SoK Paper: Security Concerns in Quantum Machine Learning as a Service

Satwik Kundu
The Pennsylvania State University
USA
sxk6259@psu.edu

Swaroop Ghosh
The Pennsylvania State University
USA
szg212@psu.edu

Abstract

Quantum machine learning (QML) is a category of algorithms that uses variational quantum circuits (VQCs) to solve machine learning tasks. Recent works have shown that QML models can effectively generalize from limited training data samples. This capability has led to an increased interest in deploying these models to address practical, real-world problems, resulting in the emergence of Quantum Machine Learning as a Service (QMLaaS). QMLaaS represents a hybrid model that utilizes both classical and quantum computing resources. Classical computers play a crucial role in this setup, handling initial pre-processing and subsequent post-processing of data to compensate for the current limitations of quantum hardware. Since this is a new area, very little work exists to paint the whole picture of QMLaaS in the context of known security threats in the domain of classical and quantum machine learning. This SoK paper is aimed to bridge this gap by outlining the complete QMLaaS workflow, which includes both the training and inference phases and highlighting security concerns involving untrusted classical and quantum providers. QML models contain several sensitive assets, such as the model architecture, training data, encoding techniques, and trained parameters. Unauthorized access to these components could compromise the model's integrity and lead to intellectual property (IP) theft. We pinpoint the critical security issues that must be considered to pave the way for a secure QMLaaS deployment.

CCS Concepts

• **Computer systems organization** → **Quantum computing**; **Neural networks**.

Keywords

Quantum machine learning, training, untrusted providers, security.

ACM Reference Format:
Satwik Kundu and Swaroop Ghosh. 2024. SoK Paper: Security Concerns in Quantum Machine Learning as a Service. In *International Workshop on Hardware and Architectural Support for Security and Privacy 2024 (HASP '24), November 02, 2024, Austin, TX, USA.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3696843.3696846

HASP '24, November 02, 2024, Austin, TX, USA

ACM ISBN 979-8-4007-1221-0/24/11
https://doi.org/10.1145/3696843.3696846

1 Introduction

Quantum computing is rapidly progressing, with companies like Atom Computing and IBM recently unveiling the largest quantum processors ever developed, featuring 1,225 and 1,121 qubits, respectively [14, 22]. The significant interest in quantum computing among academic and research communities is due to its potential to offer substantial computational speedups over classical computers for certain problems. Researchers have already begun using these noisy intermediate-scale quantum (NISQ) machines to demonstrate practical utility in this pre-fault-tolerant era [33]. Within this emergent field, quantum machine learning (QML) has also gained considerable attention, merging the power of quantum computing with classical machine learning algorithms. QML explores the potential of improving learning algorithms by leveraging the unique capabilities of quantum computers, opening new horizons in computational speed and capability. Several QML models have been explored, including quantum support vector machines (QSVMs) [52], quantum generative adversarial networks (QGANs) [16], and quantum convolutional neural networks (QCNNs) [15]. However, quantum neural networks (QNNs) [1, 18, 26, 55, 58] stand out as the most notable development, mirroring the structure and function of classical neural networks within a quantum framework.

Training QML models effectively requires integration of both quantum and classical computing resources. Currently, NISQ devices are limited by factors such as qubit count, noise levels, fidelity, and quantum volume. For instance, a quantum computer with 100 qubits is unlikely to reliably run a 100-qubit QML circuit due to inherent noise limitations. To mitigate these limitations, classical techniques are often employed beforehand to preprocess and reduce the size of input data (images or features). This preprocessing ensures that the QML circuit can execute more reliably on the quantum hardware to perform the necessary computations. Furthermore, during the QML training process, although there are quantum-native techniques available for calculating gradients of the parameters, such as the parameter-shift rule [57] and simultaneous perturbation stochastic optimization (SPSA) [61, 74], the task of final parameter optimization still relies on classical optimizers. This reliance is primarily due to the challenges associated with implementing optimization routines directly on quantum computers. Additionally, once the quantum circuit has been executed on the hardware, the measured outputs generally require further classical processing. This may involve additional computational layers or post-processing steps executed on classical computers to render the quantum computation outputs useful and interpretable.

With the rise of quantum computing access mainly provided through the cloud by various startups and companies, the transition to hosting quantum circuits, including QML models, over the cloud,

referred to as QMLaaS (Quantum Machine Learning as a Service) is imminent. For QMLaaS to function effectively, it is crucial to ensure secure communication between classical and quantum resources. However, this interdependence exposes the QMLaaS framework to increased risk of adversarial threats from both the classical and quantum domains [24, 39, 73]. Untrusted classical cloud providers can jeopardize various assets such as raw training/testing data and the final outputs of QML models, potentially leading to adversarial attacks like model inversion and inference attacks. Similarly, untrusted quantum cloud providers may threaten quantum-specific assets, including the QML architecture and novel state preparation circuit. They could also reroute the QML model's execution to compromised or low-quality hardware, compromising the confidentiality, integrity, and availability of these models. Consequently, it is crucial to conduct thorough studies to assess these security vulnerabilities and develop innovative techniques to ensure the efficient and secure operation of future QMLaaS providers.

1.1 Why QML Models are at Risk?

Apart from the vulnerabilities of hybrid QMLaaS, QML models in general face significant security risks due to the following reasons:

- **High Training Cost:** Currently, accessing quantum computers is significantly more expensive than using classical GPUs. For instance, IBM charges $1.60 per second to access their superconducting QPUs [29], which is at least 2,300 times costlier than high-performance GPUs, priced at approximately $0.0007 per second [25]. AWS Quantum also offers access to a variety of QPUs from providers like IonQ, Rigetti, and IQM, where charges are based on both the number of tasks and the shots used [3]. QML models require hundreds of training epochs, each involving thousands of quantum circuit executions, depending on factors like the size of the training dataset, gradient calculation methods, etc. Each circuit execution involves thousands of trials to obtain expectation values, making the training and even partial training of QML models very expensive. While current state-of-the-art machine learning models, such as Gemini [53], require millions to billions of dollars for training, scaling QML models could potentially cost orders of magnitude more, thereby making them extremely valuable.
- **High Training Time:** Current state-of-the-art ML models, such as GPT-4 [2] took ~ 4+ months for training using thousands of dedicated GPUs. In contrast, quantum resources are both scarce and in high demand. This scarcity leads to long wait times for both hardware access and simulators, whose computation time scales exponentially with the number of qubits. Even users with dedicated access, such as those in Quantum Hubs with a limited number of participants, experience these delays. Consequently, training a large QML model could take a significant amount of time—potentially months to years—due to these extended queue times and the need to execute hundreds of thousands of quantum circuits.
- **Hosting QMLs on Cloud:** Since QML providers may not possess their own quantum hardware, they may rely on a third-party quantum cloud for hosting the model. This will lead to the rise of QMLaaS [34] providing access to clients only through input-output queries via external APIs. The quantum cloud provider, having white-box access to both the quantum circuit and the expensive training data, could potentially expose these assets to various threats [6, 41, 48, 65].
- **Miscellaneous intellectual property (IP):** QML models possess various forms of IP. The untrained IPs of a QML model comprise its fundamental architecture, including aspects such as entanglement strategies, the number of parameters, the layer depth, and the measurement basis. Additionally, the training data is often incorporated directly into the state preparation circuit. Trained QML IPs consist of the optimized parameters, which have been fine-tuned through training processes. These parameters, along with the input data used during inference, are also embedded within the state preparation circuit.

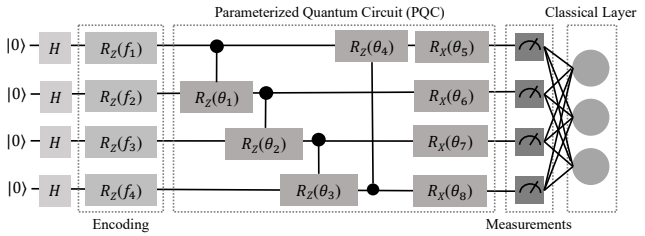

Figure 1: Architecture of a 4-qubit hybrid QNN. Classical features are encoded as angles of quantum rotation gates (R_Z). PQC transforms encoded states to explore the search space and entangle features. Measured expectation values are then fed into a classical linear layer for final prediction.

In this study, we first provide a comprehensive description of a hybrid QMLaaS framework. We discuss in detail each stage involved in the training and inference processes of a QML model within a cloud-based environment. Following this, we explore the various security vulnerabilities that could compromise the confidentiality, integrity, and availability of the hybrid QMLaaS framework. Addressing these vulnerabilities is essential for ensuring the secure and efficient operation of QMLaaS.

2 Background

2.1 Quantum Neural Network (QNN)

QNN mainly consists of three building blocks: (i) a classical to quantum data encoding (or embedding) circuit, (ii) a parameterized quantum circuit (PQC) whose parameters can be tuned (mostly by an optimizer) to perform the desired task, and (iii) measurement operations. There are a number of different encoding techniques available (basis encoding, amplitude encoding, etc.) but for continuous variables, the most widely used encoding scheme is angle encoding where a variable input classical feature is encoded as a rotation of a qubit along the desired axis [1]. As the states produced by a qubit rotation along any axis will repeat in 2π intervals, features are generally scaled within 0 to 2π (or -π to π) in a data pre-processing step. In this study, we consider *RZ* gates to encode classical features into their quantum states.

A PQC consists of a sequence of quantum gates whose parameters can be varied to solve a given problem. In QNN, the PQC is the primary and only trainable block to recognize patterns in data. The PQC is composed of entangling operations and parameterized single-qubit rotations. The entanglement operations are a set of multi-qubit operations (that may or may not be parameterized) performed between all of the qubits to generate correlated states and the parametric single-qubit operations are used to search the solution space. Finally, the measurement operation causes the qubit state to collapse to either '0' or '1'. We used the expectation value of Pauli-Z to determine the average state of the qubits. The measured values are then fed into a classical neuron layer (the number of neurons is equal to the number of classes in the dataset) in our hybrid QNN architecture as shown in Fig. 1, which performs the final classification task. Other QML architectures may directly apply a softmax function to the measured qubit values or pass them through multiple classical layers for further processing.

2.2 Quantum Cloud Services

Recently, there has been a significant increase in the number of companies offering cloud access to quantum hardware. IBM, which employs superconducting transmon qubits for their quantum processing units (QPUs), has recently removed their lower-qubit devices from cloud access [29]. They now only offer hardware ranging from 127-qubit to 156-qubit systems, with an error rate per layer of gates as low as 0.6%. Rigetti, also using superconducting qubits for their quantum processors, has recently started providing cloud access to their 84-qubit Ankaa-2 quantum hardware [54]. This system is known for its higher coherence times and fidelities. Oxford Quantum Circuits (OQC) offers access to up to 32-qubit quantum hardware, which operates on superconducting qubits within a coaxmon architecture [47]. Their OQC Toshiko Gen 1 machine boasts over 96% 2-qubit gate fidelity and is recognized as the world's first enterprise-ready platform. IQM provides access to their 20-qubit IQM Radiance, which is expected to be upgradeable to 150-qubit configurations in the near future [31]. QuEra's Aquila was the first and remains the only publicly accessible 256-qubit neutral atom quantum computer [76]. It is based on programmable arrays of neutral rubidium atoms, trapped in a vacuum by tightly focused laser beams. Xanadu, known for developing the popular PennyLane framework, offers cloud access to their X-Series devices [77]. These are the first photonic quantum computers deployed to the cloud. IonQ provides cloud access to their trapped-ion quantum computers, which achieve 2-qubit fidelity of up to 99.6% [30].

3 Training in QMLaaS

Fig. 2 presents a detailed workflow of QMLaaS. Training QML models on cloud-based quantum hardware involves a multi-step process, combining classical and quantum computing techniques. The methodology consists of several key stages, integrating classical pre-processing and QNN processing, forming a hybrid system optimized for machine learning tasks.

