studied the performance of their model under realistic quantum computing conditions by including noise models. Their obtained results showed that the hybrid model can achieve high accuracy (up to 94.7% in specific settings), outperforming traditional deep learning models in intrusion detection. In [20], Gouveia and correia proposed an unsupervised QML scheme to detect network attacks. The authors have explored the efficacy of a Quantum-Assisted NIDS that incorporates quantum circuits as an integral layer within a deep learning model.

III. System Model

In the following, we present a mathematical formulation of our proposed system model, focusing on quantum data encoding, quantum model training, federated learning aggregation, and the hybrid integration of classical and quantum machine learning.

A. Data Encoding

The first step involves encoding classical network data into quantum states. Given a classical feature vector $x_i'' \in \mathbb{R}^n$, where n is the number of features, the angle embedding strategy maps each feature x_{ij}'' to a rotation angle for a corresponding qubit in a quantum circuit. The encoding can be mathematically expressed as follows for each feature and qubit:

$$\mathrm{QE}\left(x_{ij}''\right) = R_x\left(x_{ij}''\right) \otimes R_y\left(x_{ij}''\right) \otimes R_z\left(x_{ij}''\right)|0\rangle_j, \tag{4}$$

where R_x, R_y, and R_z are rotation gates around the x, y, and z axes of the Bloch sphere, respectively, applied to the j-th qubit initialized in the state $|0\rangle$.
where $\theta_i = 2x_i$ is the rotation angle derived from the i-th element of the vector x.

The process for encoding the entire vector x into an n-qubit quantum system involves:

1) Initializing the quantum system in the ground state $|0\rangle^{\otimes n}$.
2) Applying the rotation $R_y(\theta_i)$ to qubit i, for each feature x_i in the vector x.

The quantum state of the system after encoding can be represented as:

$$|\psi\rangle = \bigotimes_{i=1}^{n} R_y(2x_i)|0\rangle_i, \tag{5}$$

where $\bigotimes$ denotes the tensor product, and $|0\rangle_i$ represents the initial state of the i^{th} qubit. This state $|\psi\rangle$ effectively encodes the classical data vector x into the quantum system.

B. Quantum Model Structure

A Quantum Neural Network (QNN) is utilized for processing the encoded data. The QNN consists of parameterized quantum circuits (PQCs) that act as variational models. The PQCs involve sequences of quantum gates, including rotation gates (R_x, R_y, R_z) and entangling gates (CNOT), parameterized by a set of angles θ. The output of the QNN for a given input $|x\rangle$ can be represented as:

$$|\psi(\theta)\rangle = U(\theta)|x\rangle, \tag{6}$$

where $U(\theta)$ represents the unitary operation of the quantum circuit parameterized by θ.

C. Measurement and Classical Integration

The output of the quantum circuit is obtained through measurement operations on each qubit, which collapses the quantum state into classical information:

$$M(|x_i''\rangle) = \left\langle x_i'' \middle| \bigotimes_{j=1}^{n} \sigma_z^{(j)} \middle| x_i'' \right\rangle,$$

where M represents the measurement operation and $\sigma_z^{(j)}$ is the Pauli-Z observable for the j-th qubit. The result of these measurements, a vector of expectation values, is then fed into the classical neural network for further processing.

