Quantum Machine Intelligence (2025) 7:13
https://doi.org/10.1007/s42484-024-00234-4

RESEARCH ARTICLE

Adaptive neural network for quantum error mitigation

Temitope Bolaji Adeniyi[1] · **Sathish A. P. Kumar**[1]

Received: 28 March 2024 / Accepted: 28 December 2024

Abstract
Quantum computing holds transformative promise, but its realization is hindered by the inherent susceptibility of quantum computers to errors. Quantum error mitigation has proved to be an enabling way to reduce computational error in present noisy intermediate scale quantum computers. This research introduces an innovative approach to quantum error mitigation by leveraging machine learning, specifically employing adaptive neural networks. With experiment and simulations done on 127-qubit IBM superconducting quantum computer, we were able to develop and train a neural network architecture to dynamically adjust output expectation values based on error characteristics. The model leverages a prior classifier module outcome on simulated quantum circuits with errors, and the antecedent neural network regression module adapts its parameters and response to each error characteristics. Results demonstrate the adaptive neural network's efficacy in mitigating errors across diverse quantum circuits and noise models, showcasing its potential to surpass traditional error mitigation techniques with an accuracy of 99% using the fully adaptive neural network for quantum error mitigation. This work presents a significant application of classical machine learning methods towards enhancing the robustness and reliability of quantum computations, providing a pathway for the practical realization of quantum computing technologies.

Keywords Quantum computing · Machine learning · Quantum error mitigation · Quantum neural network · Deep learning

1 Introduction

Quantum computing is the field of inquiry that uses quantum phenomena such as superposition, entanglement, and interference to operate on data represented by quantum states. This field holds the promise of revolutionizing information processing by exploiting the principles of quantum mechanics to perform complex computations exponentially faster than classical counterparts (Steane Feb. 1998; Spector 2008). While quantum computers may not yield improvements in every single aspect of computing, they can lead to substantial speedup in specific groups of problems, for example combinatorial problems (Pokharel and Lidar May 2023). However, the promise of exponentially faster computation of quantum computers than classical counterpart is critically challenged by the accuracy of measurements caused by noise in quantum systems (Preskill Aug. 2018). The biggest challenge of quantum computing is noise; noise causes errors in measurement resulting to corrupted or untrustworthy experiment. Powerful quantum algorithms like Shor and Grover implemented on quantum computers assume that qubits are perfect, but running them with the imperfect physical qubit in the present noisy intermediate scale quantum computer (NISQ) will lead to nonsense output (Cai et al. 2023). This imperfection comes from the fact that the slightest perturbation can cause the quantum state of a qubit to change or decohere. This presence of noise and error limits their performance and the accuracy in measurements output of quantum computers.

In the pursuit of scalable and fault-tolerant quantum computing, identifying, understanding, mitigating, or correcting errors is paramount. There have been a lot of efforts to reduce or mitigate noise with ways like quantum error correction (QEC) and quantum error mitigation (QEM). QEC is a crucial mechanism that enables the correction of errors that naturally occur in quantum computations. This process involves encoding quantum information redundantly to protect against errors caused by decoherence and other noise

✉ Sathish A. P. Kumar
s.kumar13@csuohio.edu

Temitope Bolaji Adeniyi
t.adeniyi@vikes.csuohio.edu

[1] Department of Electrical Engineering and Computer Science, Cleveland State University, Cleveland, OH 44115, USA

Published online: 28 January 2025

in quantum systems (Steane 2003; Chiaverini, et al. 2004; Schindler et al. May 2011; Brun 2020). The crucial task is to spot errors and decode them if present. This is because the effectiveness of QEC codes relies on the accurate identification of error types before any successful correction strategies can be implemented. Being able to fully achieve QEC will enable fully fault-tolerant quantum computing for the near-term quantum computing devices. QEM has proved to be an effective way to reduce computational errors and provide a path to useful computation before fault tolerance. This method does not rely on the encoding of quantum information like QEC; it involves classical post-processing that does not affect quantum circuit, avoiding the use of redundancy. It involves the implementation of methods trained to derive mitigated expectation values from noisy measurement executed on quantum computers (Fig. 1).

