scientific reports

Check for updates

OPEN

Carbon efficient quantum AI: an empirical study of ansätz design trade-offs in QNN and QLSTM models

Sarvapriya Tripathi[1]✉, Himanshu Upadhyay[1] & Jayesh Soni[2]

The rising environmental cost of deep learning has placed Green AI, which promotes focus on reducing the carbon footprint of AI, at the forefront of sustainable computing. In this study, we investigate Quantum Machine Learning (QML) as a novel and energy-efficient alternative by benchmarking two quantum models, the Quantum Neural Network (QNN) and Quantum Long Short-Term Memory (QLSTM), on the N-BaIoT anomaly detection dataset. Our first phase of experiments compares the QNN and QLSTM models using ten distinct quantum circuit designs (ansätze A1–A10). We systematically compare trade-offs between classification performance, model complexity, training time, and energy consumption. The results indicate that simpler QNN ansätze can achieve accuracy comparable to more complex ones while consuming significantly less energy and converging faster. In particular, QNN with ansatz A4 provided the optimal balance between performance and energy efficiency, consistently outperforming QLSTM across most metrics. A detailed energy breakdown confirmed GPU usage as the dominant source of power consumption, underscoring the importance of circuit-efficient quantum design. To contextualize QML's viability, we conducted a second phase of experiments comparing quantum models with three benchmark classical machine learning models: Artificial Neural Network (ANN), Long Short-Term Memory (LSTM), and CatBoost. We find that the classical models demonstrated faster training times and lower energy consumption, highlighting and contrasting the maturity of algorithmic development that classical ML algorithms have already seen. Finally, we examined the energy implications of developing quantum models on actual quantum hardware. This third phase of experiments compared training on IBM Qiskit's emulation environment (running on GPU servers) versus execution on real IBM Quantum hardware. Highlighting the significant differences in execution time and energy footprint, extrapolated results indicate that quantum hardware still incurs higher energy costs. This suggests that further hardware-aware ansätz optimization and improvements in quantum infrastructure are essential to realizing carbon-efficient QML at scale.

Keywords Ansätz, Energy-efficient AI, Quantum machine learning, Sustainability

Abbreviation

AI	Artificial Intelligence
CPU	Central Processing Unit
GPU	Graphics Processing Unit
GRU	Gated Recurrent Unit
KL	Kullback-Leibler
LSTM	Long Short-Term Memory
QLSTM	Quantum Long Short-Term Memory
QNN	Quantum Neural Network
RAM	Random Access Memory
RNN	Recurrent Neural Network
ROC AUC	Receiver Operating Characteristic - Area Under the Curve
SD	Standard Deviation

[1]Department of Electrical and Computer Engineering, Florida International University, Miami, FL, USA. [2]Applied Research Center, Florida International University, Miami, FL, USA. ✉email: strip011@fiu.edu

TDP	Thermal Design Power
T4 GPU	NVIDIA Tesla T4 Graphics Processing Unit

Quantum computing (QC) has rapidly evolved as a computational paradigm that offers theoretical advantages over classical computing, particularly in terms of complexity handling[1,2]. Unlike classical systems, quantum computers leverage principles such as superposition and entanglement to perform operations on high-dimensional data spaces with exponential scalability. There has, subsequently, been the emergence of Quantum Machine Learning (QML), which combines quantum computation with machine learning algorithms to enhance performance on data-intensive tasks[3,4]. Among QML approaches, variational quantum models such as Quantum Neural Network (QNN) and Quantum Long Short-Term Memory (QLSTM) network have gained prominence due to their adaptability and suitability for current noisy intermediate-scale quantum (NISQ) devices[5,6]. These models rely on Parameterized Quantum Circuits (PQCs), or ansätze, whose learning capacity and optimization behavior are directly defined by their topology including circuit depth and entanglements.

As quantum computing progresses toward practical deployment, integrating Green AI principles becomes critical to ensure a sustainable energy footprint[7–9]. Most of the current Quantum technology presents very high energy requirements, especially the superconducting systems that require extreme cooling. This contributes significantly to the environmental footprint for quantum computing[10–12]. Green AI in machine learning emphasizes the need for energy-efficient hardware innovations[13,14]. In addition to energy concerns inherent to quantum hardware, a significant portion of QML research and experimentation today occurs within classical emulation environments using CPUs and GPUs. These simulations, while necessary due to the limited accessibility of quantum devices, are computationally intensive and can lead to substantial energy consumption[15]. Therefore, incorporating Green AI principles into the algorithmic design in QML is crucial along with algorithmic improvements to reduce the quantum system's runtime and energy footprint. This will include optimizing ansätze designs and reducing parameter counts to lower the resource burden during model development in a simulation environment.

On the applied front, there is a growing interest in using QML to real-world domains such as cybersecurity[16–19]. Attacks from Botnets such as Mirai and BASHLITE pose serious security threats[20,21]. The N-BaIoT dataset, which comprises of real network traffic from compromised IoT devices, serves as a widely accepted benchmark for such intrusion detection tasks[22]. QML approaches, including QNNs and QLSTM networks, may offer novel advantages in accurate identification of botnet traffic.

Quantum Artificial Intelligence (QAI) offers transformative potential over classical AI by leveraging quantum phenomena such as superposition, entanglement, and quantum interference to achieve computational and representational advantages in real-world applications[23]. Recent studies have demonstrated that QAI models can outperform classical algorithms in complex domains including quantum sensing, computer vision, and object identification. For instance, Quantum Reinforcement Learning (QRL) has been shown to achieve high-accuracy state preparation in quantum many-body systems, reaching the Heisenberg limit even in noisy environments, surpassing the limitations of adiabatic methods[24]. Similarly, quantum-enhanced generative models utilizing programmable quantum circuits have exhibited superior reconstruction and generalization capabilities compared to classical deep learning frameworks, particularly in tasks such as image inpainting and colorization. In applied imaging problems, hybrid QAI approaches have achieved up to a 10% improvement in identification accuracy over classical machine learning methods, demonstrating the feasibility of quantum advantage[25]. These findings collectively suggest that QAI may provide efficiency gains and faster convergence to position it as a promising technology for intelligent and sustainable computational systems.

In this work, we unite the perspectives of QML algorithms, ansätz designs, and energy-efficient AI in the context of IoT botnet detection. We implement QNN and QLSTM models for classifying attacks in the N-BaIoT dataset while exploring ten different ansätz circuit architectures ranging from very simple to highly complex. We monitor each model's performance and measure the energy and carbon emissions of the training process using CodeCarbon, which is a tool to quantify CO_2 emissions from computing resources[26]. By analyzing accuracy, stability, training time, and carbon footprint, we provide a holistic evaluation of quantum-powered models on a real cybersecurity task.

Research motivation

The growing environmental footprint of Artificial Intelligence has prompted a shift in research priorities, bringing Green AI into focus as a critical need. The Green AI advocates for energy-aware model development by challenging the trend of increasingly large models that offer marginal performance gains at disproportionate energy costs. With some large-scale models consuming megawatt-hours of electricity, the carbon implications of AI are not hypothetical anymore. As the adoption of AI expands, so does its contribution to climate change thus making energy efficiency a scientific and ethical imperative.

While the environmental footprint of quantum computing hardware has begun to attract scholarly attention, with studies exploring energy consumption, cooling requirements, and lifecycle emissions of various quantum computing platforms, there remains a significant research gap in understanding how quantum algorithm design itself can contribute to sustainability. Existing work has predominantly focused on the physical infrastructure and system-level efficiencies, overlooking the role that algorithmic strategies might play in reducing computational burden, training time, and energy expenditure. This presents an untapped opportunity to develop algorithm-aware Green AI methodologies within QML, where circuit architecture, depth, and parameterization can be optimized not only for accuracy but also for environmental efficiency. By systematically investigating how different ansätze influence both performance and energy usage, researchers can begin to formulate principles for carbon-aware quantum algorithm design, thereby aligning QML advancements with broader goals of sustainable

computation. This paper seeks to address this gap by empirically profiling the energy consumption and carbon emissions of QML models, contributing critical insights into their potential role in sustainable AI development.

Our study focuses on a representative and high-impact use case in cybersecurity, that of botnet detection. We use N-BaIoT, an open-source botnet traffic reference dataset, and systematically examine how the design of quantum circuits (ansätze), the choice of model architecture (QNN vs. QLSTM), and training efficiency affect sustainability metrics. This dual emphasis on accuracy and environmental cost is central to the principles of Green AI. By treating energy and carbon efficiency as first-class metrics, we aim to guide future QML research toward Pareto-optimal configurations that balance sustainability with functional effectiveness.

Key contribution

This paper makes the following contributions :

1. Integration of QML and Green AI: We introduce a novel study combining sustainability metrics with quantum machine learning. While implementing and comparing both QNN and QLSTM architectures for IoT intrusion detection, we investigate not only the prediction accuracy but also the energy and carbon costs during the training process.
2. Extensive Ansätz Analysis: We evaluated 10 different parameterized quantum circuit designs for each of the two models. These ansätze follow different hardware efficient templates involving layered $R_X/R_Y/R_Z$ rotations with various entanglements. This systematic investigation clarifies how circuit depth and different types of gates affect both classification performance as well as energy consumption.
3. Energy and Emissions Measurement: Using the CodeCarbon toolkit we measure the power drawn by CPU, GPU, and RAM during the training and calculate total energy (kWh) and CO2 equivalent emissions (Kg CO_{2e}). We adopt the rigorous protocol as that of Green AI work which includes consistent hardware, 15-s sampling of GPU through NVIDIA's management library, and CPU through RAPL and regional carbon intensity factors.
4. Insights for Sustainable QML Design: We study the sustainable QML design in the context of the identification of Pareto-optimal ansätz configurations that jointly optimize classification performance and environmental efficiency. We investigate if there is a relationship between ansätz complexity and their achieving maximum accuracy while minimizing energy consumption and training time. This approach to investigation enables the development of carbon-aware quantum AI systems that are both effective and environmentally responsible.

Structure

The rest of the paper is organized as follows - The Literature Review section contains the survey of the related work on Green AI, Quantum Machine Learning, and IoT intrusion detection. Methods section details the N-BaIoT dataset, the QNN and QLSTM architectures, ten quantum ansätze designs, and the CodeCarbon Measurement setup. The Experimental Setup section explains the underlying experimental configurations, including data splitting, training parameters, and hardware environment. The Results section discusses the results in deep detail, and we compare model accuracy across ansätze, analyzing their training results and examining carbon/energy metrics. And finally, Discussion section concludes with the key findings, model performance and environmental trade-offs, strengths, limitations, and an outline of the future directions for energy-efficient quantum AI research.

Literature review

QML and Green AI have each gained rapid momentum in recent years converging around a common goal which is to build intelligent systems that are not only powerful but also environmentally sustainable. Initially introduced by Schwartz et al.[7], Green AI calls for a conscious shift in machine learning from purely performance-driven development, to approaches that measure energy consumption, training time, and carbon impact as core evaluation criteria. This vision has catalyzed the emergence of tools such as CodeCarbon[26], which enables AI researchers to log real-time energy consumption and carbon emissions of CPU, GPU, and memory during training.

At the same time, QML proposes a new paradigm for learning systems. Quantum Neural Networks (QNNs), for example, leverage quantum states to encode data and exploit entanglement and superposition to carry out complex computations with potentially fewer parameters than their classical counterparts. Massoli et al.[27] provided a comprehensive review of QNN architectures which ranges from variational quantum perceptrons to hybrid Boltzmann and convolutional quantum models. Chen et al.[28] further extended this boundary with a formal proposal for QLSTM networks while outlining how quantum circuits can replace conventional gates in LSTM cells to enhance temporal modeling capacity.

As quantum circuits become more and more complex, researchers have begun to investigate the optimal solution of a problem versus the complexity of the design. McLean et al.[29] investigated the problem of barren plateaus, where the gradient vanishes, in relation to deep circuits, which would make highly complex circuits difficult to optimize. Schuld et al.[30] demonstrated that certain shallow parameterized circuits can match the performance of deeper models thus hinting at energy savings through ansätz design. More recently Salek et al.[17] developed a hybrid QML framework to detect adversarial attacks on networks and found that quantum-enhanced classifiers not only improved accuracy but also reduced training iterations compared to the classical baselines.

The domain of cybersecurity has seen early applications of QML, especially for intrusion detection for IoT networks. N-BaIoT dataset[22,31] remains a benchmark for IoT anomaly detection which captured network traffic from devices infected with Mirai and Bashlite botnets. Several classical approaches have already demonstrated

very high accuracy on this dataset using autoencoders, RNNs and CNNs[32]. However, these models are computationally heavy. Kim and Madhavi[16] proposed a hybrid quantum-classical anomaly detector where they used parameterized quantum circuits and outlier analysis on network data where they were able to achieve 99.87% accuracy on a DDoS detection task.

While prior research has increasingly examined the environmental impact of quantum computing[33–38], including factors such as power consumption, cryogenic cooling demands, and platform-specific lifecycle emissions, there has been comparatively little attention paid to the carbon footprint of quantum algorithms themselves. Unlike classical machine learning, where Green AI initiatives have led to tangible advancements in energy-conscious model design, QML has not yet seen a systematic effort to evaluate how algorithmic choices affect sustainability. This study addresses that gap by focusing on the carbon efficiency of variational quantum circuits, specifically examining how different ansätze impact both classification performance and environmental cost. By profiling energy and emissions across a variety of QML configurations using real-world anomaly detection tasks, this work offers a novel, algorithm-level perspective on Green AI in the quantum domain.

A detailed investigation into the operational energy costs of quantum computers was conducted by Parker and Vermeer[39], who attempted to bridge the gap between abstract quantum resource estimates and concrete economic metrics such as electricity usage and cost. While they did not address the upfront capital or maintenance expenses of building a cryptanalytically relevant quantum computer (CRQC), they estimated the marginal cost of electricity required to break a single cryptographic public key. Their model decomposes energy consumption into the product of runtime, number of qubits, and the average power consumed per qubit, a metric that includes all overhead costs such as classical co-processing and cryogenic cooling. Their framing of spacetime volume $\times$ power-per-qubit as a fundamental expression for quantum energy modeling is particularly influential, offering a standardized way to compare quantum architectures and project resource scaling.

Although precise data on the operational electricity consumption and running costs of IBM quantum computers are not publicly published, several researchers have attempted rough estimates. For instance, Desdentado et al.[11] report an assumed power draw of approximately 25 kW for IBM's superconducting-qubit systems, which corresponds to roughly 600 kWh of electricity usage per day. This estimate is derived from earlier reports by Boger[40] and Castro[41], as well as industry-analyst coverage of quantum-computer power consumption in sources such as IEEE Spectrum. Such estimates underscore that the cooling, control electronics and classical infrastructure supporting IBM's quantum processors may dominate their energy footprint, thereby making system-level power consumption a non-negligible factor in assessing the sustainability of quantum computing deployments.

Methods

Dataset

The N-BaIoT dataset[22] provides labeled network traffic from nine IoT devices e.g. cameras, sockets, and thermostats that were intentionally infected with two prominent botnets, namely Mirai and Bashlite. In total, this public dataset contains over 7 million samples each of which has 115 numerical features giving an outline of a packet-level statistic. The target label is binary comprising of attack and benign categories arising from a variety of DDoS and remote access attacks. For this study, we used the L5 group of features (the 5-second history) for the data obtained from the Danmini doorbell and created a data subset of 150,000 records (50,000 records each for benign, Mirai, and Bashlite packets), as detailed in Table 1. The dataset was further split in a 75:25 train-test ratio. We normalized each feature to zero mean and unit variance. Since our primary focus is on the model and energy comparisons, we did not perform further feature selection or augmentation.