3.1 (Step-1) Data Pre-Processing

Due to the qubit limitations of current quantum hardware, raw training data must first be pre-processed in a classical cloud to effectively train QML models on large datasets. Thus, upon receiving the input data, the first stage of processing occurs in the classical cloud, encompassing the following steps:

- *Dimensionality Reduction:* This involves reducing the input data dimension to match the qubit capacity of the quantum hardware. For image classification tasks, this reduction may involve resizing or applying dimensionality reduction techniques such as Principal Component Analysis (PCA) [69], Linear Discriminant Analysis (LDA), etc. to extract essential features for classification. Non-linear dimensionaltiy reduction techniques like the convolutional autoencoder (CAE) can also be used which has been found to outperform PCA especially for image classification tasks using hybrid QNNs.
- *Normalization:* Next, these extracted features must then be normalized before training. Normalization is crucial because, during the encoding step, features are often passed as rotation angles of quantum gates, unnormalized values can cause features of different classes to appear identical to the QNN if their values differ by multiples of 2π. There exists several normalization techniques which can be used but few of the most widely used techniques are min-max scaling and max absolute scaling.

The output is a reduced and normalized data matrix X, where each element represents a feature of the input data.

3.2 (Step-2) Design and Encode

The reduced and normalized data X is then encoded into a quantum-compatible format using a data encoding circuit $D(X)$. This circuit transforms classical data into quantum states through various techniques such as angle encoding, amplitude encoding, etc. Another critical component of QML circuit involves designing an optimal PQC, which is the primary trainable block of QML models. Ideally, PQCs with a higher number of parametric gates should lead to better performance due to their increased expressive power. However, in reality, increasing the number of gates, including both 1 and 2-qubit gates, often leads to a higher error rate. This is primarily due to factors such as increased decoherence error, gate errors, and the need for swap gate insertions. As a result, using a brute force approach to select the correct QML circuit for specific quantum hardware may not be effective. In order to address these challenges, recent works have introduced efficient quantum circuit search frameworks [4, 17, 27, 68]. These frameworks are designed to identify the most performant circuit for a given quantum hardware setup. The search techniques employed are noise-guided, topology-aware, and data embedding-aware, which have collectively been shown to enable superior performance over traditional circuit design methods. Once the optimal circuit design is identified, it is sent to the cloud provider for execution.

3.3 (Step-3) Transpile and Map

Upon receiving the QML circuit, it undergoes transpilation and logical to physical qubit mapping to match the basis gates and topology of the cloud quantum hardware. This crucial step ensures

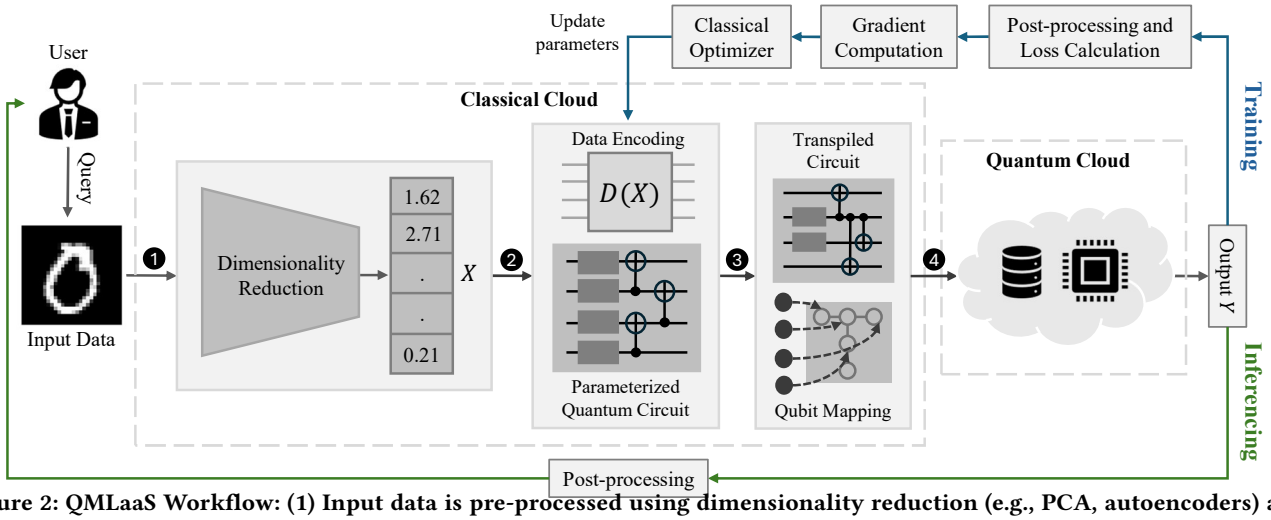

Figure 2: QMLaaS Workflow: (1) Input data is pre-processed using dimensionality reduction (e.g., PCA, autoencoders) and normalized for effective QML training. (2) The reduced features are encoded into a quantum circuit, and a suitable PQC is selected. (3) The circuit is transpiled to match the quantum hardware's topology and basis gates. (4) The circuit is sent to a quantum cloud provider for execution. Training: Post-processing of measured outputs, loss calculation, and parameter updates are performed using a classical optimizer. Inferencing: Outputs are post-processed to return the final vector/label to the user.

the circuit can be efficiently executed on the specific quantum system in use. The transpilation process involves several detailed steps to optimize the quantum circuit and ensure its compatibility with the hardware. These steps include [28]:

- *Virtual Circuit Optimization:* Simplifying the circuit at a virtual level before mapping it to physical qubits.
- *Decomposition of 3+ Qubit Gates:* Breaking down more complex multi-qubit gates into simpler 1- and 2-qubit gates.
- *Placement on Physical Qubits:* Assigning logical qubits from the virtual circuit to the physical qubits available on the hardware.
- *Routing on Restricted Topology:* Adjusting the circuit to fit the specific qubit connectivity of the hardware.
- *Translation to Basis Gates:* Converting the circuit's gates into the set of native gates supported by the quantum processor.
- *Physical Circuit Optimization:* Further refining the circuit to minimize errors and enhance performance after placement and routing.

It is important to note that all these processes are performed on a classical computer. This transpilation can be either automated and completed beforehand by the user or handled by the quantum cloud provider. If done beforehand, the transpiled circuit is directly sent for execution; otherwise, it is transpiled by the provider before execution. This ensures that the quantum circuit can be efficiently executed on the cloud quantum hardware.