The delicate nature of quantum states makes it very important to have a profound understanding of error dynamics for the development of robust quantum mitigation techniques. Unfortunately, classical methods often fall short in capturing the complexity of quantum errors, prompting the integration of machine learning as a powerful tool to discern patterns and nuances in high-dimensional quantum datasets (Wittek 2014).

The primary objective of this research is to design an adaptive error mitigation system for quantum circuits using neural networks. Unlike static error mitigation approaches, the proposed system dynamically adjusts to the error characteristics of various quantum circuits, providing a versatile and efficient solution for quantum error mitigation. We obtain knowledge and learn about noise and use the knowledge to reduce the noise.

The research methodology involves the simulation of quantum systems with intentional errors using the noise models available on IBM quantum computer. Supervised machine learning techniques are then employed to learn the specific kind of errors present in quantum systems. The goal is to leverage machine learning algorithms to discern patterns within high-dimensional and complex quantum datasets, reflecting diverse error scenarios. Subsequently, a neural network is developed and trained to adaptively shift the output probabilities and expectation value based on the identified errors in the quantum system. This adaptive learning mechanism ensures that the neural network correctly respond to individual error patterns in quantum circuits.

The contribution of the work are as follows: The proposed adaptive error mitigation system addresses the limitations of static error correction methods by providing a solution that evolves with real-time measurement and error characteristics. This adaptability is crucial for handling the inherent variability in error types and magnitudes across different quantum systems. By leveraging machine learning and neural networks, the system aims to learn from simulated errors and generalize its error mitigation strategies, enhancing the overall reliability of quantum computations. This novel approach also aims to contribute to the ongoing efforts in achieving more useful computation with NISQ quantum computers, creating a path for future fault-tolerance quantum computers. This will also help pave the way for the practical realization of quantum computational capabilities in real-world applications.

The rest of the manuscript is organized as follows. Section II explains the related work. Section III describes the theoretical framework including the basics of quantum computing and quantum error models; it also delves into the specifics of the neural networks and adaptive learning mechanism. Section IV explains the methodology of the research, the quantum circuit simulations, data description and analysis done on the data, the supervised ML method, how the neural network was trained, and the evaluation metrics used in the research. Section V presents the experimental results of the simulations, error identification with the ML model, and the validation of the proposed adaptive error mitigation system. Section VI discusses the implications of the findings, potential applications, and future directions, and finally, section VII concludes the paper.

Fig. 1 Some common errors in quantum state measurements

2 Related works

The reviewed studies present significant advancements in quantum error mitigation, employing diverse techniques such as machine learning, clustering, and evolutionary computation to address noise challenges in NISQ devices. From leveraging classical machine learning models like random forests and artificial neural networks to employing advanced approaches like fuzzy c-means clustering and genetic algorithms, these works showcase innovative strategies to enhance the stability and accuracy of quantum computations.

Liao et al.'s (2023) work offers significant insights into addressing the challenges posed by quantum errors through classical machine learning techniques. They explore various machine learning models and highlight the advantages of using random forests in particular. Crucially, they show that ML-QEM can significantly reduce overhead while maintaining or even surpassing the accuracy of conventional methods, ultimately paving the way for more practical quantum computation. Their tests employ the popular digital zero noise extrapolation method as an added reference. They also further show how to scale ML-QEM to classically intractable quantum circuits by mimicking the results of traditional mitigation results, while significantly reducing overhead (Liao et al. 2023).

In another paper by Czarnik et al. (2021), they proposed an error mitigation method termed Clifford data regression. Their basic idea is to leverage quantum circuits predominantly composed of Clifford gates, efficiently simulatable classically, to generate training data that includes both the noisy and noiseless versions of an observable's expectation value of interest. The noisy values are obtained directly from the quantum computer, while the noiseless values are simulated on a classical computer. By fitting a linear ansatz to this data, their method enables the prediction of noise-free observables for arbitrary circuits, leading to an order-of-magnitude reduction in errors for various quantum tasks, including ground-state energy estimation and quantum phase estimation, on both quantum hardware and noisy simulators (Czarnik et al. Nov. 2021).