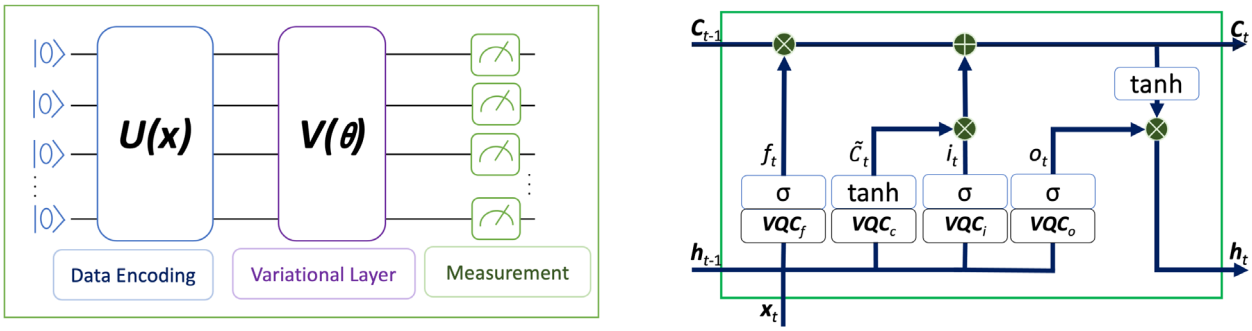
QML models

We evaluate two hybrid quantum architectures:

Quantum Neural Network (QNN) : This is a variational feedforward network[42,43] as illustrated in Fig. 1. Input features are first reduced using a classical dense layer to match the qubit count. We then use amplitude embedding to encode the inputs into the VQCs, which are described in the next section on **Ansätze Designs**. The 10 ansätze resulted in 10 different versions of QNN model which were part of this study. In each ansätz, the final measurement on each qubit is linearly combined to produce the output, which is then used for classification. The algorithm for the QNN training for the 10 ansätze is given in algorithm 1.

Attribute	N-BaIoT
Focus	IoT device behavior
Devices Used	Danmini Doorbell
Features Used	23 (L5 Group)
Attack Types	Mirai and Bashlite
Traffic Type	IoT device traffic
Dataset Size	150,000 records
Train:Test Split	75:25
Labeling	Benign vs. Attack

Table 1. N-BaIoT dataset used in this study.



(a) QNN **(b)** QLSTM

Fig. 1. QML algorithms.

Quantum Long Short-Term Memory (QLSTM) : QLSTM model[28,44,45] mimics the architecture of a classical LSTM, with each gate (input, forget, and output) being constructed from a VQC (Fig. 1). The 10 ansätze used (described in the next section) resulted in 10 versions of QLSTM model that were investigated as part of this research. In each ansätz, the final measurement on each qubit is linearly combined to produce the output, which is then used for classification. The algorithm for the QLSTM training for the 10 ansätze is given in algorithm 2.

Both models use linear activation at the quantum classical interface. Hyperparameters are kept identical except for the optimizer which in QNN's case uses Stochastic Gradient Descent (SGD) while in QLSTM's case uses Adagrad. Other shared hyperparameters are learning rate of 0.01, batch size of 32, 20 epochs and 6 qubits per circuit. We implement both models in PennyLane with a PyTorch backend while enabling automatic differentiation of the quantum circuits.

Require: IoT dataset $\mathscr{D} = \{(x_i, y_i)\}_{i=1}^{N}$, learning rate η,
initial parameters θ, ansätze set $\mathscr{A} = \{A_1, A_2, \ldots, A_{10}\}$
Ensure: Optimized QNN parameters θ^* for each ansatz
for each ansatz $A_j \in \mathscr{A}$ **do**
  Initialize parameters $\theta^{(j)}$ randomly for A_j
  for iteration $t = 1$ to 20 **do**
    for each minibatch (x, y) in $\mathscr{D}$ **do**
      Preprocess input x (e.g., normalization, PCA)
      Encode x into quantum state $|\psi(x)\rangle$ using amplitude encoding: $|\psi\rangle = \sum_{j=1}^{2^n} x_j |j\rangle$
      Apply PQC $U_{A_j}(\theta^{(j)})$ to $|\psi(x)\rangle$
      Measure expectation value $\langle Z \rangle$ on output qubit to obtain prediction $\hat{y}$
      Compute L1 loss: $\mathscr{L}(\hat{y}, y) = |\hat{y} - y|$
      Estimate gradient $\nabla_{\theta^{(j)}} \mathscr{L}$
      Update parameters: $\theta^{(j)} \leftarrow \theta^{(j)} - \eta \nabla_{\theta^{(j)}} \mathscr{L}$
    end for
  end for
end for
return Trained parameters $\{\theta^{(j)*}\}_{j=1}^{10}$

Algorithm 1. QNN training across multiple ansätze

Require: IoT dataset $\mathscr{D} = \{(X_i, y_i)\}_{i=1}^{N}$, learning rate η,
small constant ε, ansätze set $\mathscr{A} = \{A_1, A_2, \ldots, A_{10}\}$
Ensure: Optimized QLSTM parameters $\{\theta^{(j)*}\}_{j=1}^{10}$
for each ansatz $A_j \in \mathscr{A}$ **do**
  Initialize QLSTM parameters $\theta^{(j)}$ randomly
  Initialize accumulator $G^{(j)} \leftarrow 0$ (same shape as $\theta^{(j)}$)
  for iteration $t = 1$ to 20 **do**
    for each sequence-label pair (X, y) in $\mathscr{D}$ **do**
      Preprocess sequence $X = [x^{(1)}, \ldots, x^{(T)}]$ and normalize
      for each time step $t = 1$ to T **do**
        Encode $x^{(t)}$ into quantum state $\left|\psi^{(t)}\right\rangle$ using amplitude encoding: $\left|\psi^{(t)}\right\rangle = \sum_{j=1}^{2^n} x_j^{(t)} |j\rangle$
        Apply QLSTM cell $U_{A_j}(\theta^{(j)})$ to $\left|\psi^{(t)}\right\rangle$ and hidden state $h^{(t-1)}$
        Update hidden state $h^{(t)}$ and cell state $c^{(t)}$
      end for
      Use final $h^{(T)}$ to compute prediction $\hat{y}$
      Compute MSE loss: $\mathscr{L}(\hat{y}, y) = (\hat{y} - y)^2$
      Estimate gradient $\mathbf{g}^{(j)} \leftarrow \nabla_{\theta^{(j)}} \mathscr{L}$
      Update accumulator: $G^{(j)} \leftarrow G^{(j)} + (\mathbf{g}^{(j)})^2$
      Update parameters with Adagrad:
      $\theta^{(j)} \leftarrow \theta^{(j)} - \eta \cdot \frac{\mathbf{g}^{(j)}}{\sqrt{G^{(j)}} + \varepsilon}$
    end for
  end for
end for
return Trained parameters $\{\theta^{(j)*}\}_{j=1}^{10}$

Algorithm 2. QLSTM training across multiple ansätze

Ansätze designs

We designed 10 distinct ansätze as the core of our experimental setup as shown in Fig. 2. Each ansätz varies in depth, rotation gates and entanglement patterns:

A1: This hardware-efficient linear entangling circuit is the simplest of all designs. It encodes the input features into each qubit's amplitudes and applies a parameterized $R_X(\theta_x)$ rotation on every qubit. This is followed by CNOT gates which are applied sequentially between adjacent qubits to entangle them in a chain. The result is a BasicEntangler pattern ensuring nearest neighbor connectivity across the register.

A2: This ansätz builds on Ansätz 1 by adding R_Z rotations to each qubit. Each qubit is acted on by both the $R_X(\theta_x)$ and $R_Z(\theta_x)$ rotations. The qubits are then entangled through the same chain of sequential CNOT gates as that of Ansätz 1. The additional R_Z gates provide extra tunable parameters, which allows for finer adjustment of the model's weights while training.

A3: This ansätz uses two layers of qubit rotations, where every qubit is rotated by $R_X(\theta_x)$, $R_Y(\theta_y)$ and $R_Z(\theta_z)$ gates. In the first layer, CNOT gates are applied sequentially between the neighboring qubits, while in the second layer, the CNOTs connect next-to-nearest neighbors each time skipping one qubit. This two-layer scheme provides richer expressivity by introducing extra degrees of freedom in the parameters.

A4: This ansätz alternates between single qubit $R_Y(\theta_y)$ rotations and global CZ entangling gates in a staggering pattern. Each layer applies $R_Y(\theta_y)$ rotations to all qubits followed by CZ gates that entangle qubits across the register. This design generates correlations over multiple qubits at once and results in an ansätz with a fixed depth.

A5: This ansätz also uses layered rotations and entanglement but with different sequence. It begins with $R_X(\theta_x)$, $R_Y(\theta_y)$ and $R_Z(\theta_z)$ rotations, followed by two successive layers of entanglement. The first layer of entanglements is achieved via a linear chain of CNOT gates between adjacent qubits, while the second layer incorporates Controlled-RX gates also applied in a linear pattern.

A6: It is the most complex ansätz in the group and uses dense entanglement across multiple layers. It begins with $R_X(\theta_x)$, $R_Y(\theta_y)$ and $R_Z(\theta_z)$ rotations on each qubit followed by a fully connected layer of Controlled-RX gates. In this dense layer, each qubit is entangled with every other qubit through CRX operations while creating maximum entanglement across the register.

A7: This ansätz has a fixed shallow depth. Each qubit is acted upon by $R_X(\theta_x)$ and $R_Z(\theta_z)$ rotations. Moreover, the entanglement is introduced with Controlled-RX gates connecting neighboring qubits. Because of its choice of rotation gates and a single entangling operation, this circuit is one of the simplest designs with a shallow depth.

A8: This ansätz also has a fixed shallow depth, but uses a different rotation scheme. Each qubit receives $R_Y(\theta_y)$ and $R_Z(\theta_z)$ rotations while introducing entanglement through a linear chain of CNOT gates. By using $R_Y(\theta_y)$ rotations instead of the R_X, and standard CNOT entangler, this ansätz offers a contrasting structure as compared to the previous ansätz while maintaining the same shallow depth.

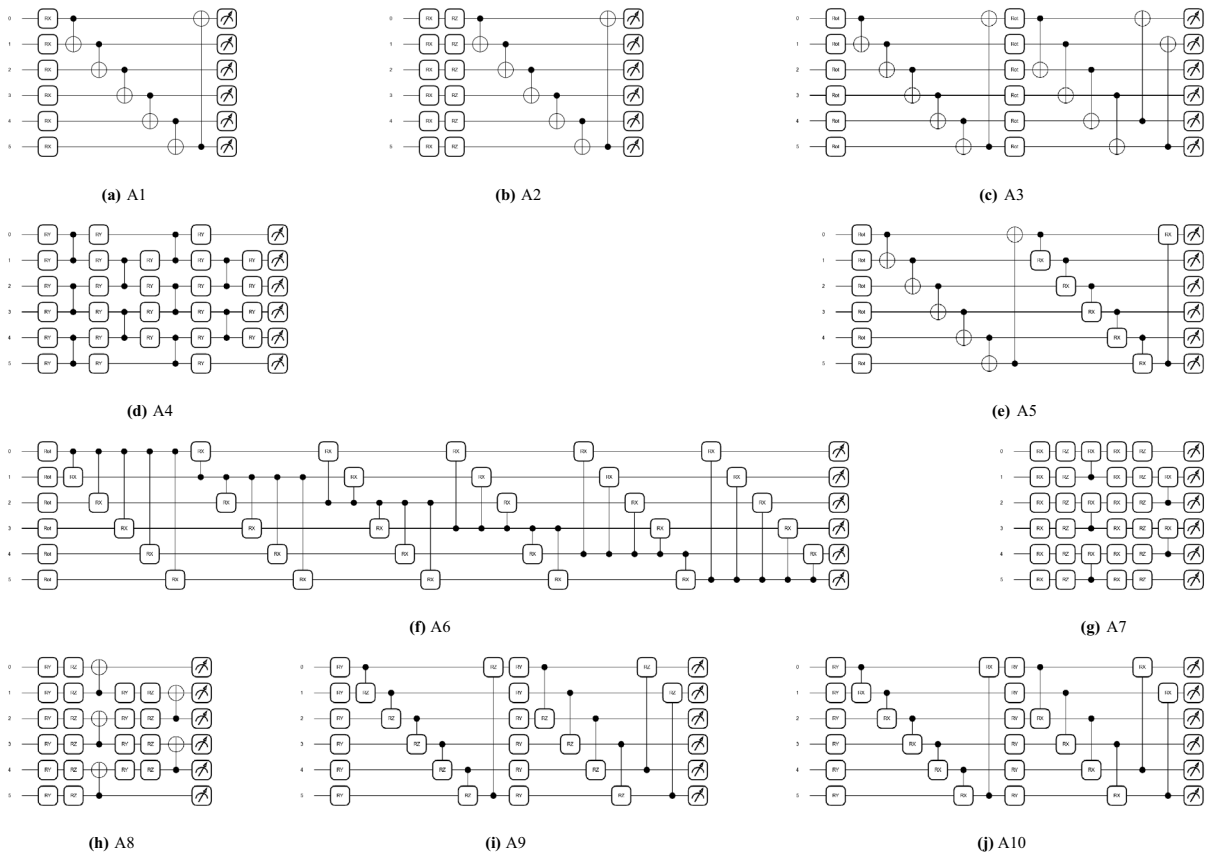


Fig. 2. Schematic representation of the 10 ansätze with varying depths, rotations, and entanglements used in this study.

Ansätz	Rotation gates	Qubit entanglement pattern	Parms	1Q	2Q	Depth
1	R_X	CNOT, Fully connected	n	$2n$	n	$n+2$
2	$R_X + R_Z$	CNOT, Fully connected	$2n$	$3n$	n	$n+3$
3	$R_X + R_Y + R_Z$ layered	CNOT, Fully connected, layered	$3n$	$7n$	$2n$	$2n+7$
4	R_Y layered	CZ on alternate qubits, layered	n	$6n-4$	$2n-2$	10
5	$R_X + R_Y + R_Z$, R_X	CNOT and CRX, Fully connected	$3n$	$4n$	$2n$	$2n+4$
6	$R_X + R_Y + R_Z$	CRX, Fully connected, layered	$3n$	$4n$	n^2	n^2+4
7	$R_X + R_Z$ pairs	CRX, asymmetric layered	$2n$	$5n$	$n-1$	7
8	$R_Y + R_Z$, asymmetric layered	CNOT, asymmetric layered	$2n$	$5n-4$	$n-1$	7
9	R_Y	CRZ, Fully connected, asymmetric layered	$2n$	$3n$	$2n$	$2n+3$
10	R_Y	CRX, Fully connected, asymmetric layered	$2n$	$3n$	$2n$	$2n+3$

Table 2. Comparison of Ansätz structures by parameter count, gate count, and depth. Here n is the number of qubits in the circuit. Column 1Q shows the number of single-qubit gates, column 2Q shows the number of two-qubit gates, and Parms indicates the total number of trainable parameters.