3.4 (Step-4) Execute and Measure

Finally, the transpiled quantum circuit is sent to the quantum cloud provider for execution on the chosen quantum hardware. Typically, the circuit execution job is added to a queue for the public quantum hardware, as a large number of users are using the machine. Once the job reaches the front of the queue, it is executed on the quantum hardware and the required qubits are measured. There are a variety of measurement techniques used to measure qubits like the basis measurement, Pauli measurement (X, Y, Z), quantum state tomography etc. The raw measured values obtained are then subjected to post-processing (classical), which is essential for their practical usage. Depending on the architecture of the Hybrid QML model, this post-processing might include operations like the softmax function, which normalizes the output probabilities, or even integration with classical layers, such as linear layers. This step is crucial for calculating the loss needed for the training and optimization process.

3.5 (Step-5) Gradient Calculation

To update the parameters of the QML circuit, the gradient of the parameters needs to be calculated. Unlike classical neural networks, backpropagation is not feasible on quantum computers due to the No-Cloning Theorem [12, 75], which prohibits copying intermediate quantum states for use in a backward pass. As a result, alternative techniques such as the parameter-shift rule [45, 57] and finite differences are employed to calculate gradients on quantum hardware. The parameter-shift rule, for instance, is quite resource-intensive. To calculate the gradients of n parameters, it requires $2n$ circuit executions. Due to the high computational demands of these methods, researchers have begun exploring more efficient approaches, such as Simultaneous Perturbation Stochastic Approximation (SPSA) [62, 63]. SPSA is a zero-order gradient estimation technique that significantly reduces computational overhead; it requires only 2 circuit executions to estimate the gradients of all parameters. However, while SPSA offers a dramatic reduction in the number of required circuit executions, it comes with a trade-off: the gradients it produces are noisier compared to those obtained through methods like the parameter shift.

3.6 (Step-6) Parameter Optimization

After calculating the gradients and determining the loss function, a classical optimizer is employed to update the circuit parameters. This optimization step aims to minimize the loss function, thereby improving the model's performance on the training data. Empirically, it has been found that optimizers like Adam and AMSGrad work well with SPSA for optimizing quantum circuits executed in a noisy environment [74]. Steps 1-5 are iteratively executed until the QML model achieves the required accuracy or a predefined loss threshold is met.

4 Inferencing in QMLaaS

4.1 Hosting QML in Quantum-Classical Cloud

In the QMLaaS framework, the deployment and inferencing process leverages both classical and quantum computing resources to efficiently process and analyze data. To host a trained QML model on the cloud and provide access via an API, first, a suitable classical cloud platform, such as AWS, Google Cloud, or Azure, and a quantum cloud platform like IBM Quantum or OQC, should be selected, such that they support both machine learning and quantum circuit deployments. The trained dimensionality reduction model and the QML circuit need to be packaged in a format compatible with the chosen platforms. Next, a cloud instance should be set up, or a managed service like AWS Lambda, Google Cloud Functions, or Azure Functions can be used to deploy these models. Once deployed, an API endpoint, created using frameworks like Flask or cloud-specific services such as AWS API Gateway, will manage incoming requests. The API processes the data through the dimensionality reduction model, encodes the pre-processed data into the trained QML model, and then performs transpilation to optimize the quantum circuit for specific quantum hardware based on a pre-defined algorithm before sending it to the quantum cloud provider for execution. Finally, the circuit is executed on the designated quantum cloud hardware and the measured values are post-processed to get the final output.

4.2 Inference Operation

The workflow initiates when a user submits a query accompanied by input data, such as an image of a handwritten digit. Initially, the data undergoes classical pre-processing in the cloud, which includes dimensionality reduction using techniques like PCA or t-SNE, and normalization to scale the data suitably for quantum processing. The pre-processed data is then encoded into QML model using the encoding technique used while training. Once encoded, the circuit is transpiled and sent to the quantum cloud where the trained QML model is executed on available quantum hardware, such as IBM Quantum Experience or Rigetti Aspen.

Post-execution, the quantum results are sent back to the classical cloud for post-processing. This includes transforming the raw quantum outputs, often probability distributions or measurements, into meaningful classical information through techniques like softmax or linear transformations. The final processed results are then delivered to the user, providing relevant outputs such as classifications or predictions based on the original query.

5 Security Concerns

5.1 Assets in QMLaaS

5.1.1 Training/Testing Data. Data used for training QML models or during inferencing are critical assets because they are often highly sensitive, difficult to obtain, and expensive to acquire and process [32]. In a hybrid QMLaaS framework, this data may be processed locally or over the cloud for tasks like dimensionality reduction, making it vulnerable to threats such as data theft attacks [44, 59]. Sensitive data, including personal health or financial records, must be handled securely to prevent privacy breaches and legal complications. Moreover, acquiring high-quality data is particularly challenging in specialized domains, requiring significant effort, time, and adherence to regulatory standards. The cost associated with collecting, labeling, and preparing this data further adds to its value, as it directly influences the performance and reliability of machine learning models, making it a sensitive and valuable asset.

5.1.2 Data Encoding Circuit. The data encoding circuit in a QML model is used for embedding classical data into its corresponding quantum state, making it one of the crucial components of any QML model. Selecting the optimal data encoding circuit is a challenging task, as it directly influences the performance of QML models [7, 37, 38, 56]. This process often requires extensive evaluation of different encoding strategies on noisy quantum hardware to identify the most suitable circuit for a given system. Techniques such as Quantum Circuit Search (QCS) [4] have also been employed to optimize the choice of encoding circuits for QML models. The process is further complicated when encoding sensitive or expensive private data, which adds significant value to the data encoding component of the QML model. As a result, if an adversary gains access to this circuit, it could pose a serious threat to the confidentiality and integrity of the QML model.