Algorithm 1 QFL-IDS Process

```
Input: Distributed datasets $D$ across $K$ clients, where each dataset $D_k$ consists of pairs $(x_i, y_i)$
Input: Number of features in each data vector $x_i$ after preprocessing, $n$
Input: Number of layers in the Quantum Neural Network (QNN), $L$
Input: Total number of federated learning rounds, $T$
Output: Trained global QNN parameters $\theta_{\text{global}}$ and classical model parameters $W_{\text{global}}$
Initialize global QNN parameters $\theta_{\text{global}}$ and classical model parameters $W_{\text{global}}$
for $t = 1$ to $T$ do
    Distribute the current global parameters $\theta_{\text{global}}$ and $W_{\text{global}}$ to all $K$ clients
    for each client $k = 1$ to $K$ do
        Quantum Encoding:
            Transform each data vector $x_i$ in $D_k$ into a quantum state $|\psi_i\rangle$ using angle encoding
        QNN Processing:
            Apply the QNN with parameters $\theta_k$ to each quantum state $|\psi_i\rangle$, resulting in $|\psi_i'\rangle$
            Measure $|\psi_i'\rangle$ to obtain classical output vectors $o_i$
        Classical Neural Network Processing:
            Input the classical vectors $o_i$ into the classical model with parameters $W_k$
            Perform forward propagation, loss calculation, and backpropagation to update $\theta_k$ and $W_k$ based on $D_k$
    end for
    Aggregate updates from all clients to obtain $\theta_{\text{global}}^{\text{new}}$ and $W_{\text{global}}^{\text{new}}$
    Distribute $\theta_{\text{global}}^{\text{new}}$ and $W_{\text{global}}^{\text{new}}$ back to each client
end for
Finalize $\theta_{\text{global}}$ and $W_{\text{global}}$ as the trained model parameters
Evaluate the trained model on a separate test dataset to assess IDS performance
```

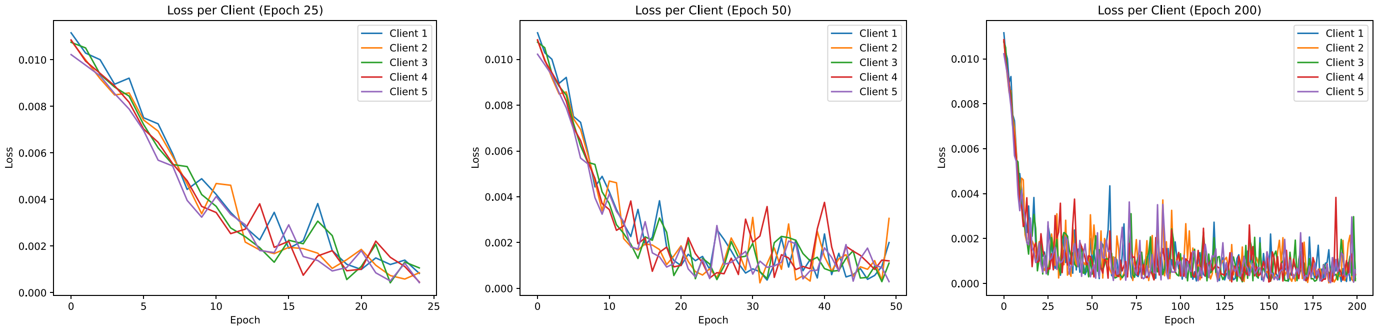
D. QFL-IDS Training Process

The training process involves adjusting the parameters θ to minimize a loss function that measures the difference between the predicted and actual labels. The loss function for a binary classification task can be defined as:

$$L(\theta) = \frac{1}{M}\sum_{m=1}^{M}(y_m - \langle\psi(\theta)|O|\psi(\theta)\rangle)^2, \tag{7}$$

where M is the number of samples, y_m is the actual label of the m-th sample, and O is an observable whose expectation value determines the prediction.

Consider a federated system of N nodes, each possessing a quantum dataset D_i, where $i \in 1, 2, \ldots, N$. Each node trains a local QNN model, characterized by quantum parameters θ_i,

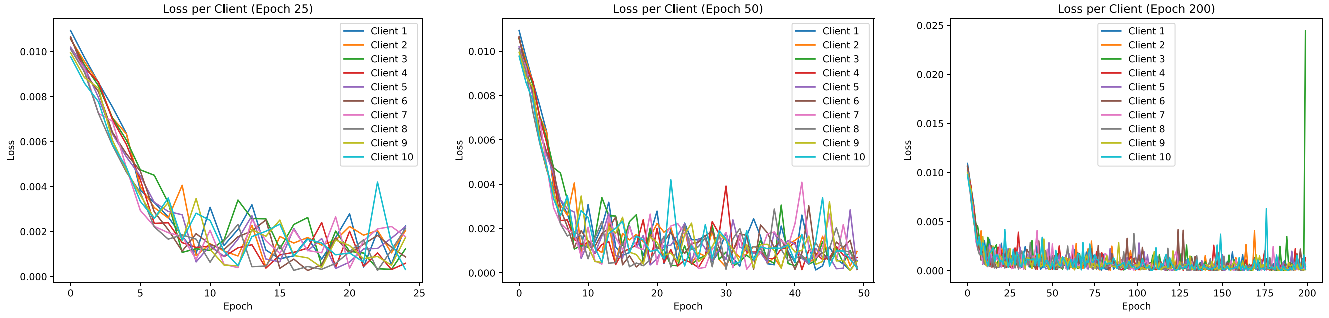


(a)