A9: This ansätz applies $R_Y(\theta_y)$ rotations on each qubit and uses Controlled-RZ gates to entangle them. This ansätz differs from Ansätz 10 only in its choice of entangler gates.

A10: This ansätz is structurally identical to the previous ansätz, except the entanglement is done with Controlled-RX gates. This variation yields a design that performs differently from the previous ansätz.

Carbon measurement tools

To quantify energy usage and emissions we use the CodeCarbon library following the Green AI methodology. CodeCarbon is an open-source Python package that tracks real-time energy consumption of hardware components and estimates CO2 emissions based on the grid carbon intensity. We configure CodeCarbon to log

Metric	Description	Calculation method
Total Energy (kWh)	Total energy consumed by CPU, GPU, and RAM	Total Energy = CPU Energy + GPU Energy + RAM Energy
Total carbon Emissions (kg)	Emissions as carbon-equivalents [carboneq]	Emissions = $C \times E$, where C is the carbon intensity and E is the energy consumed.
CPU Energy (kWh)	Energy used by the CPU	Tracked by Intel Power Gadget (Windows/Mac Intel), powermetrics (Apple Silicon), or Intel RAPL files (Linux). Defaults to TDP-based approximation if unavailable. This experiment was done using Linux, so CPU Energy was tracked using Intel RAPL files.
GPU Energy (kWh)	Energy used by GPU	Tracked using Nvidia GPUs via the `pynvml` library
RAM Energy (kWh)	Energy used by RAM	Estimated with a fixed ratio of 3 W per 8 GB of RAM
Emissions Rate (kg/s)	Emissions divided by duration	Emissions Rate = Total Emissions / Duration(seconds)

Table 3. Metrics and calculation methods used by CodeCarbon.

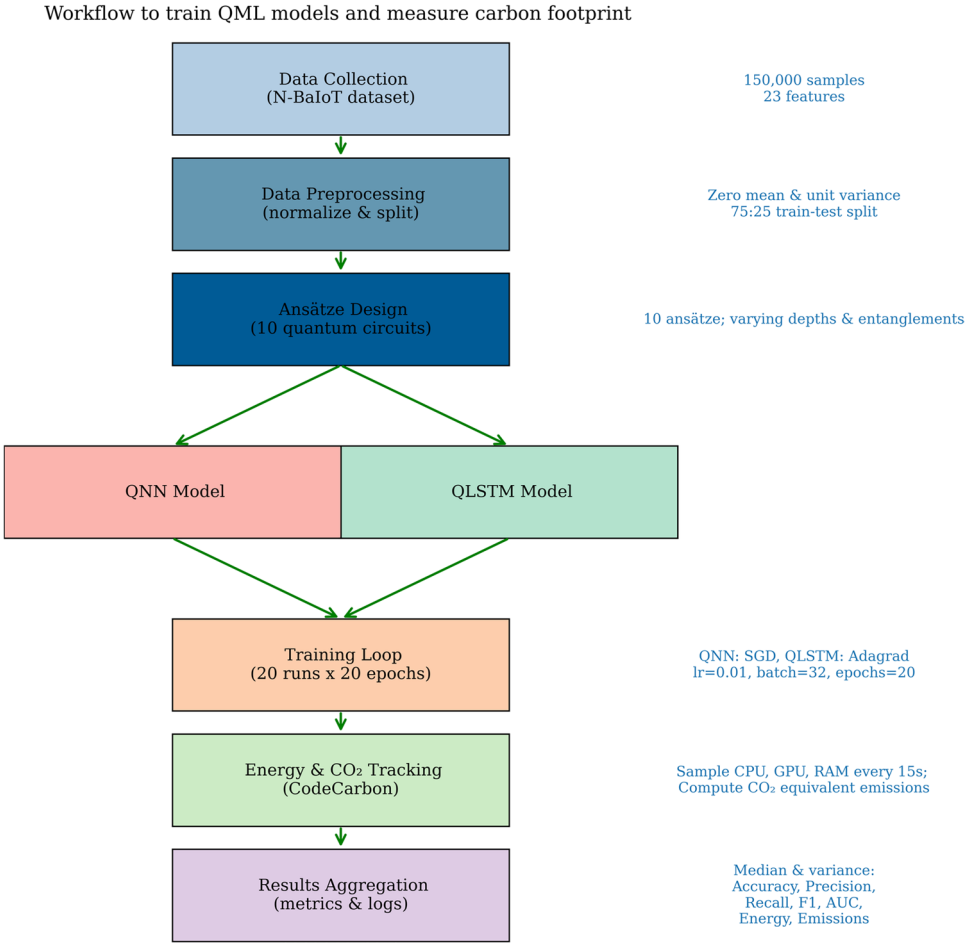


Fig. 3. Green AI framework for evaluating QNN and QLSTM models using 10 distinct ansätze.

CPU, GPU, and RAM power every 15 seconds throughout training. GPU power is read through the NVIDIA management library, CPU through intel RAPL counters, and RAM is approximated by the standard rule of thumb which is almost 3W/8GB (Table 2).

We verify that CodeCarbon's geographic settings match our lab's location. If location data are unavailable it defaults to 475g CO_2/kWh. For each training run, CodeCarbon outputs Total Energy (kWh) which is calculated by integrating power over time, Total Emissions (kg CO_{2e}) which is computed as Energy multiplied by the carbon intensity and Emissions Rate (kg CO_{2e}/sec). By comparing these metrics across models and ansätze we isolate differences in carbon cost corresponding to model efficiency. This allows us to report, for instance the percentage energy reduction of a simpler ansätz versus a complex one or QNN versus QLSTM under the same workload.

The metric definitions and details have been documented in Table 3. These metrics together provide a detailed overview of our models' environmental impact, allowing for comparisons of model efficiency and guiding energy-efficient development practices.

Parameters	QLSTM	QNN
Data Encoding	Amplitude	Amplitude
Qubits	$\lceil \log_2(n_f) \rceil$	$\lceil \log_2(n_f) \rceil$
Learning Rate	0.01	0.01
Batch Size	32	32
Epochs	20	20
Optimizer	Adam	Adam
Loss Function	MSE	MSE
Activation	linear	linear

Table 4. Hyperparameters used for training QNN and QLSTM using PennyLane environment.

Parameters	ANN	LSTM	CatBoost
Learning Rate	0.01	0.01	0.01
Batch Size	32	32	32
Epochs	20	20	20
Optimizer	Adam	Adam	CatBoost
Loss Function	binary_crossentropy	binary_crossentropy	Logloss(binary_crossentropy)
Activation	Sigmoid	Sigmoid	Probability(Sigmoid)

Table 5. Hyperparameters used for training ANN, LSTM, and CatBoost classical ML algorithms.

Parameters	QNN
Data Encoding	Amplitude
Qubits	$\lceil \log_2(n_f) \rceil$
Learning Rate	0.01
Batch Size	50
Optimizer	Adam
Loss Function	MSE
Activation	linear

Table 6. Hyperparameters used for training the QNN model for IBM Qiskit emulation and IBM Quantum Hardware environments.

QML training workflow

The Fig. 3 illustrates the workflow used for evaluating QML models using 10 distinct ansätze (Fig. 2). The pipeline begins with the N-BaIoT dataset, preprocessed through normalization and train-test splitting. The diverse set of ansätze forms the basis for experimental comparison across both QNN and QLSTM architectures. Each model-ansatz combination is trained across 20 independent runs, with 20 epochs per run.

Experimental setup

Training and testing configurations

We split N-BaIoT into a fixed 75% training set and 25% test set while ensuring that the benign and attack ratio is preserved. All models were evaluated on this split. Each model-ansätz combination was trained 20 times, each time starting from different random weights for proper initialization. Each training run consisted of 20 epochs passing through the full data. Keeping the batch size as 32 we monitored training and validation loss for each epoch.

During each run, the CodeCarbon tracker was activated at the beginning of training and stopped at the end of the final epoch. This ensured that we measured only the forward and backward training computations. After training we evaluated model performance on the held-out test data, reporting accuracy, precision, recall, F1-score, and ROC AUC. We aggregated results over the 20 runs to compute the median and variance of each metric for each model-ansätz pair.

Hyperparameters, experimental setup and design

The experiments were conducted in three sequential phases to comprehensively evaluate the energy efficiency and performance of quantum versus classical models. The models were fine-tuned with hyperparameters as shown in Tables 4, 5, and 6. The selection of hyperparameters for all models was informed by both empirical

Component	Specification
Operating System	Linux 6.1.0-25-amd64-x86_64 with glibc 2.36
CPU	Intel(R) Xeon(R) Gold 5218 CPU @ 2.30GHz (12 physical cores)
RAM	128 GB
GPU	NVIDIA Tesla T4 with 16 GB of GDDR6 memory
NVIDIA Driver Version	560.35.03
CUDA Version	12.6

Table 7. System specifications for experimental setup for phase 1 and 2.

Component	Specification
Operating System	Linux 6.1.0-25-amd64-x86_64 with glibc 2.36
CPU	Intel(R) Xeon(R) Gold 5218 CPU @ 2.30GHz (12 physical cores)
RAM	128 GB
GPU	NVIDIA A100 (80 GB) and A100 (40 GB)
NVIDIA Driver Version	570.133.20
CUDA Version	12.8

Table 8. System specifications for experimental setup for phase 3.

Parameter	Value
Number of Qubits	127
Best Two-Qubit Gate Error	2.74×10^{-3}
Layered Two-Qubit Gate Error	1.87×10^{-2}
CLOPS (Circuit Layer Operations per Second)	180 K

Table 9. Specifications of IBM `ibm_brisbane` Quantum Computer.

experimentation and established practices reported in the literature. Preliminary exploratory runs were conducted to identify stable parameter ranges that ensured convergence while avoiding overfitting. Drawing from prior studies in quantum and classical machine learning, key parameters such as learning rate, batch size, optimizer choice, and number of epochs were fine-tuned to balance training efficiency with model generalization. Once finalized, these hyperparameters were applied uniformly across all experimental runs to ensure methodological consistency and enable direct comparability between architectures and ansätze.

In the first phase, ten distinct quantum circuit architectures (ansätze A1–A10) were evaluated under two model types (QNN and QLSTM) using the Pennylane framework in simulation mode. These experiments were executed in the computational environment described in Table 7, where both classification accuracy and energy usage were monitored over multiple trials.

For the QML models, amplitude encoding was employed, with circuit structures designed to accommodate the dimensionality of the input features. We set the learning rate to 0.01, batch size to 32, and number of epochs to 20 . The optimizer for both of the models, QNN and QLSTM, was Adagrad. We used linear activation on the classical output of each model. Except for the quantum circuit itself, all other components were identical between runs. We used PennyLane's "default.qubit" simulator for all quantum operations. This setup ensured that our energy measurements would reflect classical simulation of quantum circuits. In practice, true quantum hardware might have a different energy profiling. However, the simulator approach allowed us to make consistent comparisons, and to ensure that the relative differences between the ansätze would still hold conceptually.

All experiments were executed on the same hardware to ensure fairness. We used a server equipped with an NVIDIA Tesla T4 GPU and an Intel Xeon Gold 5218 CPU, with 128 GB of RAM. The software environment consisted of Python 3.11 with PennyLane and PyTorch. The Tesla T4's power and performance characteristics were typical to that of modern data center GPUs. By keeping hardware consistent, we ensured that any differences in energy use were due to model-ansätz choices and not due to device changes. All CodeCarbon logs automatically detected this GPU and CPU setup while attributing power correctly. The server configuration is detailed in Table 7.

In the second phase, benchmark *classical models* including Artificial Neural Network (ANN), Long Short-Term Memory (LSTM), and CatBoost were trained and evaluated on the same dataset to enable a comparative baseline against quantum models. These experiments were also conducted under the same hardware and software environment to ensure consistency. The hyperparameters for the three classical models are given in Table 5.

In the third phase, experiments were designed to compare energy consumption between a quantum model operating in an emulated setting and on actual quantum hardware. For this phase, IBM Qiskit was selected as the

framework, and the experiment was confined to QNN model, utilizing ansätz A1. Emulation-based experiments were performed using high-performance GPU servers equipped with NVIDIA A100 units, as detailed in Table 8. For the quantum hardware runs, IBM's `ibm_brisbane` backend was used (Table 9). Due to access limitations on IBM Quantum systems, full dataset training was not feasible within the allocated runtime. As a result, the dataset was partitioned into smaller subsets of 50 samples each, allowing for successful execution and timing of training runs. The model used hyperparameters shown in Table 6.

The aggregate runtime from these subsets was then extrapolated to estimate the total time required for full dataset training. This extrapolated value, in conjunction with literature-based estimates of IBM Quantum's average power usage (25 kW), was used to compute the expected energy consumption of a full training run on quantum hardware. This is discussed in more details in next section.

Energy consumption estimation on IBM quantum and emulated platforms

To estimate the energy consumption associated with training quantum models on IBM Quantum hardware, a two-pronged experimental approach was adopted. First, a Qiskit-equivalent implementation of the QNN model using Ansatz A1 was developed and executed in an emulation environment. The emulation was performed using Qiskit version 1.4.3 on high-performance computing servers equipped with NVIDIA A100 GPUs. The QNN model was trained for 20 epochs and repeated across 20 trials to ensure statistical robustness. During each trial, real-time energy consumption was monitored and recorded using the CodeCarbon library, enabling the collection of detailed power usage metrics for subsequent analysis.

When deploying the same model on IBM Quantum hardware, it was observed that a full-dataset training run could not be completed within the allocated quantum runtime quota due to extended execution times. To address this limitation, the dataset was partitioned into smaller subsets of 50 records each, enabling multiple complete training runs within the available resource window. Training runs were conducted on several such subsets, and the corresponding total training time was recorded. This average run-time was then extrapolated to estimate the total time required to train the model on the full dataset. Using publicly reported estimates of IBM Quantum's energy consumption, approximately 25 kW (or 25 kWh per hour), the extrapolated training time was multiplied by the power consumption rate to derive an estimated total energy usage for a full QNN training cycle on IBM Quantum hardware.

Experimental design for classical model energy profiling

To establish a robust comparative baseline for evaluating the energy efficiency of quantum models, a series of experiments were conducted using three widely adopted classical machine learning models: Artificial Neural Networks (ANN), Long Short-Term Memory (LSTM) networks, and CatBoost. Each model was trained on the same preprocessed version of the N-BaIoT dataset used in the quantum experiments, ensuring consistency in data conditions and evaluation. For each model, 20 independent trials were performed using identical hardware configurations. During each trial, real-time energy consumption was monitored using the CodeCarbon tracking library, which logs power usage across CPU and GPU components. The resulting energy values were aggregated to compute median and standard deviation statistics for each classical model's total energy consumption, providing a reliable estimate of their computational footprints under repeated training scenarios.

Results

Phase 1: energy consumption and carbon emission comparison for various ansätze in QNN and QLSTM models

One of the primary aims of this study was to evaluate and compare the energy efficiency and carbon footprint of QML architectures, deployed across 10 different ansätze using PennyLane. The evaluation criteria included Total Energy Consumption, Carbon Emissions, Energy Breakdown by H/W components (CPU, GPU, and RAM), Emission Rate, and Training Time. The results for these statistics are detailed in Table 10 and also depicted in Fig. 4.