5.1.3 PQC Architecture. During training, the parameters of the quantum gates within the PQC are iteratively optimized to minimize a loss function, enabling the model to perform its designated task. Designing an optimal PQC is also a complex and resource-intensive process, as it requires careful consideration of factors such as expressibility, entanglement capability, and the reachability of the quantum states [8, 13, 60]. Additionally, the PQC must be tailored to the specific quantum hardware, accounting for noise levels, available basis gates, and the device's topology [4, 68]. However, designing an effective PQC goes beyond these considerations, especially given the problem of the barren plateau, which can hinder optimization when the PQC is too deep, highly entangled, or overly expressive [36, 43, 49]. Interestingly, recent studies suggest that reducing entanglement in PQCs can actually improve performance, making the design of a robust PQC even more nuanced [10]. Furthermore, the intermediate parameter values during training and final parameter values during inferencing can be considered as assets since they take significant time and cost to obtain. Given the considerable time, resources, and expertise required to develop a well-functioning PQC, it is also considered a valuable asset in the realm of QML.

5.2 Adversary Motivation

Adversary will be motivated to steal the QNN and/or its assets to avoid paying for (i) the time and resources needed to design a QNN from scratch, (ii) the training/inferencing data and (iii) the time and resources needed for training the model. Thus, even though QMLaaS provides easier access to wider variety of quantum hardwares and architectures, it also opens up several security vulnerabilities. In the following section we will discuss few of the major security concerns (Fig. 3) which comes with QMLaaS and how it affects the confidentiality, integrity and availability of the QML models.

5.3 Confidentiality

5.3.1 Threats from Classical Cloud. Similar to traditional Machine Learning as a Service (MLaaS), there is a risk of raw data theft, either during training or inferencing [59]. This risk pertains to sensitive data sent for preprocessing on classical cloud systems by untrusted providers especially during the training stage when both data and labels are sent to train the feature extraction model. Even during inference, although the classical cloud may only have access to input data and not the labels, adversaries can employ techniques such as clustering to reverse engineer the original labels or collude with the cloud provider responsible for loss calculation to obtain the labels. Stealing sensitive training data can provide adversaries with confidential business insights and personally identifiable information, which can be exploited for financial gain or competitive advantages.

5.3.2 Threats from Quantum Cloud. In QMLaaS, this threat extends to quantum-encoded data, which becomes vulnerable when handled by untrusted quantum cloud providers. An adversary could potentially extract the encoding circuit from QML models and use it to either sell the circuit or train their own QML model, thereby offering a similar service [72]. However, training their own QML model would either require collusion with cloud provider responsible for loss calculation or further analysis like majority voting, to extract the labels, as the adversary would not have direct access to them. This dual threat underscores the unique security risks associated with both the classical and quantum components of the QMLaaS framework.

Furthermore, QML models that incorporate novel encoding techniques and architectures are particularly vulnerable to threats from potentially untrusted quantum cloud-based adversaries. Given that these cloud providers would have white-box access to the QML circuits, there is a risk of intellectual property (IP) theft. Such access enables adversaries to steal these specialized architectures and encoding techniques, which they could then potentially sell to competing businesses. Similarly, trained QML models hosted on these untrusted platforms are also at risk of being stolen, highlighting the confidentiality and security challenges faced in the QMLaaS framework. This situation underscores the critical need for robust security measures to protect against the theft of both data and intellectual assets [35, 40, 71].

QML circuits in a cloud-based quantum computing environment are also threatened by physical attacks, especially as users lack direct control over the hardware. As quantum computers increasingly handle sensitive IP through complex algorithms, the risk of these circuits being compromised grows. Malicious insiders in data

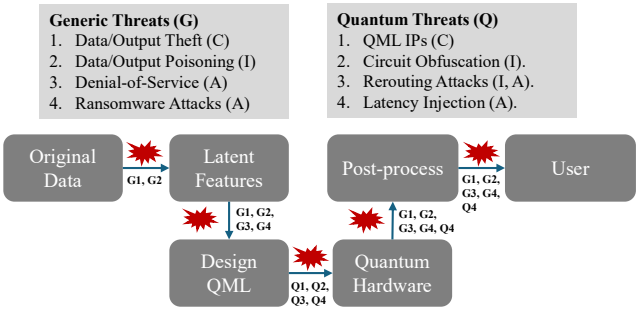

Figure 3: Key Threats to Confidentiality (C), Integrity (I), and Availability (A) in the QMLaaS Pipeline.

centers could execute power-based side-channel attacks to extract information about the control pulses used in quantum operations [79]. By analyzing these pulses, attackers can reverse-engineer the gate-level description of the QML circuits, revealing the underlying algorithms or sensitive data embedded within the circuits. Such attacks can compromise the confidentiality of proprietary quantum algorithms and data.

QML circuits are also at significant risk due to state leakage, particularly arising from noisy and erroneous reset operations necessary between circuit executions [80]. When these operations are flawed, residual quantum information from previous executions can persist and carry over to subsequent ones, leading to "horizontal" leakage. This leakage allows attackers to infer sensitive quantum states used in a victim's QML circuits. Additionally, "vertical" leakage, occurring simultaneously between qubits due to issues like crosstalk, further compromises confidentiality by allowing adversaries to extract information from multiple qubits within the same execution.

Finally, even when trained QML models are hosted on trusted cloud providers, they remain susceptible to external threats, such as model stealing or model inversion attacks. Consider a scenario where the cloud-hosted QML model operates as a black box—users do not have access to information about the model's architecture or the dataset on which it was trained, only the input and output data format. In this setup, an adversary could systematically query the cloud-hosted model, gathering significant information that could be used to replicate the model's functionality or extract details about the training data [20, 21, 34].

5.4 Integrity

5.4.1 Threats from Classical Cloud. The raw training/ testing data sent to the classical cloud for dimensionality reduction is vulnerable to threats such as data poisoning attacks [19, 23, 46, 70, 81]. Adversary could either tamper or introduce adversarial examples to the original raw data. Such manipulations can corrupt the data before it is even encoded into quantum format, compromising the QML model's reliability from the outset.