(b)

(c)

Fig. 1. Loss values per client for 5 distributed clients over (a) 25 training epochs; (b) 50 training epochs; and (c) 200 training epochs.



(a)

(b)

(c)

Fig. 2. Loss values per client for 10 distributed clients over (a) 25 training epochs; (b) 50 training epochs; and (c) 200 training epochs.

TABLE I
HYPERPARAMETERS AND SCENARIOS

Hyperparameters/Configurations	**Values**
Quantum Framework	PennyLane
Number of Qubits	5
Encoding technique	AngleEmbedding
Optimizer	Adam
Step Size for Adam Optimizer	0.0005
Number of Clients	5 to 15
Epochs	25 to 200
Parameters for QNN	Randomly initialized between $-\pi$ and π
Loss Function	Binary Cross-Entropy Loss

on its dataset. The aim is to minimize the local quantum loss function $L_i(\theta_i)$, reflective of the quantum dataset D_i.

Upon completing T training epochs on local quantum models, nodes transmit their model parameters θ_i^t to a centralized server. The server then performs quantum aggregation to update the global model parameters θ_{global}^t for the t-th round:

$$\theta_{global}^t = \frac{\sum_{i=1}^N |D_i| \theta_i^t}{\sum_{i=1}^N |D_i|}, \tag{8}$$

where $|D_i|$ represents the quantum data's magnitude or weight at node i, adjusting the contribution of each node's parameters based on its dataset size or importance.

The iterative quantum training and aggregation proceed until the global model's parameter change between successive rounds meets a specified quantum convergence criterion:

$$\Delta\theta_{quantum} = |\theta_{global}^t - \theta_{global}^{t-1}|_{quantum}, \tag{9}$$

where $| \cdot |quantum$ denotes a norm appropriate for quantum parameters, and iteration ceases when $\Delta\theta quantum$ is less than a predetermined threshold $\epsilon_{quantum}$.

Each node's local quantum model aims to minimize its quantum loss function. The global quantum optimization objective is defined as:

$$\min_\theta L_{global}^{quantum}(\theta) = \min_\theta \left(\frac{1}{N} \sum_{i=1}^N L_i^{quantum}(\theta) \right), \tag{10}$$

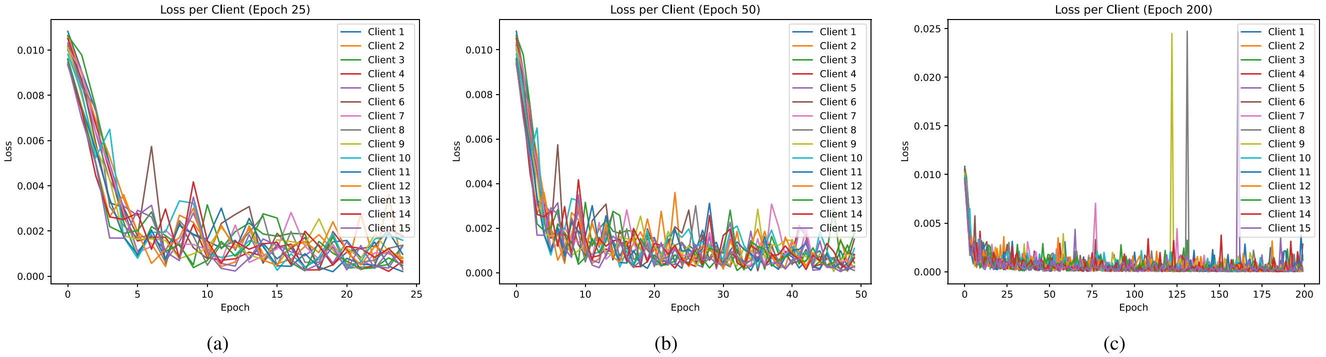
In the quantum setting, the update rule for a local model's quantum parameters at node i during the t-th round, considering a quantum-specific learning rate $\eta_{quantum}$, is:

$$\theta_i^{t+1} = \theta_i^t - \eta_{quantum} \nabla L_i^{quantum}(\theta_i^t), \tag{11}$$

where $\nabla L_i^{quantum}(\theta_i^t)$ represents the gradient of the quantum loss function $L_i^{quantum}$ with respect to the quantum parameters θ_i^t at node i. Algorithm 1 summarizes our QFL-IDS training process. QFL-IDS framework underscores the intricacies of integrating quantum computing techniques with classical machine learning in a federated learning setting, highlighting the innovative approach to enhancing IDS capabilities.

IV. Performance Evaluation

To develop QFL-IDS, we leverage PennyLane [26], a quantum machine learning library. For the attack traffic, we have used the NSL-KDD dataset that is derived from the original KDD Cup 1999 dataset, addressing various shortcomings such as redundancy and bias. The dataset comprises network

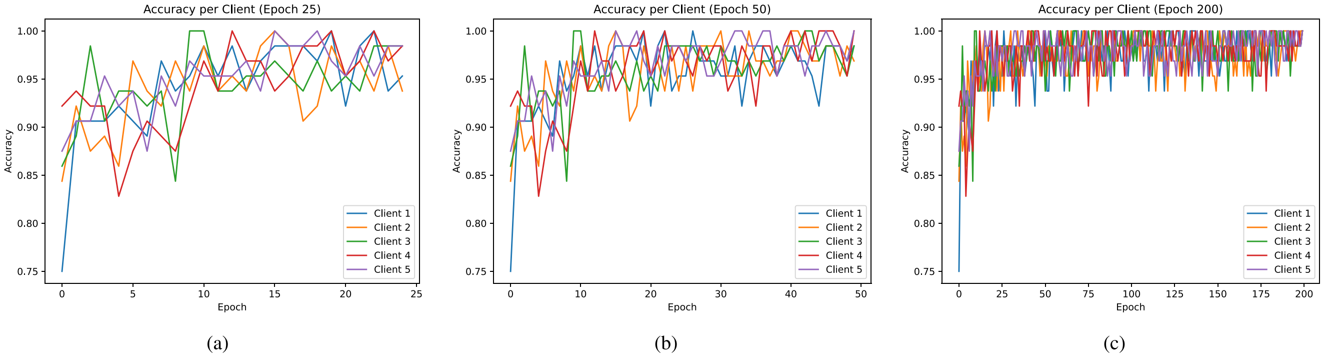


(a)

(b)

(c)

Fig. 3. Loss values per client for 25 distributed clients over (a) 25 training epochs; (b) 50 training epochs; and (c) 200 training epochs.

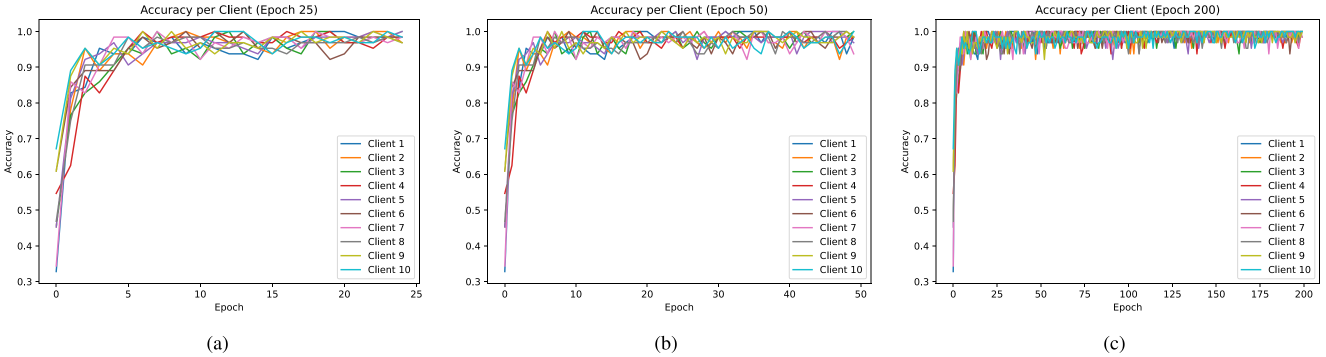


(a)