Kim et al. (2020) also present an approach to quantum error mitigation utilizing classical machine learning techniques. Their method employs artificial neural networks (ANNs) trained on shallow-depth quantum circuits with known measurement outcomes, incorporating gate counts and actual outcome probabilities as input features. The ANN is trained to minimize error, encompassing cumulative gate and qubit measurement errors, and subsequently used to infer error distributions for new quantum circuits. Through simulations and experiments, including IBM's quantum cloud service, the effectiveness of the method is demonstrated across various qubit configurations, showcasing error reduction by up to a factor of 1/2. Notably, the proposed scheme addresses time-dependent noise, offering mitigation even for noise frequencies lower than the training frequency (Kim et al. 2020) (Fig. 2).

Kim et al.'s (2022) approach utilizes deep learning techniques, specifically training an artificial neural network (NN) with real measurement outcomes from simple quantum circuits to correct non-linear readout errors. Unlike existing methods that rely on linear inversion and may produce non-physical results, their deep learning-based quantum readout error mitigation (QREM) protocol consistently generates physically valid outcomes. They state that their method does not require additional quantum resources beyond those needed for the quantum algorithm itself, as it only relies on gathering noisy measurement results. They experimentally validated and demonstrated the superiority of their QREM approach over traditional linear inversion methods on IBM's five-qubit quantum devices (Kim et al. Jul. 2022).

Seif et al.'s (2018) study employs machine learning techniques, a versatile neural network trained on experimental data, to enhance the accuracy of quantum computer measurements, demonstrating a 30% reduction in detection error for trapped-ion qubit readout through efficient treatment of crosstalk and offering a flexible and generalizable approach applicable to various quantum computing platforms (Seif et al. Sep. 2018).

In another approach for error mitigation, Acampora and Vitiello (2021) proposed an error mitigation approach using fuzzy c-means (FCM) clustering to improve quantum measurement accuracy in noisy intermediate-scale quantum (NISQ) devices. The method constructs mitigation matrices by clustering count vectors from quantum experiments based on their noise characteristics, effectively reducing the impact of errors on quantum computations. By leveraging the adaptability of fuzzy clustering, this approach dynamically adjusts to varying error patterns without requiring extensive prior knowledge or assumptions about the quantum system. Experimental results demonstrated that the FCM-based method outperforms traditional mitigation strategies, such as random matrix selection, in terms of stability and accuracy, particularly when dealing with high-noise environments (Acampora and Vitiello 2021).

Acampora et al. (2021) proposed a genetic algorithm (GA)–based approach for quantum error mitigation by optimizing mitigation matrices to reduce measurement errors in noisy intermediate-scale quantum (NISQ) devices. The GA iteratively refines the mitigation matrix using fitness functions to minimize the mean squared error between noisy and ideal quantum outcomes. Experimental results on simulated quantum systems showed that the GA outperformed traditional calibration matrix techniques, achieving higher