Across all ansatz configurations, QNN models consistently demonstrated lower energy consumption and carbon emissions compared to their QLSTM counterparts, owing to the higher complexity of a QLSTM architecture as compared to QNN. For example, for A6 (the most complex ansätz), the median for QNN model was at 0.61 kWh while for QLSTM it was 0.78 kWh. While this difference may appear modest in magnitude, it is amplified when considering large-scale training runs or multiple model iterations common in practical deployments. Additionally, simpler circuits like A1 and A4 in QNN showed the lowest total energy consumption, making them especially attractive for low-carbon AI implementations. On average, QLSTM models consume 25–30% more energy than their QNN counterparts for the same ansatz.This trend holds consistently across CPU, GPU, and RAM energy usage, with the GPU being the dominant contributor in both models.

Similarly, QNN models maintained lower total CO_2 emissions, with configurations such as A4 having emission median at 0.063 kg CO_2 compared to QLSTM's at 0.085 kg under the same ansätz. CO_2 emissions mirror the energy consumption pattern. The consistent trend of reduced carbon footprint in QNNs suggests that the structure of QNNs imposes less computational burden and energy demand than the QLSTM models.

Disaggregating energy use by component provides further insight. In both models, the GPU was the dominant energy consumer, accounting for more than 70% of total energy usage across most configurations. QLSTM mirrored the trend of QNN, albeit at slightly higher absolute levels, indicating the energy-intensive nature of quantum circuit execution on current quantum simulators accelerated by GPUs.

Interestingly, RAM energy consumption remained relatively constant and marginal across all configurations, indicating that RAM usage is not a significant differentiator for environmental performance in these quantum models.

Model/ Metric	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10
QNN - Quantum Neural Network										
Total Energy (kWh)	0.1933 ± 0.0104	0.3140 ± 0.0161	0.3240 ± 0.0128	0.1363 ± 0.0169	0.3042 ± 0.0082	0.6082 ± 0.0151	0.2827 ± 0.0139	0.2227 ± 0.0154	0.3053 ± 0.0150	0.2840 ± 0.0135
CO2 Emissions (kg)	0.0897 ± 0.0048	0.1457 ± 0.0075	0.1504 ± 0.0060	0.0632 ± 0.0078	0.1410 ± 0.0041	0.3124 ± 0.0075	0.1312 ± 0.0064	0.1034 ± 0.0071	0.1417 ± 0.0070	0.1318 ± 0.0063
CPU Energy (kWh)	0.0340 ± 0.0018	0.0549 ± 0.0028	0.0812 ± 0.0032	0.0340 ± 0.0041	0.0849 ± 0.0024	0.1328 ± 0.0031	0.0551 ± 0.0027	0.0432 ± 0.0029	0.0776 ± 0.0043	0.0713 ± 0.0034
GPU Energy (kWh)	0.1390 ± 0.0074	0.2268 ± 0.0116	0.2098 ± 0.0083	0.0885 ± 0.0110	0.1988 ± 0.0053	0.4432 ± 0.0112	0.1948 ± 0.0096	0.1537 ± 0.0107	0.1960 ± 0.0090	0.1837 ± 0.0087
RAM Energy (kWh)	0.0203 ± 0.0011	0.0324 ± 0.0017	0.0329 ± 0.0013	0.0139 ± 0.0017	0.0205 ± 0.0006	0.0324 ± 0.0008	0.0329 ± 0.0016	0.0258 ± 0.0018	0.0315 ± 0.0018	0.0290 ± 0.0014
Emissions Rate (Kg/s)	$6.10 \times 10^{-6} \pm$ 7.58×10^{-8}	$6.10 \times 10^{-6} \pm$ 8.27×10^{-8}	$6.12 \times 10^{-6} \pm$ 1.01×10^{-7}	$6.09 \times 10^{-6} \pm$ 6.79×10^{-8}	$6.05 \times 10^{-6} \pm$ 9.08×10^{-8}	$6.08 \times 10^{-6} \pm$ 5.68×10^{-8}	$5.53 \times 10^{-6} \pm$ 8.63×10^{-8}	$5.54 \times 10^{-6} \pm$ 8.15×10^{-8}	$6.22 \times 10^{-6} \pm$ 1.03×10^{-7}	$6.12 \times 10^{-6} \pm$ 9.90×10^{-8}
Training Time (sec)	1341 ± 10	2153 ± 9	2178 ± 16	961 ± 8	1401 ± 10	2088 ± 22	2189 ± 24	1778 ± 52	2074 ± 46	1970 ± 5
QLSTM - Quantum Long Short-Term Memory										
Total Energy (kWh)	0.2628 ± 0.0177	0.4848 ± 0.0171	0.4856 ± 0.0179	0.1821 ± 0.0174	0.3841 ± 0.0106	0.7799 ± 0.0136	0.4448 ± 0.0146	0.3458 ± 0.0158	0.4748 ± 0.0198	0.4692 ± 0.0126
CO2 Emissions (kg)	0.1221 ± 0.0083	0.2251 ± 0.0080	0.2254 ± 0.0083	0.0845 ± 0.0081	0.1901 ± 0.0052	0.4091 ± 0.0066	0.2065 ± 0.0068	0.1604 ± 0.0074	0.2203 ± 0.0092	0.2178 ± 0.0059
CPU Energy (kWh)	0.0465 ± 0.0032	0.0867 ± 0.0030	0.0964 ± 0.0037	0.0358 ± 0.0034	0.0822 ± 0.0024	0.1286 ± 0.0027	0.0883 ± 0.0029	0.0680 ± 0.0031	0.1192 ± 0.0050	0.1177 ± 0.0032
GPU Energy (kWh)	0.1888 ± 0.0127	0.3469 ± 0.0123	0.3320 ± 0.0121	0.1250 ± 0.0120	0.2616 ± 0.0072	0.5896 ± 0.0096	0.3039 ± 0.0100	0.2372 ± 0.0108	0.3076 ± 0.0128	0.3039 ± 0.0081
RAM Energy (kWh)	0.0275 ± 0.0019	0.0510 ± 0.0018	0.0572 ± 0.0022	0.0213 ± 0.0020	0.0409 ± 0.0012	0.0616 ± 0.0014	0.0525 ± 0.0017	0.0406 ± 0.0019	0.0481 ± 0.0020	0.0475 ± 0.0013
Emissions Rate (Kg/s)	$6.10 \times 10^{-6} \pm$ 6.75×10^{-8}	$6.14 \times 10^{-6} \pm$ 6.52×10^{-8}	$6.45 \times 10^{-6} \pm$ 6.65×10^{-8}	$6.32 \times 10^{-6} \pm$ 7.58×10^{-8}	$6.14 \times 10^{-6} \pm$ 7.78×10^{-8}	$6.10 \times 10^{-6} \pm$ 5.29×10^{-8}	$6.18 \times 10^{-6} \pm$ 7.92×10^{-8}	$6.14 \times 10^{-6} \pm$ 6.97×10^{-8}	$6.32 \times 10^{-6} \pm$ 7.36×10^{-8}	$6.31 \times 10^{-6} \pm$ 6.53×10^{-8}
Training Time (sec)	1425 ± 13	2879 ± 19	3187 ± 40	1118 ± 10	2228 ± 25	3551 ± 37	2932 ± 34	2224 ± 22	2681 ± 37	2657 ± 24

Table 10. Energy consumption and Total Emission comparison between QNN and QLSTM models across 10 ansätz configurations. All numbers are represented in median ± standard deviation notation.

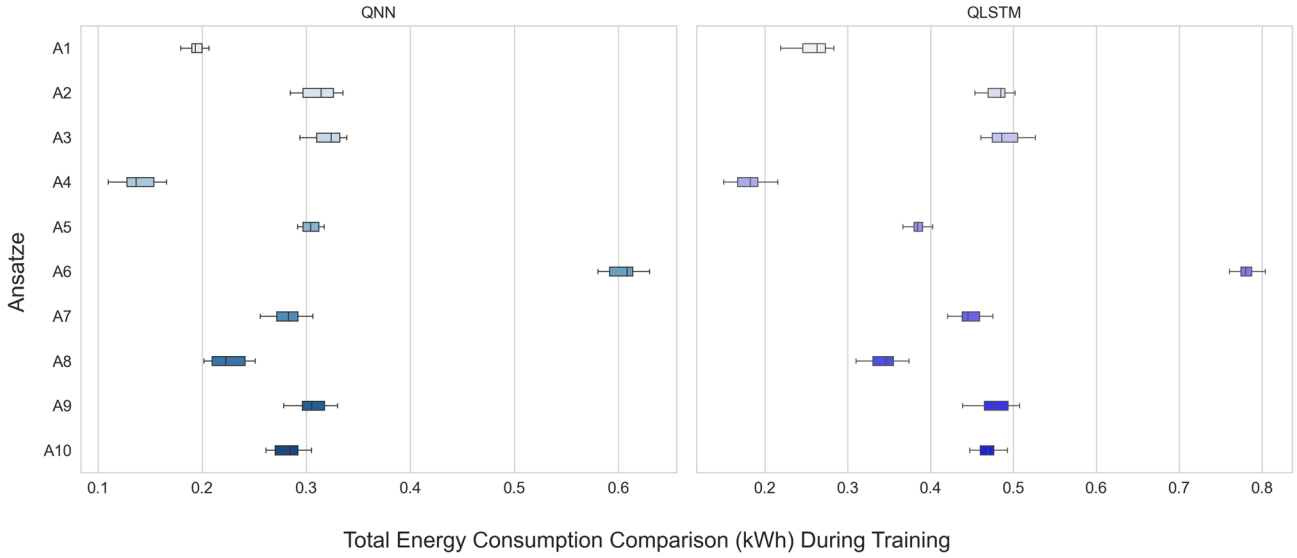


Fig. 4. Energy consumption comparison of QNN and QLSTM for ansatze across runs.

Model/ Metric	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10
QNN - Quantum Neural Network										
Accuracy	0.9343 ± 0.0309	0.9288 ± 0.0332	0.9299 ± 0.0335	0.9536 ± 0.0243	0.9570 ± 0.0332	0.9361 ± 0.0248	0.9456 ± 0.0257	0.9484 ± 0.0262	0.9413 ± 0.0688	0.9602 ± 0.0341
Precision	0.9846 ± 0.0288	0.9902 ± 0.0056	0.9926 ± 0.0064	0.9881 ± 0.0066	0.9856 ± 0.0053	0.9883 ± 0.0060	0.9908 ± 0.0058	0.9899 ± 0.0059	0.9829 ± 0.0403	0.9905 ± 0.0071
Recall	0.9396 ± 0.0226	0.9636 ± 0.0169	0.9469 ± 0.0270	0.9609 ± 0.0254	0.9474 ± 0.0256	0.9483 ± 0.0191	0.9600 ± 0.0331	0.9765 ± 0.0223	0.9442 ± 0.0451	0.9348 ± 0.0432
F1-Score	0.9635 ± 0.0210	0.9704 ± 0.0094	0.9715 ± 0.0140	0.9727 ± 0.0113	0.9688 ± 0.0183	0.9675 ± 0.0118	0.9681 ± 0.0184	0.9806 ± 0.0141	0.9681 ± 0.0467	0.9738 ± 0.0239
AUC ROC	0.9624 ± 0.0311	0.9742 ± 0.0098	0.9669 ± 0.0134	0.9703 ± 0.0146	0.9599 ± 0.0140	0.9722 ± 0.0145	0.9685 ± 0.0167	0.9827 ± 0.0174	0.9658 ± 0.0642	0.9603 ± 0.0208
QLSTM - Quantum Long Short-Term Memory										
Accuracy	0.8952 ± 0.0733	0.9048 ± 0.0625	0.9155 ± 0.0724	0.9644 ± 0.0384	0.8970 ± 0.0768	0.9139 ± 0.0525	0.9319 ± 0.0706	0.9337 ± 0.0440	0.9057 ± 0.0570	0.9261 ± 0.0452
Precision	0.9129 ± 0.0641	0.9352 ± 0.0635	0.9566 ± 0.0557	0.9764 ± 0.0295	0.9067 ± 0.0691	0.9465 ± 0.0444	0.9545 ± 0.0815	0.9421 ± 0.0282	0.9299 ± 0.0623	0.9540 ± 0.0298
Recall	0.9199 ± 0.0435	0.9163 ± 0.0441	0.9434 ± 0.0439	0.9495 ± 0.0235	0.9050 ± 0.0454	0.9295 ± 0.0302	0.9294 ± 0.0412	0.9216 ± 0.0203	0.9414 ± 0.0498	0.9345 ± 0.0296
F1-Score	0.9166 ± 0.0501	0.9306 ± 0.0512	0.9488 ± 0.0490	0.9607 ± 0.0257	0.8907 ± 0.0559	0.9360 ± 0.0313	0.9401 ± 0.0531	0.9362 ± 0.0174	0.9327 ± 0.0510	0.9401 ± 0.0226
AUC ROC	0.8921 ± 0.0826	0.9279 ± 0.0849	0.9369 ± 0.0681	0.9596 ± 0.0306	0.8459 ± 0.1000	0.9121 ± 0.0560	0.8948 ± 0.1542	0.9288 ± 0.0295	0.9093 ± 0.0786	0.9242 ± 0.0381

Table 11. Performance metrics comparison between QNN and QLSTM models across 10 ansätz configurations. All numbers are represented in median ± standard deviation notation.

The emissions rate (total emission in a unit training duration) varied modestly across ansätze but was generally in the same order of magnitude (10^{-6} kg/s). QNN configurations showed more favorable emissions rates compared to their QLSTM equivalents, affirming the potential of QNNs for low-carbon applications (Table 11).

Training time varied significantly across ansätze, with A4 being the fastest in both models. In contrast, some QLSTM configurations such as A3 and A6 required significantly longer training time, owing to the computational overhead associated with quantum recurrent architectures. Generally, A6 configuration was the most computationally expensive, largely due to its complexity. The general trend in the data suggests that the added complexity of QLSTM does not always translate to proportionally improved performance, especially when weighed against energy and time costs.

Phase 2: energy and training time comparison between classical and quantum models

To assess the computational efficiency and environmental impact of quantum versus classical models, we measured both training time and energy consumption across five representative architectures: Classical Models

Category	Model	Training time	Energy consumed
Classical	ANN	$5.84 \times 10^{1} \pm 1.08 \times 10^{1}$	$7.86 \times 10^{-4} \pm 1.45 \times 10^{-4}$
	LSTM	$1.15 \times 10^{2} \pm 2.08 \times 10^{1}$	$1.54 \times 10^{-3} \pm 2.79 \times 10^{-4}$
	Catboost	$8.88 \times 10^{0} \pm 2.63 \times 10^{-1}$	$1.20 \times 10^{-4} \pm 3.53 \times 10^{-6}$
Quantum	QNN	$1.98 \times 10^{3} \pm 4.13 \times 10^{2}$	$2.92 \times 10^{-1} \pm 1.18 \times 10^{-1}$
	QLSTM	$2.66 \times 10^{3} \pm 7.22 \times 10^{2}$	$4.55 \times 10^{-1} \pm 1.55 \times 10^{-1}$

Table 12. Comparison of energy consumed by classical versus quantum models during training runs. All numbers are in kWh and are represented in median ± standard deviation notation.

Metric	A100 (80 GB)	A100 (40 GB)
Duration per Epoch (sec)	3774	8788
Energy Consumed (kWh)	0.2276	1.2639

Table 13. Comparison of energy consumption and training time per epoch using IBM Qiskit in emulation mode across different GPU systems.

for ANN and LSTM (using Tensorflow), and CatBoost, and Quantum Models for QNN and QLSTM (using PennyLane, for ansätze A1–A10). The results are summarized in Table 12.