5.4.2 Threats from Quantum Cloud. The integrity of QML models is particularly at risk in the hybrid cloud-based QMLaaS framework due to a variety of factors that go beyond classical data integrity issues. These include challenges specific to quantum data and quantum circuits. Especially, once the data is encoded into quantum

circuits, it faces additional threats. Adversarial obfuscation of the quantum encoded circuit can severely degrade the model's performance. This issue is particularly critical for the primary trainable block of the QML model, the PQC. Adversaries may attempt to manipulate the circuit architecture or even tamper the parameters or the measurement outputs, leading to significant performance degradation [66, 67].

Furthermore, an adversary could exploit crosstalk between qubits to launch a fault injection attack on a victim's QML circuit execution in multi-tenant computing environment [5]. By continuously operating their own qubits with quantum gates, such as CNOT, the adversary can induce errors in the neighboring qubits used by the victim. This interference degrades the accuracy and reliability of the victim's computational results due to crosstalk. Attackers can also compromise the integrity of QML models by targeting various components within the quantum computing system, including the QPU, Quantum Computer Controller, and Classical Co-processor [78]. By manipulating physical qubits and couplings through voltage changes or electromagnetic radiation, attackers can introduce faults that alter quantum operations. They can also interfere with the analog control pulses, modifying their frequency, phase, or envelope to induce errors in gate operations. Furthermore, attacks on the digital specifications used to generate these pulses, as well as classical registers that store critical data, can lead to incorrect quantum operations and corrupted outputs. In the Classical Co-processor, similar attacks on classical registers can distort the computations and optimizations in QML, ultimately compromising the model's accuracy and reliability.

Another concern arises from the variety of available quantum hardware. An adversary operating within the cloud infrastructure might allocate lower-quality quantum hardware for executing QML circuits [50]. This allocation strategy might be motivated by the lower costs associated with running circuits on less capable hardware. However, this not only compromises the integrity of the QML models but also negatively affects their performance. Each of these factors underscores the complex integrity challenges faced by QML models in a cloud-based, hybrid quantum-classical environment, necessitating robust strategies to ensure the security and reliability of these systems.

5.5 Availability

5.5.1 Threats from Classical Cloud. When pre-/post-processing data over classical cloud, several security concerns related to availability can rise as well. For instance, denial of service (DoS) attacks could be targeted at the classical computing resources like CPUs and GPUs, effectively disrupting training/inferencing process and making computational resources inaccessible [9, 42]. Another critical concern is ransomware attacks where malicious adversaries can encrypt sensitive pre-/post-processed data or computational resources, demanding payment for access restoration. Such attacks not only halt model training but could also result in loss of valuable data [11, 64].

5.5.2 Threats from Quantum Cloud. Adversaries based in the quantum cloud could also launch DoS attacks, disrupting the availability of quantum hardware, or they could withhold measured outputs from quantum hardware, leading to ransomware attacks. Furthermore, as discussed earlier, the QMLaaS workflow involves using both classical and quantum resources during the training and inference stages. This dependency creates a latency issue, which can be exploited by a cloud-based adversary to delay the training and inference processes. For example, during the training phase, an adversary can reroute quantum circuit executions to slower or more congested quantum hardware. This intentional rerouting to cheaper hardware with longer queue times can significantly delay the gradient calculation process, thereby increasing the overall training time. A similar strategy can be applied during the inference stage, where fast responses are often crucial. Thus, adversaries could introduce latency to degrade the runtime performance of the QML models, affecting the responsiveness of the service. Additionally, adversaries can also introduce an artificial demand on specific quantum hardware, creating a bottleneck. This can be achieved by submitting a large number of low-priority tasks to certain quantum processors, causing important QML tasks to experience significant delays [51]. These tactics can greatly affect the availability of QMLaaS, compromising the efficiency and reliability of the service.

6 Conclusion

The rapid development of quantum computers and the growing interest in harnessing their practical utility have sparked significant exploration in various applications, with QML being one of the most heavily researched areas. The implementation of QML models is expected to lead to the emergence of QMLaaS, a hybrid framework that leverages both classical and quantum resources to deliver QML services. In this work, we provided a detailed description of each component of the QMLaaS framework and highlighted the various security concerns inherent in this hybrid approach. Addressing these security issues will be crucial for achieving a secure and reliable QMLaaS deployment in the future.

Acknowledgments

The work is supported in parts by NSF (CNS-2129675, CCF-2210963, OIA-2040667, DGE-1821766 and DGE-2113839) and gifts from Intel.

References

[1] Amira Abbas et al. 2021. The power of quantum neural networks. *Nature Computational Science* 1, 6 (2021), 403–409.
[2] Josh Achiam et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774* (2023).
[3] Amazon. 2024. Amazon Braket Pricing. https://aws.amazon.com/braket/pricing/ Accessed: 08/12/2024.
[4] Sashwat Anagolum et al. 2024. Élivágar: Efficient quantum circuit search for classification. In *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2*. 336–353.
[5] Abdullah Ash-Saki et al. 2020. Analysis of crosstalk in nisq devices and security implications in multi-programming regime. In *Proceedings of the ACM/IEEE International Symposium on Low Power Electronics and Design*. 25–30.
[6] Ramin Ayanzadeh et al. 2023. Enigma: Privacy-Preserving Execution of QAOA on Untrusted Quantum Computers. *arXiv preprint arXiv:2311.13546* (2023).
[7] Jan Balewski et al. 2024. Quantum-parallel vectorized data encodings and computations on trapped-ion and transmon QPUs. *Scientific Reports* 14, 1 (2024), 3435.
[8] Kishor Bharti et al. 2022. Noisy intermediate-scale quantum algorithms. *Reviews of Modern Physics* 94, 1 (2022), 015004.
[9] Adrien Bonguet and Martine Bellaiche. 2017. A survey of denial-of-service and distributed denial of service attacks and defenses in cloud computing. *Future Internet* 9, 3 (2017), 43.

[10] Joseph Bowles, Shahnawaz Ahmed, and Maria Schuld. 2024. Better than classical? the subtle art of benchmarking quantum machine learning models. *arXiv preprint arXiv:2403.07059* (2024).

[11] Ross Brewer. 2016. Ransomware attacks: detection, prevention and cure. *Network security* 2016, 9 (2016), 5–9.