(b)

(c)

Fig. 4. Model accuracy values per clients for 5 distributed clients over (a) 25 training epochs; (b) 50 training epochs; and (c) 200 training epochs.

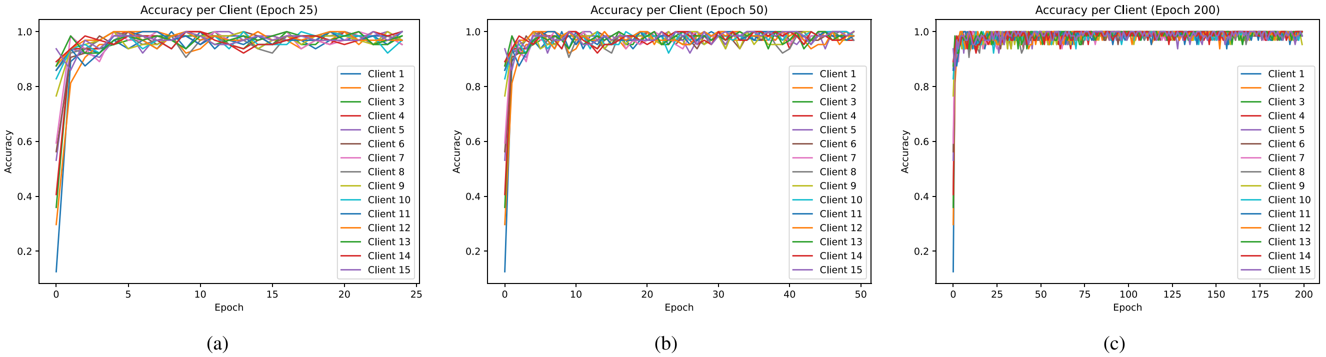


(a)

(b)

(c)

Fig. 5. Model accuracy values per clients for 10 distributed clients over (a) 25 training epochs; (b) 50 training epochs; and (c) 200 training epochs.



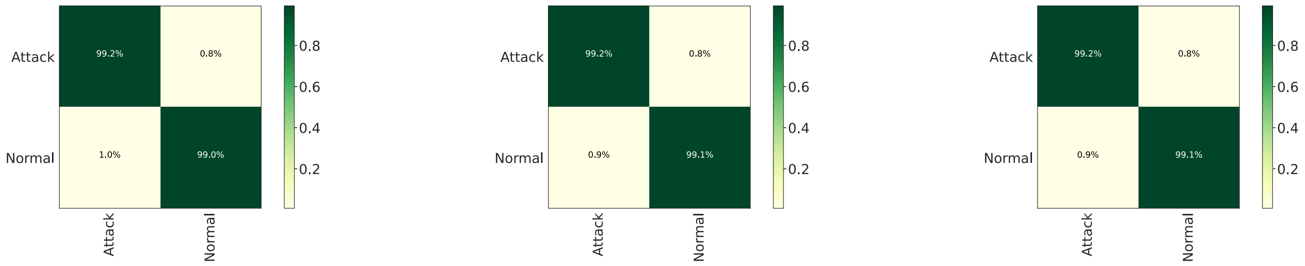
(a)

(b)

(c)