Classical models completed training within a relatively short timeframe, ranging from less than ten seconds for CatBoost (8.88×10^{0} s) to approximately two minutes for LSTM (1.15×10^{2} s). ANN exhibited moderate training time of about a minute (5.84×10^{1} s).

In contrast, quantum models demonstrated significantly longer training durations. The QNN required a median of 1.98×10^{3} seconds (approximately 33 minutes), while the QLSTM extended further to 2.66×10^{3} seconds (about 44 minutes), representing a one to two order-of-magnitude increase in training time relative to classical counterparts. This disparity is primarily due to the computational cost of quantum circuit simulation and repeated expectation value measurements when executed on emulated quantum environments using classical computers.

A similar, and related, trend was observed in energy usage. Classical models consumed minimal energy, ranging from 1.20×10^{-4} kWh for CatBoost to 1.54×10^{-3} kWh for LSTM. In contrast, quantum models exhibited markedly higher energy consumption, with QNN consuming 2.92×10^{-1} kWh and QLSTM peaking at 4.55×10^{-1} kWh.

These results emphasize the substantial energy cost associated with quantum machine learning when run in classical simulation environments. This overhead arises primarily from the need to emulate qubit interactions, apply parameterized gates repeatedly, and perform stochastic sampling to estimate observables–processes that scale exponentially with the number of qubits.

This discrepancy underscores the computational overhead of simulating quantum circuits on classical hardware. Unlike classical models that have already seen tremendous algorithmic development and optimization, quantum circuit simulators are still in early stages of development and require complex linear algebraic operations, which are computationally expensive even on high-performance GPUs. These results highlight the fact that while theoretical quantum advantage promises asymptotic gains in performance or efficiency, practical implementations continue to be impacted by the limitations of classical simulation.

Phase 3: comparison of energy consumption between quantum emulation and physical quantum hardware

In order to quantify and compare the energy consumption between quantum neural network (QNN) models operating in simulated versus actual quantum hardware environments, we conducted a dedicated set of experiments using IBM Quantum systems and the Qiskit framework. Simulated training was executed on a high-performance emulation environments powered by NVIDIA A100 GPUs, with complete system specifications outlined in Table 8. These simulations served as a baseline for understanding the resource demands under idealized execution conditions. The results are described in Table 13.

Due to time and access constraints associated with the IBM Quantum hardware, training the model on the full dataset was not feasible. To address this, the original dataset was segmented into smaller batches of 50 records each, allowing each training run to complete within the available execution window. Multiple such runs were conducted, and their cumulative execution time was used to extrapolate the total training duration that would have been required for the full dataset. By combining this extrapolated training time with published estimates of the power draw of IBM Quantum hardware, approximately 25 kW, we estimated the total energy consumption for executing the QNN training on actual quantum systems. These numbers are detailed in Table 14.

The energy cost of training a QNN model on actual IBM Quantum hardware was estimated by first measuring the time required to process batches of 50 samples and extrapolating the total runtime needed for a complete epoch across 112,500 training samples. Given the average per-batch execution time of 68 s, the estimated epoch-level runtime amounted to approximately 151,875 seconds. Using a widely cited power consumption estimate

Metric	Value
Number of samples in training dataset	112,500
Number of samples per batch	50
Number of batches needed for 1 epoch	2250
Mean runtime of 1 batch (seconds)	68
Approximate execution time for 1 epoch (seconds)	151,875
Approximate energy consumption for 1 epoch (kWh)	1055

Table 14. Estimation of energy consumption per epoch for model training on IBM Quantum hardware.

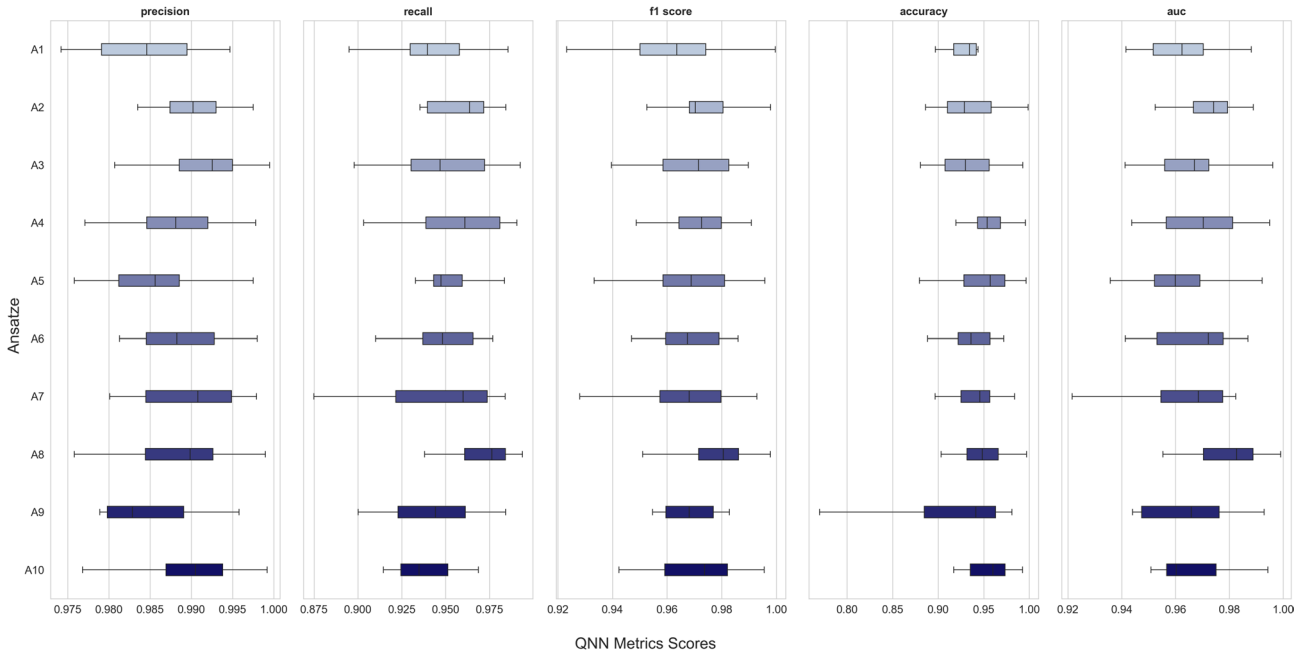


Fig. 5. QNN accuracy metrics for various ansatze.

of 25 kW for quantum computing infrastructure, this corresponds to an energy footprint of approximately 1055 kWh per epoch.

By contrast, the same model executed in a Qiskit-based emulation environment on high-performance classical hardware exhibited substantially lower energy consumption. Training runs conducted on three GPU platforms, NVIDIA A100 (40 GB and 80 GB), yielded the following per-epoch energy usage: 0.2276 kWh (A100-80GB) and 1.2639 kWh (A100-40GB). These results demonstrate that the energy required for emulated quantum training is several orders of magnitude lower than that of real quantum systems under current hardware implementations.

This disparity can be attributed primarily to the non-computational overhead intrinsic to quantum hardware. Contemporary superconducting quantum computers demand continuous cryogenic cooling, precision control electronics, and error correction subsystems, all of which operate independent of the quantum workload size. As such, a significant portion of the energy consumption is dedicated to maintaining quantum coherence and system stability rather than direct computation.

In contrast, classical emulation environments benefit from energy-proportional hardware, where power usage scales more directly with task complexity and duration. The A100-40GB energy-intensive emulation consumed over 800× less energy than IBM Quantum hardware for the same training task, while the more efficient A100-80GB setup demonstrated an even larger efficiency gap exceeding 4000×.

These findings highlight that quantum hardware currently lacks energy efficiency advantages in its present form. While QML models may show potential in terms of accuracy or expressiveness, their environmental and operational costs are significant. As the field progresses, advances in hardware efficiency will be necessary before such systems become competitive in terms of sustainability and energy economics.

Phase 1 results revisited: performance analysis

To complement the energy and environmental assessments, we evaluated the predictive performance of the QNN and QLSTM models across the same ten ansätz configurations (A1 to A10), using standard classification metrics of Accuracy, Precision, Recall, F1-Score, and AUC-ROC. These metrics, detailed in Table 11, provide a

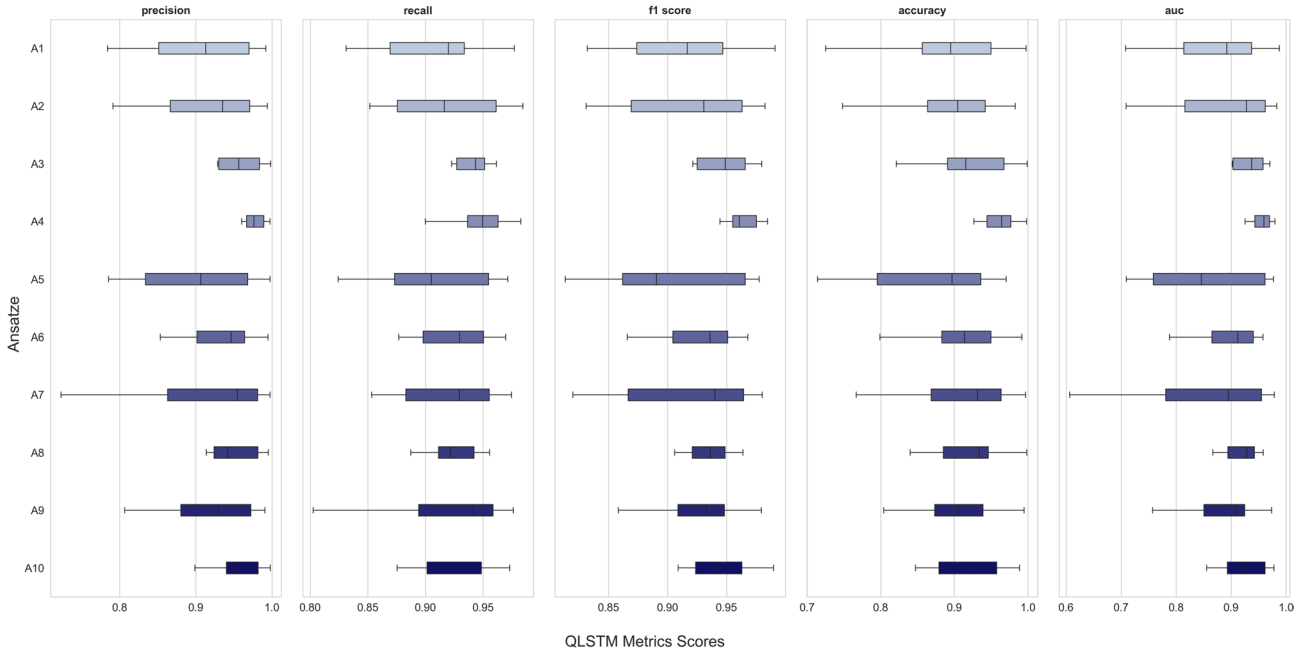
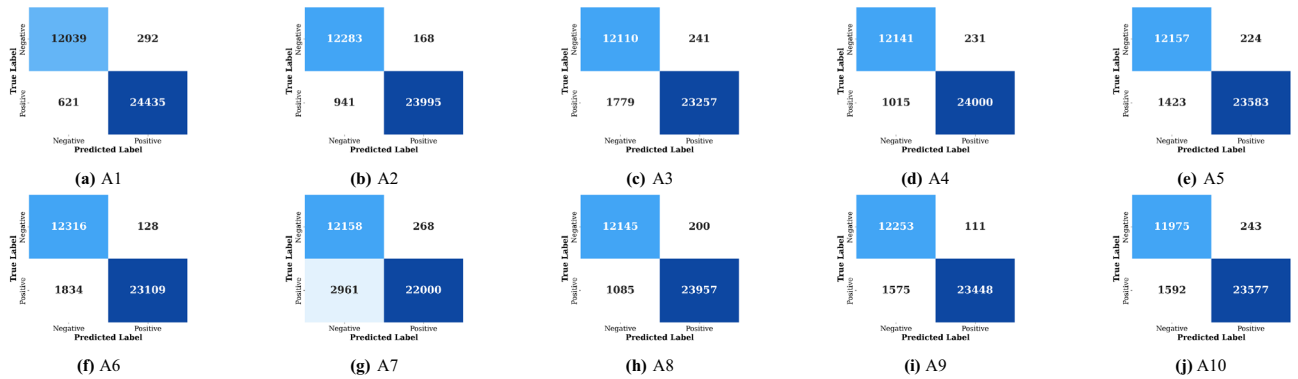


Fig. 6. QLSTM accuracy metrics for various ansatze.



(a) A1 **(b)** A2 **(c)** A3 **(d)** A4 **(e)** A5

(f) A6 **(g)** A7 **(h)** A8 **(i)** A9 **(j)** A10

Fig. 7. Confusion matrices for QNN with various ansätze.

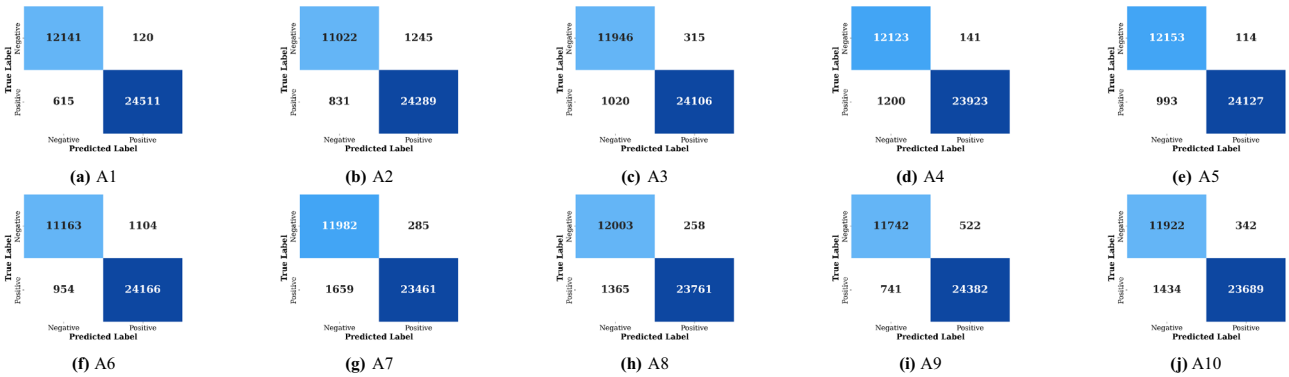
comprehensive view of the models' capabilities for anomaly detection in IoT networks. These metrics are also visually represented in Figs. 5 and 6.

Across most configurations, QNN models continued to demonstrate stronger performance consistency and superior accuracy. QNN achieved accuracy scores between 0.9288 and 0.9602, with multiple configurations such as A4, A5, A8, and A10 exceeding 0.94. In contrast, QLSTM exhibited greater variability, with accuracies ranging from 0.8952 to 0.9644, but only A4 surpassed the 0.96 mark.

QNN models also outperformed QLSTM in precision, with most values above 0.98, indicating minimal false positives. QLSTM, while achieving strong precision in select configurations (A3, A4, A6), showed lower values overall, with wider standard deviations reflecting less consistent performance. Notably, QNN's precision remained tightly clustered with lower variance, suggesting better reliability.