[12] Vladimir Bužek and Mark Hillery. 1996. Quantum copying: Beyond the no-cloning theorem. *Physical Review A* 54, 3 (1996), 1844.

[13] Marco Cerezo et al. 2021. Variational quantum algorithms. *Nature Reviews Physics* 3, 9 (2021), 625–644.

[14] Atom Computing. 2023. Quantum startup Atom Computing first to exceed 1,000 qubits. *Press Release. Accessed: Oct* 28 (2023).

[15] Iris Cong et al. 2019. Quantum convolutional neural networks. *Nature Physics* 15, 12 (2019), 1273–1278.

[16] Pierre-Luc Dallaire-Demers et al. 2018. Quantum generative adversarial networks. *Physical Review A* 98, 1 (2018), 012324.

[17] Yuxuan Du, Tao Huang, Shan You, Min-Hsiu Hsieh, and Dacheng Tao. 2022. Quantum circuit architecture search for variational quantum algorithms. *npj Quantum Information* 8, 1 (2022), 62.

[18] Edward Farhi and Hartmut Neven. 2018. Classification with quantum neural networks on near term processors. *arXiv preprint arXiv:1802.06002* (2018).

[19] Ji Feng, Qi-Zhi Cai, and Zhi-Hua Zhou. 2019. Learning to confuse: Generating training time adversarial data with auto-encoder. *Advances in Neural Information Processing Systems* 32 (2019).

[20] Matt Fredrikson et al. 2015. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*. 1322–1333.

[21] Zhenxiao Fu et al. 2024. QuantumLeak: Stealing Quantum Neural Networks from Cloud-based NISQ Machines. *arXiv preprint arXiv:2403.10790* (2024).

[22] Jay Gambetta. 2023. The hardware and software for the era of quantum utility is here.

[23] Jonas Geiping et al. 2020. Witches' brew: Industrial scale data poisoning via gradient matching. *arXiv preprint arXiv:2009.02276* (2020).

[24] Weiyuan Gong et al. 2024. Enhancing quantum adversarial robustness by randomized encodings. *Physical Review Research* 6, 2 (2024), 023020.

[25] Google. 2024. Google Cloud. https://cloud.google.com/compute/gpus-pricing#gpu-pricing Accessed: 08/12/2024.

[26] Gian Giacomo Guerreschi and Mikhail Smelyanskiy. 2017. Practical optimization for hybrid quantum-classical algorithms. *arXiv preprint arXiv:1701.01450* (2017).

[27] Yuhan Huang, Qingyu Li, Xiaokai Hou, Rebing Wu, Man-Hong Yung, Abolfazl Bayat, and Xiaoting Wang. 2022. Robust resource-efficient quantum variational ansatz through an evolutionary algorithm. *Physical Review A* 105, 5 (2022), 052414.

[28] IBM. 2023. Qiskit Transpiler Documentation. https://docs.quantum.ibm.com/api/qiskit/transpiler Accessed: 07/18/2024.

[29] IBM. 2024. IBM Quantum. https://quantum.ibm.com/ Accessed: 08/12/2024.

[30] IonQ. 2024. IonQ Quantum Cloud. https://ionq.com/quantum-cloud Accessed: 08/15/2024.

[31] IQM. 2024. IQM Radiance. https://www.meetiqm.com/products/iqm-radiance Accessed: 08/15/2024.

[32] Abhinav Jain et al. 2020. Overview and importance of data quality for machine learning tasks. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*. 3561–3562.

[33] Youngseok Kim et al. 2023. Evidence for the utility of quantum computing before fault tolerance. *Nature* 618, 7965 (2023), 500–505.

[34] Satwik Kundu et al. 2024. Evaluating Efficacy of Model Stealing Attacks and Defenses on Quantum Neural Networks. In *Proceedings of the Great Lakes Symposium on VLSI 2024*. 556–559.

[35] Satwik Kundu and Swaroop Ghosh. 2024. STIQ: Safeguarding Training and Inferencing of Quantum Neural Networks from Untrusted Cloud. *arXiv preprint arXiv:2405.18746* (2024).

[36] Martin Larocca et al. 2024. A review of barren plateaus in variational quantum computing. *arXiv preprint arXiv:2405.00781* (2024).

[37] Ryan LaRose and Brian Coyle. 2020. Robust data encodings for quantum classifiers. *Physical Review A* 102, 3 (2020), 032420.

[38] Guangxi Li et al. 2022. Concentration of data encoding in parameterized quantum circuits. *Advances in Neural Information Processing Systems* 35 (2022), 19456–19469.

[39] Haoran Liao, Ian Convy, William J Huggins, and K Birgitta Whaley. 2021. Robust in practice: Adversarial attacks on quantum machine learning. *Physical Review A* 103, 4 (2021), 042427.

[40] Chao Lu et al. 2024. Quantum Leak: Timing Side-Channel Attacks on Cloud-Based Quantum Services. *arXiv preprint arXiv:2401.01521* (2024).

[41] Yao Ma et al. 2022. QEnclave-A practical solution for secure quantum cloud computing. *npj Quantum Information* 8, 1 (2022), 128.

[42] Tasnuva Mahjabin et al. 2017. A survey of distributed denial-of-service attack, prevention, and mitigation techniques. *International Journal of Distributed Sensor Networks* 13, 12 (2017), 1550147717741463.

[43] Jarrod R McClean et al. 2018. Barren plateaus in quantum neural network training landscapes. *Nature communications* 9, 1 (2018), 4812.

[44] Fatemehsadat Mireshghallah et al. 2020. Privacy in deep learning: A survey. *arXiv preprint arXiv:2004.12254* (2020).

[45] Kosuke Mitarai, Makoto Negoro, Masahiro Kitagawa, and Keisuke Fujii. 2018. Quantum circuit learning. *Physical Review A* 98, 3 (2018), 032309.

[46] Blaine Nelson et al. 2008. Exploiting machine learning to subvert your spam filter. *LEET* 8, 1-9 (2008), 16–17.

[47] OQC: Oxford Quantum Circuits. 2024. OQC Toshiko. https://oqc.tech/tech/toshiko/ Accessed: 08/15/2024.