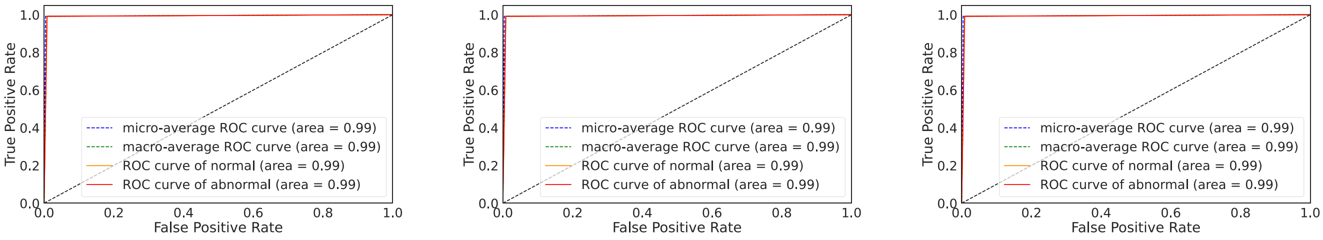
Fig. 6. Model accuracy values per clients for 15 distributed clients over (a) 25 training epochs; (b) 50 training epochs; and (c) 200 training epochs.

traffic data collected from a simulated environment, covering different types of attacks and normal activities. Each data instance includes a comprehensive set of features such as duration, protocol type, service, and flags, providing valuable insights into network behavior. Furthermore, the dataset offers a diverse range of attack scenarios, including Denial of Service



(a) (b) (c)

Fig. 7. Confusion matrices for 15 distributed clients over (a) 25 training epochs; (b) 50 training epochs; and (c) 200 training epochs.



(a) (b) (c)

Fig. 8. ROC Curves for 15 distributed clients over (a) 25 training epochs; (b) 50 training epochs; and (c) 200 training epochs.

(DoS), User to Root (U2R), Remote to Local (R2L), and Probing attacks, making it suitable for training and evaluating intrusion detection models. In our simulated environment (see Table I), we explored a range of configurations by varying the number of clients from 5 to 15 and adjusting the number of epochs from 25 to 200. For encoding, we employed the angle embedding technique with 5 qubits. We also utilized the Adam optimizer and binary entropy as the loss function for training the model. The evaluation of our proposed framework involved an extensive examination using several performance metrics, including accuracy, F1 score, receiver operating characteristic (ROC) curves, and a comprehensive confusion matrix. The ROC curve is a graphical plot that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied. It is created by plotting the true positive rate (TPR) against the false positive rate (FPR), while the Area Under the Curve (AUC) is the area under the ROC curve and provides an aggregate measure of performance across all classification thresholds. To evaluate QFL-IDS, we used the following metrics:

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \tag{12}$$

$$\text{Precision} = \frac{TP}{TP+FP} \tag{13}$$

$$\text{Recall (Sensitivity)} = \frac{TP}{TP+FN} \tag{14}$$

$$\begin{aligned}\text{F1 Score} &= 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \\ &= \frac{2TP}{2TP+FP+FN}\end{aligned} \tag{15}$$

where TP (True Positives) indicate network attacks that are correctly classified as intrusions, FN (False Negatives) indicate network attacks that are identified as normal samples, FP (False Positives) indicate normal samples that are classified as network attacks, and TN (True Negatives) indicate normal samples that are classified as normal ones.

TABLE II
PERFORMANCE METRICS OF OUR PROPOSED FRAMEWORK AND CURRENT AI-BASED METHODS USING $NSL-KDDTest^+$

Methods	Accuracy	Precision	Recall	F1	Time (second)
J48 Algorithm [21]	0.81	N/A	N/A	N/A	N/A
Naïve Bayes classifie [21]	0.76	N/A	N/A	N/A	N/A
Random forest [21]	0.80	N/A	N/A	N/A	N/A
Multilayer perceptron [21]	0.77	N/A	N/A	N/A	N/A
Support Vector Machine [21]	0.70	N/A	N/A	N/A	N/A
Character-LevelCNN-IDS [22]	0.85	0.91	0.81	0.86	N/A
ResNet50 [23]	0.79	0.91	0.69	0.79	N/A
GoogleNet [23]	0.77	0.91	0.65	0.76	N/A
Recurrent Neural Network (RNN) [24]	0.83	N/A	0.83	N/A	5516
Autoencoder-SVM-IDS [25]	0.84	0.96	0.76	0.85	673.031
QFL-IDS	**0.98**	**0.98**	**0.98**	**0.98**	**14.21**