In terms of recall, both models performed well, but QNN again led slightly. The QNN recall ranged from 0.9396 to 0.9765, while QLSTM ranged from 0.9050 to 0.9495, with a slightly higher spread in results. This suggests that QNN models were more effective in capturing true positives, a critical factor in security contexts where missing an attack can have serious implications.

The F1-score also highlights QNN's dominance, with all configurations achieving scores above 0.96, peaking at 0.9806 (A8). In comparison, QLSTM F1-scores were generally lower and more variable, ranging from 0.8907 to 0.9607, with only A4 nearing the QNN levels. The AUC-ROC values further support these findings, with QNN configurations consistently recorded AUCs above 0.95. QLSTM showed greater fluctuation, including a low of 0.8459 (A5) and a high of 0.9596 (A4). This variability suggests that QLSTM models may struggle with consistent class separation across different ansätze, while QNNs provide more dependable performance.



(a) A1 **(b)** A2 **(c)** A3 **(d)** A4 **(e)** A5

(f) A6 **(g)** A7 **(h)** A8 **(i)** A9 **(j)** A10

Fig. 8. Confusion matrices for QLSTM with various ansätze.

Metric	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10
Accuracy	0.1167	**0.0439**	0.4094	0.4989	**0.0006**	0.1404	0.2235	0.0989	0.2235	0.1136
Precision	**0.0002**	**0.0000**	**0.0001**	**0.0207**	**0.0000**	**0.0000**	**0.0001**	**0.0001**	**0.0018**	**0.0007**
Recall	**0.0106**	**0.0040**	0.1264	0.1762	**0.0090**	**0.0155**	**0.0385**	**0.0000**	0.5609	0.6359
F1-Score	**0.0013**	**0.0002**	**0.0021**	0.0583	**0.0008**	**0.0000**	**0.0023**	**0.0000**	**0.0098**	**0.0047**
AUC ROC	**0.0005**	**0.0001**	**0.0002**	**0.0265**	**0.0051**	**0.0000**	**0.0002**	**0.0000**	**0.0028**	**0.0053**

Table 15. Mann–Whitney U Test p-values (QNN vs. QLSTM) across all ansätze. Bold values indicate statistically significant differences ($p < 0.05$).

Looking at the confusion matrices in Figs. 7 and 8, we find that across all ansätze, QNN consistently achieved higher accuracy than QLSTM, albeit marginally. This trend suggests that QNNs were able to generalize slightly better in this classification task. QNN models also displayed higher precision (>98%) across ansätze as compared to QLSTM models.

To evaluate whether the observed performance differences between QNN and QLSTM models are statistically significant across various ansätze, we conducted the non-parametric Mann–Whitney U test on five key classification metrics: Accuracy, Precision, Recall, F1-Score, and AUC ROC. The results, shown in Table 15, indicate that Precision, F1-Score, and AUC ROC metrics consistently exhibit statistically significant differences ($p < 0.05$) across nearly all ansätze, suggesting that QNN and QLSTM models differ meaningfully in how they optimize classification boundaries. These metrics highlight QNN's advantage in balancing classification reliability and discriminative capacity. For Accuracy, statistical differences are more nuanced, only A2 and A5 exhibit significant divergence, indicating that accuracy alone may not capture the broader performance dynamics. Overall, the QNN model demonstrates consistently better median precision and F1 performance with lower variance across trials. These findings affirm that in general the QNN configurations show better performance compared to identical QLSTM configurations. Given that QNN models also demonstrated lower carbon emissions in prior sections, this makes QNN based designs more favorable as compared to QLSTM based designs.

Deeper look at emissions and energy consumption

Figure 9 illustrates the total CO_2 emissions, in kilograms, generated during the training of QNN and QLSTM models across ten distinct ansätze (A1–A10). This comparison directly reflects the environmental cost of quantum model training. The complexity of the different circuit architectures directly influences power usage and sustainability profiles. A1 and A4 were the most carbon-efficient configurations for both QNN and QLSTM models, with emissions at or under 0.09 kg, while A6 was the least carbon-efficient.

Figure 10 illustrates the computational demands of various ansätze and their implications for performance optimization and energy sustainability. Across all ansätze, GPU energy dominates total energy usage. A6 shows the highest GPU consumption, in accordance with its gate-heavy circuit structure. In contrast, A4 stands out in both models with the lowest energy consumption, aligning with its overall minimal energy profile. CPU energy remains modestly variable, with A6 demanding the most. RAM consumption contributes the smallest fraction of total energy across all ansätze. QLSTM RAM usage is higher than QNN for all ansätze owing to its memory-intensive architecture.

The diagram in Fig. 11 shows the ansätze performances in the context of their energy costs, which is especially relevant for large-scale or environmentally conscious deployments. This chart clearly illustrates that QNN models consistently outperform QLSTM in accuracy for a given energy budget. A4 is notable for achieving the highest accuracy ($\approx$ 96%) at moderate energy consumption, thus representing a Pareto-optimal configuration within the group. QLSTM models, in general, show diminishing returns, consuming more energy per unit of

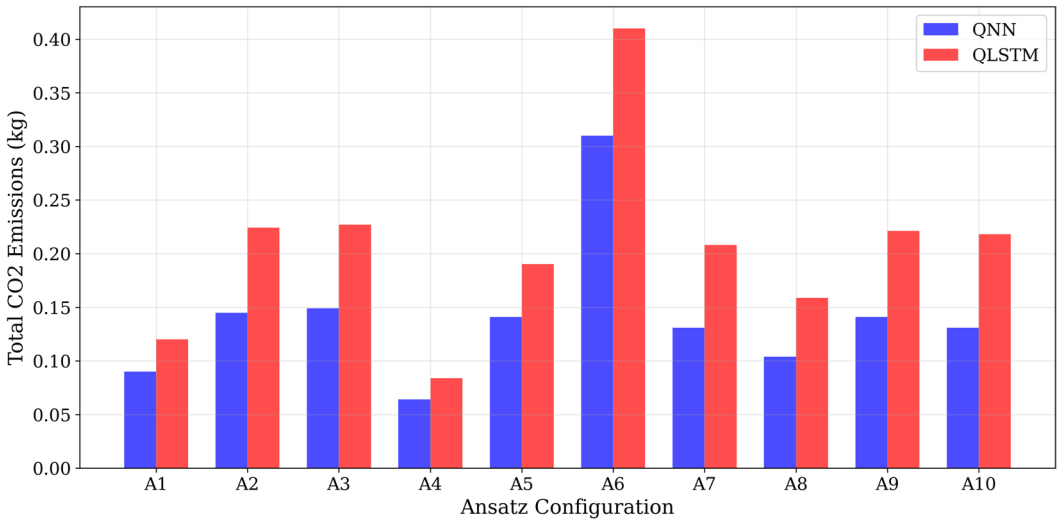


Fig. 9. Carbon emissions by model configuration. The environmental impact varies significantly across ansätz configurations.

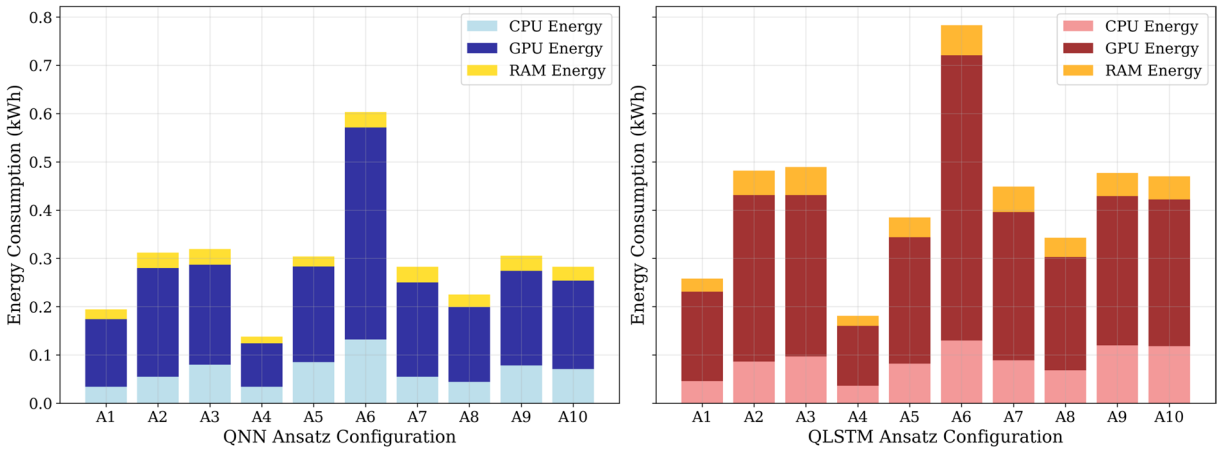


Fig. 10. Hardware energy consumption breakdown. GPU energy dominates total consumption for both models.

accuracy. With QLSTM, only A4 crosses the 94% accuracy threshold while consuming the least energy, making it an outlier for QLSTM.

Figure 12 examines the convergence behavior and generalization performance of QNN A4 and QLSTM A4 models, which were the two most promising ansätz configurations. QNN A4 exhibits a steep learning trajectory, with both training and test accuracy surpassing 90% by epoch 5, and plateauing near 97% by epoch 7. It shows stable performance, with the variance narrowing over time, indicating good learning and convergence. The gap between training and test accuracy remains minimal, suggesting minimal overfitting. QLSTM A4, on the other hand, reaches 94% test accuracy by epoch 10, with a relatively slower slope in the early epochs, even though it starts with a higher base accuracy ($\approx$ 75%) in the first epoch. While the gap between train and test accuracy is small, it saturates earlier and at a lower performance level than QNN A4. Additionally, the variance in test accuracy is smaller in early epochs compared to QNN.

The training and test loss curves for three representative ansätze, A1, A4, and A8, across QNN and QLSTM models, are shown in Fig. 13. The top row shows the training and test loss for QNN while the bottom row shows the same for QLSTM for A1, A4, and A8 ansätze. For QNN, all three ansätze exhibit smooth, monotonic convergence. A4 and A8 outperform A1, converging to a lower final training loss (($\approx$ 0.05). A1 trails slightly, leveling at $\approx$ 0.09, owing to its design of that of a relatively less expressive and shallower ansätz. A8 and A4 again show good generalization on the test dataset, with the parallel trajectories between training and test losses for A4 and A8 suggesting minimal overfitting. In QLSTM models, A4 demonstrates the fastest and steepest convergence, reaching $\approx$ 0.27 by epoch 10. A1 and A8 converge more slowly. The difference in convergence speed is more pronounced in QLSTM than in QNN. On test dataset, A4 again achieves the lowest final test loss indicating excellent generalization. A8 performs moderately ($\approx$ 0.07), while A1 levels off at a higher loss ($\approx$

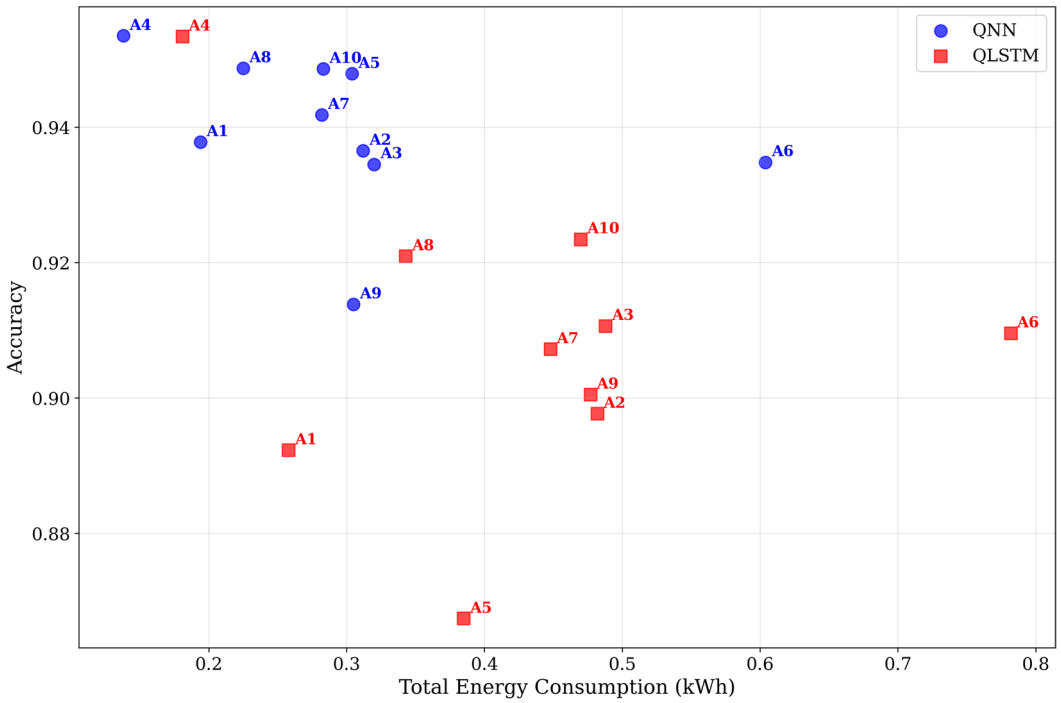


Fig. 11. Energy-accuracy trade-off analysis. QNN achieves higher accuracy with variable energy consumption.

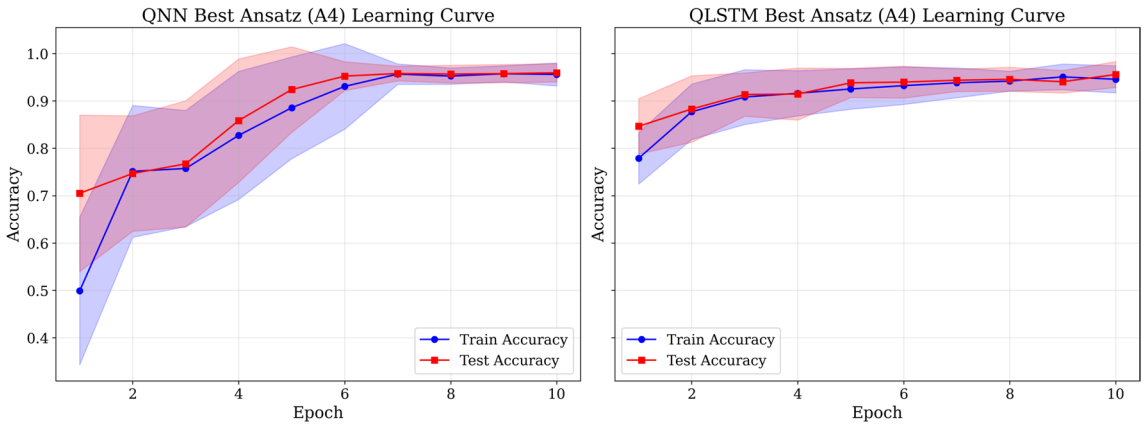


Fig. 12. Learning curves for optimal configurations. QNN A8 demonstrates faster convergence than QLSTM A4.

0.11–0.12). Additionally, A1 exhibits higher variance in test loss (visible through minor oscillations post-epoch 5), which could suggest early overfitting.

Figure 14 represents the holistic evaluation of the classification capabilities of QNN and QLSTM models for the average taken on the results from the ten ansätze A1 - A10. The comparison makes use of five metrics (Accuracy, Precision, Recall, F1-Score, and AUC-ROC). As represented visually in the diagram, QNN (Blue pentagon) dominates in all five metrics, and its balanced shape suggests consistency across metrics. QLSTM (Red pentagon) exhibits slightly lower but uniform performance across the metrics. This radar chart confirms that QNN consistently outperforms QLSTM in all of the classification metrics evaluated.