[48] Tirthak Patel et al. 2023. Toward privacy in quantum program execution on untrusted quantum cloud computing machines for business-sensitive quantum needs. *arXiv preprint arXiv:2307.16799* (2023).

[49] Arthur Pesah et al. 2021. Absence of barren plateaus in quantum convolutional neural networks. *Physical Review X* 11, 4 (2021), 041011.

[50] Koustubh Phalak et al. 2021. Quantum puf for security and trust in quantum computing. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* 11, 2 (2021), 333–342.

[51] Koustubh Phalak and Swaroop Ghosh. 2024. QuaLITi: Quantum Machine Learning Hardware Selection for Inferencing with Top-Tier Performance. *arXiv preprint arXiv:2405.11194* (2024).

[52] Patrick Rebentrost et al. 2014. Quantum support vector machine for big data classification. *Physical review letters* 113, 13 (2014), 130503.

[53] Machel Reid et al. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530* (2024).

[54] Rigetti. 2024. Rigetti Systems. https://qcs.rigetti.com/qpus Accessed: 08/15/2024.

[55] Maria Schuld et al. 2014. The quest for a quantum neural network. *Quantum Information Processing* 13 (2014), 2567–2586.

[56] Maria Schuld et al. 2021. Effect of data encoding on the expressive power of variational quantum-machine-learning models. *Physical Review A* 103, 3 (2021), 032430.

[57] Maria Schuld, Ville Bergholm, Christian Gogolin, Josh Izaac, and Nathan Killoran. 2019. Evaluating analytic gradients on quantum hardware. *Physical Review A* 99, 3 (2019), 032331.

[58] Maria Schuld, Alex Bocharov, Krysta M Svore, and Nathan Wiebe. 2020. Circuit-centric quantum classifiers. *Physical Review A* 101, 3 (2020), 032308.

[59] Reza Shokri and Vitaly Shmatikov. 2015. Privacy-preserving deep learning. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*. 1310–1321.

[60] Sukin Sim, Peter D Johnson, and Alán Aspuru-Guzik. 2019. Expressibility and entangling capability of parameterized quantum circuits for hybrid quantum-classical algorithms. *Advanced Quantum Technologies* 2, 12 (2019), 1900070.

[61] James C Spall. 1997. A one-measurement form of simultaneous perturbation stochastic approximation. *Automatica* 33, 1 (1997), 109–112.

[62] James C Spall. 1998. Implementation of the simultaneous perturbation algorithm for stochastic optimization. *IEEE Transactions on aerospace and electronic systems* 34, 3 (1998), 817–823.

[63] James C Spall. 1998. An overview of the simultaneous perturbation method for efficient optimization. *Johns Hopkins apl technical digest* 19, 4 (1998), 482–492.

[64] Noor Thamer and Raaid Alubady. 2021. A survey of ransomware attacks for healthcare systems: Risks, challenges, solutions and opportunity of research. In *2021 1st Babylon International Conference on Information Technology and Science (BICITS)*. IEEE, 210–216.

[65] Theodoros Trochatos et al. 2024. Dynamic Pulse Switching for Protection of Quantum Computation on Untrusted Clouds. In *2024 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*. IEEE, 404–414.

[66] Suryansh Upadhyay and Swaroop Ghosh. [n. d.]. Trustworthy and reliable computing using untrusted and unreliable quantum hardware. *Frontiers in Computer Science* 6 ([n. d.]), 1431788.

[67] Suryansh Upadhyay and Swaroop Ghosh. 2024. Obfuscating quantum hybrid-classical algorithms for security and privacy. In *2024 25th International Symposium on Quality Electronic Design (ISQED)*. IEEE, 1–8.

[68] Hanrui Wang et al. 2022. Quantumnas: Noise-adaptive search for robust quantum circuits. In *2022 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*. IEEE, 692–708.

[69] Hanrui Wang et al. 2022. Quantumnat: quantum noise-aware training with noise injection, quantization and normalization. In *Proceedings of the 59th ACM/IEEE design automation conference*. 1–6.

[70] Zhibo Wang et al. 2022. Threats to training: A survey of poisoning attacks and defenses on machine learning systems. *Comput. Surveys* 55, 7 (2022), 1–36.

[71] Zhepeng Wang et al. 2023. Qumos: A framework for preserving security of quantum machine learning model. In *2023 IEEE International Conference on Quantum Computing and Engineering (QCE)*, Vol. 1. IEEE, 1089–1097.

[72] Zhepeng Wang et al. 2024. PristiQ: A Co-Design Framework for Preserving Data Security of Quantum Learning in the Cloud. *arXiv preprint arXiv:2404.13475* (2024).

[73] Maxwell T West et al. 2023. Towards quantum enhanced adversarial robustness in machine learning. *Nature Machine Intelligence* 5, 6 (2023), 581–589.

[74] Marco Wiedmann et al. 2023. An empirical comparison of optimizers for quantum machine learning with spsa-based gradients. In *2023 IEEE International Conference on Quantum Computing and Engineering (QCE)*, Vol. 1. IEEE, 450–456.
[75] William K Wootters and Wojciech H Zurek. 1982. A single quantum cannot be cloned. *Nature* 299, 5886 (1982), 802–803.
[76] Jonathan Wurtz et al. 2023. Aquila: QuEra's 256-qubit neutral-atom quantum computer. *arXiv preprint arXiv:2306.11727* (2023).
[77] Xanadu. 2024. X-series. https://www.xanadu.ai/products/x-series/ Accessed: 08/15/2024.
[78] Chuanqi Xu et al. 2023. Classification of quantum computer fault injection attacks. *arXiv preprint arXiv:2309.05478* (2023).
[79] Chuanqi Xu et al. 2023. Exploration of power side-channel vulnerabilities in quantum computer controllers. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*. 579–593.
[80] Chuanqi Xu et al. 2024. A Thorough Study of State Leakage Mitigation in Quantum Computing with One-Time Pad. In *2024 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*. IEEE, 55–65.
[81] Mengchen Zhao, Bo An, Wei Gao, and Teng Zhang. 2017. Efficient label contamination attacks against black-box learning models.. In *IJCAI*. 3945–3951.