Figs. 1, 2, and 3 show our obtained QFL-IDS loss values per client across varying numbers of training epochs (from 25 to 200), with 5, 10, and 15 distributed clients respectively. The loss values for each client consistently decrease over time, reaching a minimum, indicating rapid learning in detecting attacks without compromising the data privacy of other clients. Figs. 4, 5, and 6 show QFL-IDS model accuracy value per client across varying numbers of QFL training epochs (from 25 to 200), with 5, 10, and 15 distributed clients respectively. We observe that the accuracy of our proposed QFL-IDS framework increases, reaching a maximum of 98%. This shows that the federated clients learn from each other while preserving privacy, leading to a high accuracy score in the implemented model. This shows the effectiveness of our proposed framework in collaborative learning while preserving privacy. Figs. 7(a), 7(b), and 7(c) show the confusion matrix curves of our framework, which shows its performance across 25, 50, and 200 QFL training epochs, respectively. Moreover, Figs. 8(a), 8(b), and 8(c) show the ROC curves of QFL-IDS over 25, 50, and 200 QFL training epochs, respectively, providing, while Table II shows the values of the metrics (*i.e.*, accuracy, precision, recall, F1 score, and training time) of our proposed QFL-IDS and current AI-based methods, using the $NSL-KDDTest^{+}$ dataset. We observe that QFL-IDS achieves the highest accuracy of 98% and the highest F1 score of 98% with only 14.21 seconds of training time. This results in a time savings of over 99% compared to the RNN method. Thus, Our proposed QFL-IDS framework demonstrates not only high accuracy in intrusion detection but also significant time efficiency, making it a promising solution for efficient and secure consumer networks.

V. Conclusion

In this paper, we have introduced a novel framework, called QFL-IDS, that uses QFL to ensure an efficient, robust, and privacy-preserving approach for detecting network intrusions in consumer networks. Leveraging the decentralized nature of FL, our framework enables multiple devices to collaboratively train a global intrusion detection model while preserving the privacy of individual user data. We demonstrated the efficacy of our framework through extensive experiments using the NSL-KDD public dataset, which shows significant improvements in accuracy by 15.29%, F1 Score by 13.95%, and computational efficiency by 99% compared to the current traditional AI-based approaches. Our results confirm the potential of our proposed QFL-IDS as a promising solution for efficient and privacy-preserving network intrusion detection in consumer networks.

References

[1] "Cybercrime damages $6 trillion by 2021." 2020. Accessed: Mar. 31, 2024. [Online]. Available: https://www.cybersecurityventures.com/hackerpocalypse-cybercrime-report-2016/

[2] M. Yamauchi, Y. Ohsita, M. Murata, K. Ueda, and Y. Kato, "Anomaly detection in smart home operation from user behaviors and home conditions," *IEEE Trans. Consum. Electron.*, vol. 66, no. 2, pp. 183–192, May 2020.

[3] S. Mumtaz, A. Alsohaily, Z. Pang, A. Rayes, K. F. Tsang, and J. Rodriguez, "Massive Internet of Things for industrial applications: Addressing wireless IIoT connectivity challenges and ecosystem fragmentation," *IEEE Ind. Electron. Mag.*, vol. 11, no. 1, pp. 28–33, Mar. 2017.

[4] M. Wazid, A. Kumar Das, and S. Shetty, "BSFR-SH: Blockchain-enabled security framework against ransomware attacks for smart healthcare," *IEEE Trans. Consum. Electron.*, vol. 69, no. 1, pp. 18–28, Feb. 2023.

[5] M. K. Hasan et al., "Federated learning for computational offloading and resource management of vehicular edge computing in 6G-V2X network," *IEEE Trans. Consum. Electron.*, vol. 70, no. 1, pp. 3827–3847, Feb. 2024.

[6] J. Huang et al., "Incentive mechanism design of federated learning for recommendation systems in MEC," *IEEE Trans. Consum. Electron.*, vol. 70, no. 1, pp. 2596–2607, Feb. 2024.

[7] S. Anbalagan, G. Raja, S. Gurumoorthy, R. D. Suresh, and K. Ayyakannu, "Blockchain assisted hybrid intrusion detection system in autonomous vehicles for industry 5.0," *IEEE Trans. Consum. Electron.*, vol. 69, no. 4, pp. 881–889, Nov. 2023.