To assess the practical viability of QNN and QLSTM architectures, Figure 15 shows a twofold view of their training efficiency. The left panel shows the average time per epoch across ten different ansätze, while the right panel shows final test accuracy against total training duration. As seen on the left chart, QNN consistently incurs higher average training time per epoch as compared to QLSTM for most ansätze. The most computationally expensive QLSTM configuration is A6 (median $\approx$ 3551/epoch), while for QNN it is A7 (median $\approx$ 2189 s/epoch). The fastest ansätz is A4 (median $\approx$ 961 s for QNN and $\approx$ 1118 s for QLSTM). Looking at the right chart, we see that QNN achieves higher accuracy consistently, with most data points above 0.90, even for moderate training times. Additionally, for all training times, QNN accuracy exhibits low variance and strong

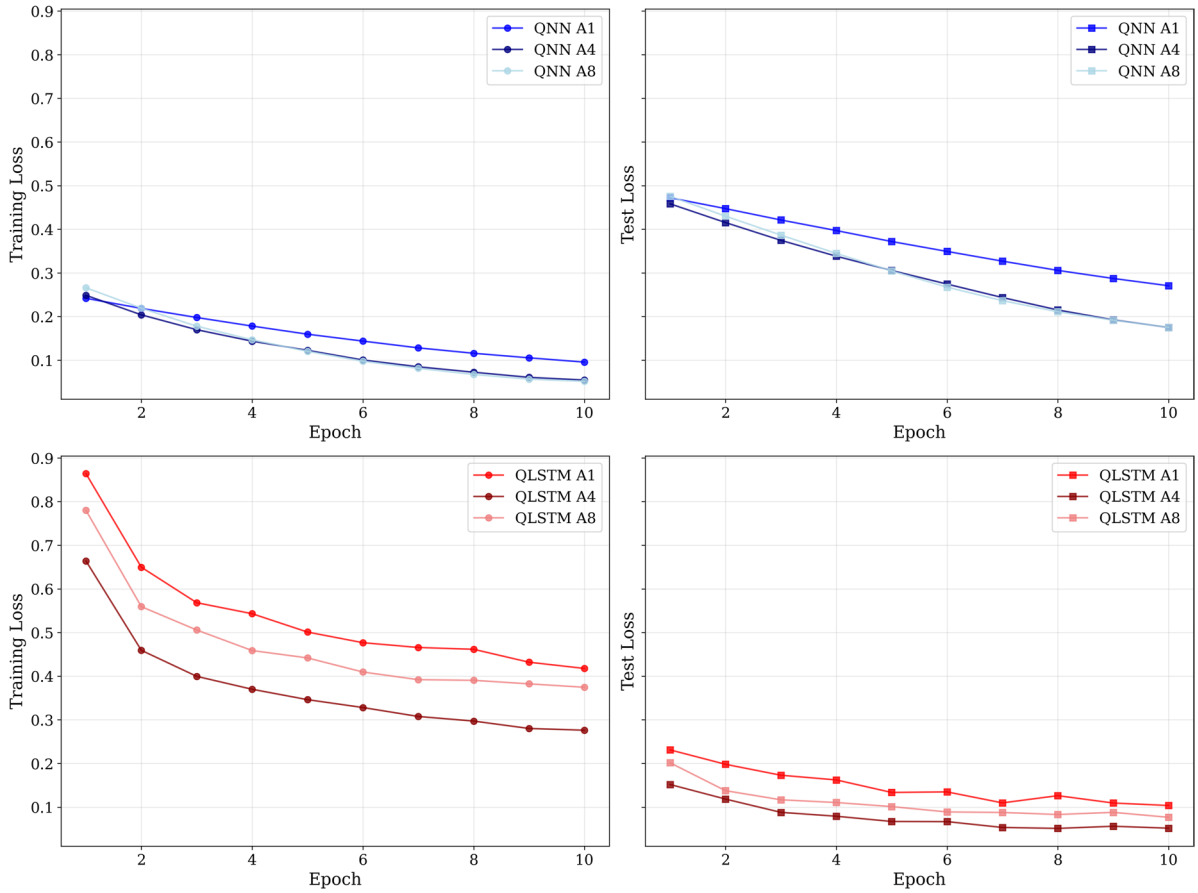


Fig. 13. Loss convergence over training epochs. QNN demonstrates faster convergence than QLSTM. A4 performs best on both QNN and QLSTM models.

generalization. QLSTM demonstrates greater scatter in both accuracy and training time. While some QLSTM models reach 0.95 accuracy, many fail to reach that level even after extended training durations, with accuracy dropping below 0.85 or even 0.75 in some cases.

Figure 16 depicts the energy efficiency of the QNN and QLSTM models by evaluating the correlation between training time (in minutes) and total energy consumption (in kWh) across the ten distinct ansätze. As the diagram visually shows, both QNN and QLSTM models exhibit a positive correlation between training time and energy consumption. QLSTM models generally lie higher on the energy axis for comparable training durations, indicating higher energy demands due to quantum circuit complexity. A4 (in both QNN and QLSTM) presents an ideal configuration for time-energy balance. A6 consistently consumes the most energy and time across both models, owing to its circuit depth and complexity. Overall, this diagram establishes that the training time is a strong predictor of energy cost for both QNN and QLSTM models and that QLSTM models tend to be more energy-demanding as compared to QNN.

Discussion

This study presents a comprehensive evaluation of Quantum Machine Learning through a three-phase experimental framework designed to assess performance, energy efficiency, and hardware implications across quantum and classical paradigms. In the first phase, we conducted a comparative assessment of QNN and QLSTM architectures across ten distinct ansätze. This phase revealed critical trade-offs between classification accuracy, energy usage, and training overhead. QNN consistently outperformed QLSTM across all major evaluation metrics–including Accuracy, Precision, Recall, F1-score, and AUC-ROC–with the QNN-A4 ansatz delivering the best overall performance. Furthermore, certain ansätze, such as A6, were notably more energy-intensive and slower to train, highlighting the importance of designing resource-efficient circuits. On the other hand, ansätze like A4 and A8 achieved favorable energy-accuracy trade-offs, positioning them as ideal candidates for practical deployment.

The second phase extended the investigation to benchmark QML models against three classical machine learning architectures: ANN, LSTM, and CatBoost. This analysis enabled a holistic view of energy-performance trade-offs. Classical models demonstrated orders-of-magnitude lower training times and energy consumption compared to their quantum counterparts, with CatBoost being the most efficient.

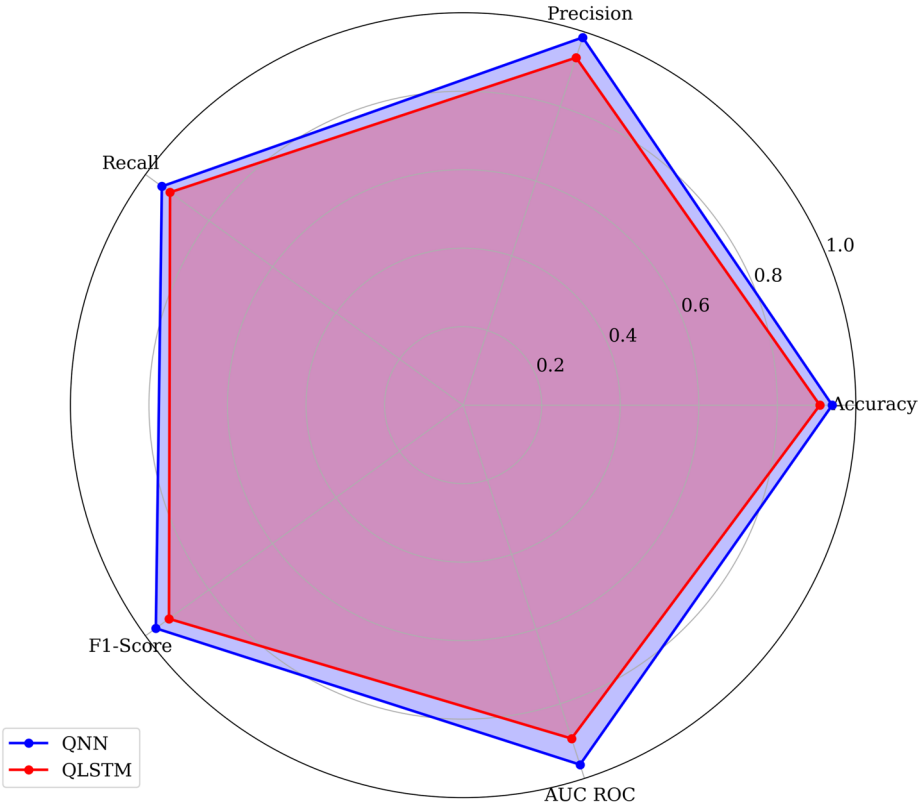


Fig. 14. Multi-metric performance comparison. QNN outperforms QLSTM across all evaluation metrics.

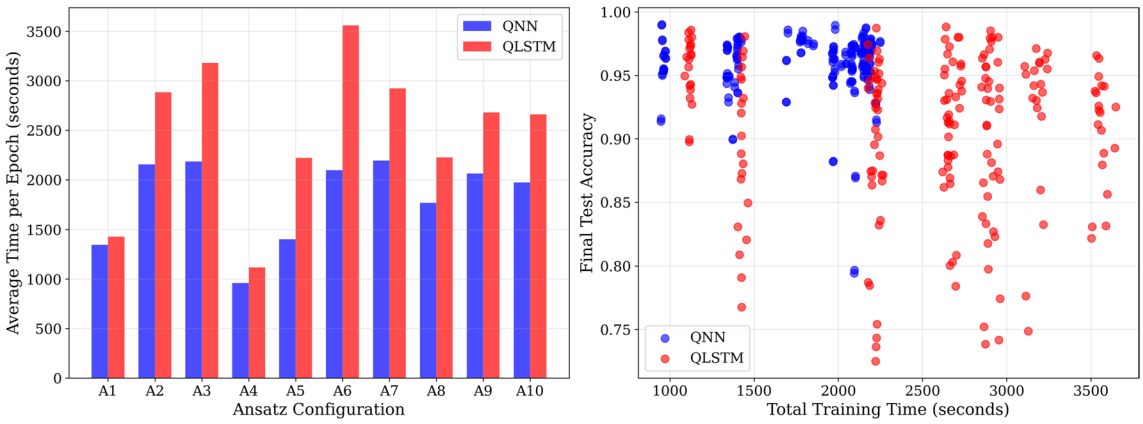


Fig. 15. Training efficiency analysis.

Within the quantum models, QNNs demonstrated better classification performance per unit energy costs, suggesting that they may serve as viable alternatives in scenarios where interpretability of quantum circuits adds value. This phase also reinforced the importance of using standardized energy profiling tools (such as CodeCarbon) to ensure consistent evaluation of carbon impact across paradigms.

In the third phase, we analyzed the energy consumption of QML models on actual IBM quantum hardware and compared it against Qiskit-based quantum emulation running on high-performance GPUs. The extrapolated energy usage on IBM's real quantum hardware was significantly higher–reaching over 1,000 kWh per epoch–compared to sub-kWh levels observed in GPU emulation environments. This stark contrast underscores the current overheads and limitations associated with quantum hardware, particularly in terms of accessibility, runtime constraints, and energy scalability. Nevertheless, these findings provide a realistic baseline for energy budgeting and emphasize the importance of circuit compression, data batching, and judicious ansätz selection in mitigating quantum training costs.

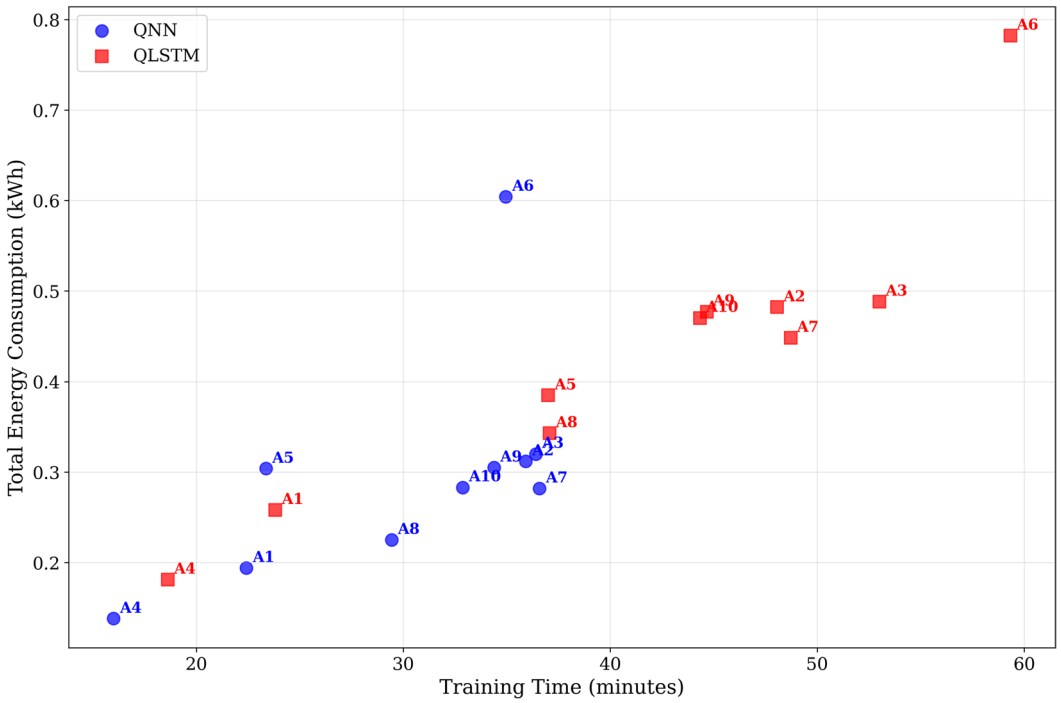


Fig. 16. Training time versus energy usage.

Collectively, these results highlight that while classical models currently dominate in terms of energy efficiency and accessibility, QML models, especially QNNs, hold promising potential for cases where appropriate. The insights into ansätz-driven energy behaviors, cross-paradigm performance benchmarking, and real-vs-emulated hardware analysis contribute to a more nuanced understanding of the practical considerations in building scalable and carbon-aware quantum AI systems.

Unexpected findings

While it was anticipated that deeper or more complex ansätz configurations would generally lead to higher model performance, one of the more surprising outcomes was that A4 delivered the highest classification accuracy with relatively lower energy consumption and training time compared to several other configurations (e.g., A6), which consumed significantly more resources without proportionate performance gains.

This challenges the conventional assumption that expressibility and depth directly translate to better generalization in QML models. The efficiency of A4 suggests that there exists an optimal complexity-to-efficiency trade-off, where moderately complex circuits can outperform deeper ones if better aligned with the data and training dynamics. This result suggests for a more principled approach to ansätz design based on empirical energy-performance profiles, rather than defaulting to depth or entanglement as proxies for capability.

Trade-offs and broader implications

One of the central insights emerging from this study is the distinct trade-off between model accuracy and resource efficiency (energy consumption) across both QNN and QLSTM architectures. While certain ansätz configurations (e.g., QNN-A6 and QLSTM-A6) achieved decent classification performance, they did so at a significantly higher cost in terms of CO_2 emissions, GPU energy draw, and prolonged training durations. Conversely, QNN-A4 and QLSTM-A4 demonstrated a good balance between high accuracy and minimal energy footprint and training time. These findings indicate that beyond a certain threshold, increasing ansätz complexity leads to diminishing returns in accuracy while increasing computational overhead. This suggests the need for a more judicious ansätz selection process.

The broader implications of this study take us into the domain of Sustainable-AI. As quantum computing moves toward practical deployment, especially in resource-constrained and eco-sensitive contexts, it becomes imperative to look into minimizing environmental impact without sacrificing performance. This study finds empirical evidence that eco-sensitive QML is attainable if the choice of ansätz design is strategically aligned.

Strengths, limitations, and future work

A notable strength of this work lies in its comprehensive and multi-faceted evaluation framework for comparing QNN and QLSTM architectures across ten distinct ansätz configurations. The study rigorously benchmarks models not only on conventional classification metrics, but also incorporates environmental and operational considerations, including CO_2 emissions, energy breakdown by hardware component, training time, and convergence dynamics. This integrated approach allows for a holistic understanding of performance-to-efficiency trade-offs. This work highlights configurations that optimally balance accuracy and sustainability.

Additionally, the consideration of real-world system-level energy and emission data adds practical relevance to the experimental outcomes.

On the other hand, this study also has several limitations. First, the experiments were conducted in a simulated quantum environment. They do not account for real quantum noise, decoherence, and hardware-specific constraints which may significantly impact actual model performance on near-term quantum hardware. Additionally, modern quantum computing infrastructure is significantly more complex than classical computing infrastructure and has more dimensions to energy consumption than just GPU, CPU, and RAM which were not studied as part of this work. Second, the dataset and classification task selected for this evaluation may not be generalizable across all domains, and results may vary when applied to more complex or higher-dimensional problems. Lastly, while the ten ansätze provide a meaningful spectrum of architectural diversity, the search space remains constrained, and additional exploration of circuit depth, entanglement patterns, and hybrid designs could further enrich the comparative analysis.

Additionally, as done in the third phase of the study, we compared the energy profiles of Quantum Machine Learning models between a Qiskit-based emulation environment and an actual quantum processing environment. The strength of this approach lies in the novel attempt to bridge simulation-based experimentation with real quantum hardware evaluation. This dual-perspective analysis provided valuable insights into the scalability and sustainability of QML models when transitioned from controlled emulation to physical execution. However, a notable limitation in this experiment was the constrained access to IBM Quantum hardware, which prevented complete execution on the full dataset. Consequently, energy consumption metrics for full-scale training were derived through extrapolation from smaller batch runs. While this approach offered a reasonable estimation framework, the extrapolated results are subject to uncertainty due to non-linear hardware behaviors and potential variations in qubit stability and execution latency. Future work should aim to extend this analysis using expanded runtime allocations or hybrid sampling strategies to capture more representative hardware measurements. Additionally, integrating real-time telemetry data from quantum systems such as power draw, cooling overhead, and qubit utilization would enhance the precision of energy modeling. Such advancements would help refine our understanding of true energy costs associated with quantum workloads and improve the fidelity of sustainable quantum AI research.

Given the increased complexity and operational variability inherent in modern quantum computing infrastructure, future studies should prioritize empirical validation of energy-to-performance trade-offs on real quantum hardware. Simulated environments, while useful for preliminary benchmarking, cannot fully capture the stochasticity introduced by quantum noise, qubit decoherence, gate fidelity variations, and readout errors. As quantum computing becomes more prevalent and accessible, the study and quantification of the environmental impact becomes all the more important for sustainable AI. In addition, real-time energy monitoring integrated into quantum hardware stacks will be crucial to establishing standardized, cross-platform protocols for measuring carbon impact in future quantum AI workflows.

In parallel, ansätz design and optimization should evolve toward hardware-aware and task-adaptive architectures. The findings of this study suggest that more complex circuits do not always yield better outcomes, especially when considering energy and training time. Future work should investigate automated or learning-based ansätz search strategies that co-optimize for complexity, energy efficiency, and robustness to noise. Additionally, a broader group of QML tasks should be evaluated to generalize these insights beyond binary classification. Interdisciplinary approaches that combine quantum compiler optimization, variational algorithm tuning, and environmental modeling may also yield valuable frameworks for developing sustainable and scalable quantum AI strategies.

Data availability

1. The open dataset used in this study (N-BaIoT) is publicly available on Kaggle: https://www.kaggle.com/datasets/mkashifn/nbaiot-dataset 2. The code supporting the findings of this study is available from the corresponding author, Sarvapriya Tripathi (strip011@fiu.edu), upon reasonable request.

Received: 9 June 2025; Accepted: 11 November 2025
Published online: 29 December 2025

References

1. Nielsen, M. A. & Chuang, I. L. *Quantum Computation and Quantum Information* 10th Anniversary. (Cambridge University Press, Cambridge, New York, 2010).
2. Preskill, J. Quantum computing in the NISQ era and beyond. *Quantum* **2**, https://doi.org/10.22331/q-2018-08-06-79 (2018). Publisher: Verein zur Forderung des Open Access Publizierens in den Quantenwissenschaften, arXiv: 1801.00862.
3. Schuld, M., Sinayskiy, I. & Petruccione, F. An introduction to quantum machine learning. *Contemp. Phys.* **56**, 172–185. https://doi.org/10.1080/00107514.2014.964942 (2015).
4. Biamonte, J. et al. Quantum machine learning. *Nature* **549**, 195–202. https://doi.org/10.1038/nature23474 (2017) Publisher: Nature Publishing Group, arXiv: 1611.09347.
5. Schetakis, N., Aghamalyan, D., Griffin, P. & Boguslavsky, M. Review of some existing QML frameworks and novel hybrid classical–quantum neural networks realising binary classification for the noisy datasets. *Sci. Rep.* **12**, 11927. https://doi.org/10.1038/s41598-022-14876-6 (2022).
6. Farhi, E. & Neven, H. Classification with quantum neural networks on near term processors. ArXiv (2018). arXiv: 1802.06002.
7. Schwartz, R., Dodge, J., Smith, N. A. & Etzioni, O. Green AI. *Commun. ACM* **63**, 54–63. https://doi.org/10.1145/3381831 (2020).
8. Tabbakh, A. et al. Towards sustainable AI: A comprehensive framework for green AI. *Discover Sustain.* **5**, 408. https://doi.org/10.1007/s43621-024-00641-4 (2024).
9. Dhar, P. The carbon impact of artificial intelligence. *Nat. Mach. Intell.* **2**, 423–425. https://doi.org/10.1038/s42256-020-0219-9 (2020).

10. Ikonen, J., Salmilehto, J. & Möttönen, M. Energy-efficient quantum computing. *npj Quantum Inform.* **3**, 17. https://doi.org/10.1038/s41534-017-0015-5 (2017).
11. Desdentado, E., Calero, C., Moraga, M. A., Serrano, M. & Garcia, F. Exploring the trade-off between computational power and energy efficiency: An analysis of the evolution of quantum computing and its relation to classical computing. *J. Syst. Softw.* **217**, 112165. https://doi.org/10.1016/j.jss.2024.112165 (2024).
12. Li, J., Guan, Q., Tao, D. & Jiang, W. Carbon emissions of quantum circuit simulation: More than you would think. In *Proceedings of the 14th International Green and Sustainable Computing Conference* 11–13 (ACM, 2023). https://doi.org/10.1145/3634769.3634800
13. Henderson, P. et al. Towards the systematic reporting of the energy and carbon footprints of machine learning. *J. Mach. Learn. Res.* https://doi.org/10.5555/3455716 (2020).
14. Hershcovich, D., Webersinke, N., Kraus, M., Bingler, J. A. & Leippold, M. Towards climate awareness in NLP research, https://doi.org/10.48550/arXiv.2205.05071. arXiv: 2205.05071 [cs].
15. Giani, A. & Eldredge, Z. Quantum computing opportunities in renewable energy. *SN Comput. Sci.* **2**, 393. https://doi.org/10.1007/s42979-021-00786-3 (2021).
16. Kim, T. H. & Madhavi, S. Quantum intrusion detection system using outlier analysis. *Sci. Rep.* **14**, 27114. https://doi.org/10.1038/s41598-024-78389-0 (2024).
17. Salek, M. S. et al. A novel hybrid quantum-classical framework for an in-vehicle controller area network intrusion detection. *IEEE Access* **11**, 96081–96092. https://doi.org/10.1109/ACCESS.2023.3304331 (2023).
18. Kukliansky, A., Orescanin, M., Bollmann, C. & Huffmire, T. Network anomaly detection using quantum neural networks on noisy quantum computers. *IEEE Trans. Quantum Eng.* **5**, 1–11. https://doi.org/10.1109/TQE.2024.3359574 (2024).
19. Guo, M. et al. Quantum algorithms for anomaly detection using amplitude estimation. *Physica A* **604**, 127936. https://doi.org/10.1016/j.physa.2022.127936 (2022).
20. Kambourakis, G., Kolias, C. & Stavrou, A. The mirai botnet and the IoT zombie armies. In *MILCOM 2017 - 2017 IEEE Military Communications Conference (MILCOM)* 267–272 (IEEE, 2017) https://doi.org/10.1109/MILCOM.2017.8170867.
21. Marzano, A. et al. The evolution of bashlite and mirai IoT botnets. In *2018 IEEE Symposium on Computers and Communications (ISCC)* 00813–00818 (IEEE, 2018) https://doi.org/10.1109/ISCC.2018.8538636.
22. Meidan, Y. et al. N-BaIoT-network-based detection of IoT botnet attacks using deep autoencoders. *IEEE Pervasive Comput.* **17**, 12–22. https://doi.org/10.1109/MPRV.2018.03367731 (2018).
23. Xiao, T., Zhai, X., Huang, J., Fan, J. & Zeng, G. Quantum deep generative prior with programmable quantum circuits. *Commun. Phys.* **7**, 276. https://doi.org/10.1038/s42005-024-01765-9 (2024).
24. Xu, H. et al. Toward Heisenberg Limit without Critical Slowing Down via Quantum Reinforcement Learning. *Phys. Rev. Lett.* **134**, 120803. https://doi.org/10.1103/PhysRevLett.134.120803 (2025).
25. Xiao, T., Zhai, X., Wu, X., Fan, J. & Zeng, G. Practical advantage of quantum machine learning in ghost imaging. *Commun. Phys.* **6**, 171. https://doi.org/10.1038/s42005-023-01290-1 (2023).
26. Courty, B. et al. mlco2/codecarbon: v2.4.1, https://doi.org/10.5281/zenodo.11171501 (2024).
27. Massoli, F. V., Vadicamo, L., Amato, G. & Falchi, F. A leap among quantum computing and quantum neural networks: A survey. *ACM Comput. Surv.* **55**, 1–37. https://doi.org/10.1145/3529756 (2023).
28. Chen, S. Y.-C., Yoo, S. & Fang, Y.-L. L. Quantum long short-term memory. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 8622–8626 (IEEE, 2022) https://doi.org/10.1109/ICASSP43922.2022.9747369.
29. McClean, J. R., Boixo, S., Smelyanskiy, V. N., Babbush, R. & Neven, H. Barren plateaus in quantum neural network training landscapes. *Nature Communications* **9**, https://doi.org/10.1038/s41467-018-07090-4 (Publisher: Nature Publishing Group, 2018) arXiv: 1803.11173.
30. Schuld, M., Bocharov, A., Svore, K. M. & Wiebe, N. Circuit-centric quantum classifiers. *Phys. Rev. A* **101**, 032308. https://doi.org/10.1103/PhysRevA.101.032308 (2020).
31. Meidan, Y. et al. detection_of_iot_botnet_attacks_n_baiot, https://doi.org/10.24432/C5RC8J (2018). https://www.kaggle.com/datasets/mkashifn/nbaiot-dataset.
32. Hossain, M., Al Mamun Rudro, R., Razzaque, R. & Nur, K. Machine learning approaches for detecting IoT botnet attacks: A comparative study of n-BaIoT dataset. In *2024 International Conference on Decision Aid Sciences and Applications (DASA)* 1–7 (IEEE, 2024) https://doi.org/10.1109/DASA63652.2024.10836622.
33. Xu, J., Zhou, W., Fu, Z., Zhou, H. & Li, L. A survey on green deep learning, https://doi.org/10.48550/arXiv.2111.05193 (2021). arXiv: 2111.05193 [cs].
34. Ho, K. T. M. et al. Quantum computing for climate resilience and sustainability challenges. In *2024 IEEE International Conference on Quantum Computing and Engineering (QCE)* 262–267 (IEEE, 2024) https://doi.org/10.1109/QCE60285.2024.10289.
35. Sood, V. & Chauhan, R. P. Quantum computing: Impact on energy efficiency and sustainability. *Expert Syst. Appl.* **255**, 124401. https://doi.org/10.1016/j.eswa.2024.124401 (2024).
36. Cordier, S., Thibault, K., Arpin, M.-L. & Amor, B. Scaling up to problem sizes: An environmental life cycle assessment of quantum computing. *Quantum Sci. Technol.* **10**, 025058. https://doi.org/10.1088/2058-9565/adc0ba (2025).
37. Munawar, G. & Surendro, K. Utilizing quantum algorithms to achieve carbon neutrality in urban areas: A systematic review. *Alex. Eng. J.* **108**, 911–936. https://doi.org/10.1016/j.aej.2024.09.043 (2024).
38. Arora, N. & Kumar, P. Sustainable quantum computing: Opportunities and challenges of benchmarking carbon in the quantum computing lifecycle, https://doi.org/10.48550/arXiv.2408.05679. arXiv: 2408.05679 [quant-ph].
39. Parker, E. & Vermeer, M. J. D. Estimating the energy requirements to operate a cryptanalytically relevant quantum computer (2023). arXiv: 2304.14344.
40. Boger, Y. The dual-pronged energy-saving potential of quantum computers. https://www.datacenterdynamics.com/en/opinions/the-dual-pronged-energy-saving-potential-of-quantum-computers/.
41. Castro, A. Quantum Computing in the Roadmap to Greener Calculations. https://medium.com/pasqal-io/quantum-computing-in-the-roadmap-to-greener-calculations-6adf15478509.
42. Hafeez, M. A., Munir, A. & Ullah, H. H-QNN: A hybrid quantum-classical neural network for improved binary image classification. *AI* **5**, 1462–1481. https://doi.org/10.3390/ai5030070 (2024).
43. Zhao, C. & Gao, X. S. QDNN: Deep neural networks with quantum layers. *Quantum Machine Intelligence***3**, https://doi.org/10.1007/s42484-021-00046-w (Publisher: Springer Science and Business Media Deutschland GmbH, 2021).
44. Hochreiter, S. & Urgen Schmidhuber, J. Long short-term memory. *Neural Comput.* **9**, 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735 (1997).
45. Khan, S. Z. et al. Quantum long short-term memory (QLSTM) vs classical LSTM in time series forecasting: A comparative study in solar power forecasting, https://doi.org/10.3389/fphy.2024.1439180. arXiv: 2310.17032 [quant-ph].

Acknowledgements

This research received no specific grant from any funding agency for the submitted work.

Author contributions

S.T., H.U., and J.S. conceived the experiment, S.T. conducted the experiment, and S.T. and J.S. analyzed the re-

sults. All authors reviewed the manuscript.

Declarations

Competing interests
The authors declare no competing interests.

Additional information
Correspondence and requests for materials should be addressed to S.T.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.