[8] P. Verma, N. Bharot, J. G. Breslin, D. O'Shea, A. Vidyarthi, and D. Gupta, "Zero-day guardian: A dual model enabled federated learning framework for handling zero-day attacks in 5G enabled IIoT," *IEEE Trans. Consum. Electron.*, vol. 70, no. 1, pp. 3856–3866, Feb. 2024.

[9] C. Qiao, M. Li, Y. Liu, and Z. Tian, "Transitioning from federated learning to quantum federated learning in Internet of Things: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, early access, May 10, 2024, doi: 10.1109/COMST.2024.3399612.

[10] J. Smith and A. Doe, "Leveraging quantum computing for security applications," *J. Quant. Comput. Cybersecurity*, vol. 5, no. 3, pp. 234–245, 2022.

[11] M. Chen, S. Mao, and Y. Liu, "Big data: A survey," *Mobile Netw. Appl.*, vol. 19, no. 2, pp. 171–209, 2019.

[12] Y. Chen, "Quantum dilated convolutional neural networks," *IEEE Access*, vol. 10, pp. 20240–20246, 2022.

[13] A. Safari and A. A. Ghavifekr, "Quantum neural networks (QNN) application in weather prediction of smart grids," in *Proc. 11th Smart Grid Conf. (SGC)*, 2021, pp. 1–6.

[14] T. D. Manjunath and B. Bhowmik, "Quantum-enhanced deep Q learning with parametrized quantum circuit," in *Proc. IEEE 4th Int. Conf. VLSI Syst., Archit., Technol. Appl. (VLSI SATA)*, 2024, pp. 1–6.

[15] K. Blekos et al., "A review on quantum approximate optimization algorithm and its variants," *Phys. Rep.*, vol. 1068, pp. 1–66, Jun. 2024. [Online]. Available: http://dx.doi.org/10.1016/j.physrep.2024.03.002

[16] M. S. Salek et al., "A novel hybrid quantum-classical framework for an in-vehicle controller area network intrusion detection," Aug. 2023, Preprint. [Online]. Available: http://dx.doi.org/10.36227/techrxiv.21907443.v2

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

[18] M. Kalinin and V. Krundyshev, "Security intrusion detection using quantum machine learning techniques," *J. Comput. Virol. Hack Tech.*, vol. 19, pp. 125–136, Mar. 2023. [Online]. Available: https://doi.org/10.1007/s11416-022-00435-0

[19] H. Suryotrisongko and Y. Musashi, "Evaluating hybrid quantum-classical deep learning for cybersecurity botnet DGA detection," *Procedia Comput. Sci.*, vol. 197, pp. 223–229, Jan. 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050921023590

[20] A. Gouveia and M. Correia, "Towards quantum-enhanced machine learning for network intrusion detection," in *Proc. IEEE 19th Int. Symp. Netw. Comput. Appl. (NCA)*, 2020, pp. 1–8.

[21] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *Proc. IEEE Symp. Comput. Intell. Security Defense Appl.*, 2009, pp. 1–6.

[22] S. Z. Lin, Y. Shi, and Z. Xue, "Character-level intrusion detection based on convolutional neural networks," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, 2018, pp. 1–8.

[23] Z. Li, Z. Qin, K. Huang, X. Yang, and S. Ye, "Intrusion detection using convolutional neural networks for representation learning," in *Proc. Neural Inf. Process.*, 2017, pp. 858–866.

[24] C. Yin, Y. Zhu, J. Fei, and X. He, "A deep learning approach for intrusion detection using recurrent neural networks," *IEEE Access*, vol. 5, pp. 21954–21961, 2017.

[25] M. Al-Qatf, Y. Lasheng, M. Al-Habib, and K. Al-Sabahi, "Deep learning approach combining sparse autoencoder with SVM for network intrusion detection," *IEEE Access*, vol. 6, pp. 52843–52856, 2018.

[26] "PennyLane," 2024. [Online]. Available: https://pennylane.ai/