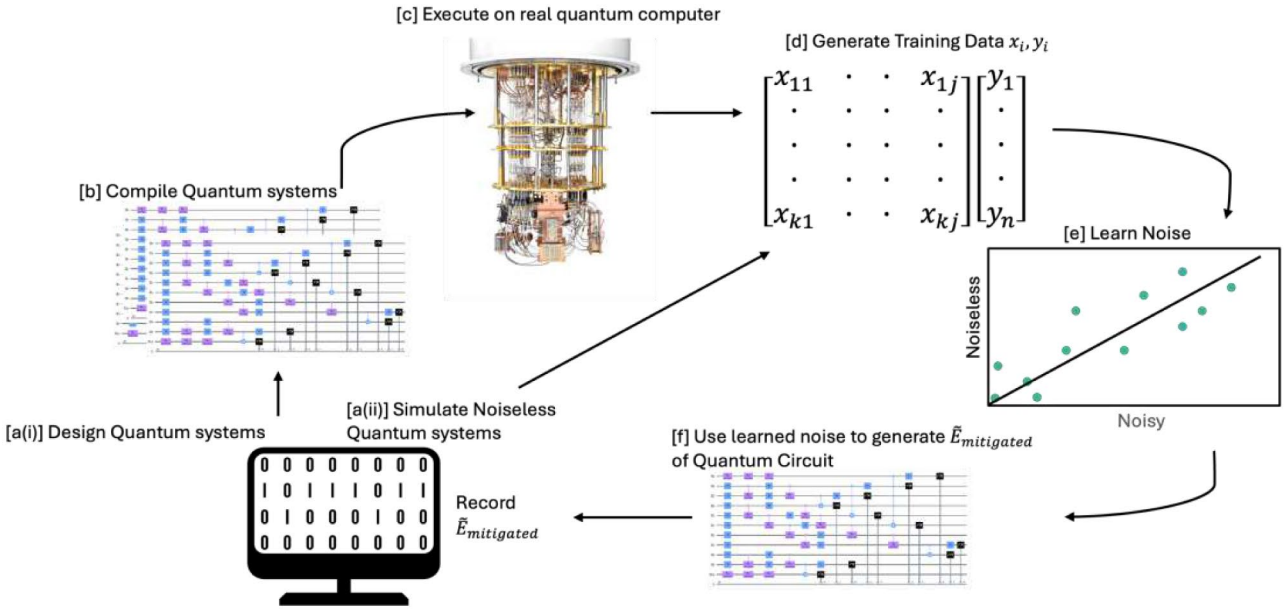

Fig. 2 The architecture of research workflow from the design stage to the final quantum error mitigation. The process begins with the design and simulation of quantum circuits comprising of both noisy and noiseless systems (**a**(i) and **a**(ii)), followed by compilation into executable quantum circuits (**b**). These circuits are executed on real quantum hardware (**c**), generating noisy measurement data, while the noiseless simulations are conducted in parallel to serve as a reference. The collected data is processed to generate training datasets $x_i y_i$ in **d**, which are used to train a noise-learning model (**e**). This model learns the error characteristics and uses the derived insights to generate mitigated expectation values $\widetilde{E}_{\text{mitigated}}$

accuracy and stability across diverse error scenarios (Acampora et al. 2021).

However, while these research present promising results, it predominantly rely on static or predefined models that may not adapt effectively to the dynamic and diverse error patterns inherent in quantum systems. In contrast, this work proposes a novel approach that dynamically adjusts the neural network's parameters based on real-time error data and error characteristics, offering a more adaptive and responsive solution to quantum error mitigation. This adaptability not only enhances its effectiveness in mitigating errors across diverse quantum circuits and noise environments but also positions it as a more versatile and robust solution for quantum error mitigation.

3 Background

3.1 A. Basics of quantum computing

As bits is the elementary unit of information in classical computing, the elementary unit of quantum information is the qubit (Schumacher Apr. 1995). In quantum mechanics, the state of a quantum system is described by a complex vector in a Hilbert space. Let $|\psi\rangle = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}$, where α and β are complex probability amplitudes corresponding to the basis states $|0\rangle$ and$|1\rangle$, respectively. Unlike classical bits that exist as either 0 or 1, a qubit can exist in a quantum superposition of both 0 and 1 simultaneously. That is, it takes the states $\alpha|0\rangle + \beta|1\rangle$. Additionally, qubits can exhibit entanglement, where the state of one qubit becomes linked to the state of another, even when separated by large distances; this allows for the creation of quantum states that are highly correlated. Both phenomena are fundamental aspect of quantum mechanics that enable quantum computers to process information in a different way than classical computer; it also allows it to perform certain calculations much faster than classical computers. The ability of qubits to exist in multiple states simultaneously and leverage entanglement forms the basis for the potential computational power of quantum computers.

While superposition and entanglement form the bedrock of quantum computational superiority, they also necessitate sophisticated error mitigation techniques to harness their benefits effectively (Reed 2013).

In quantum computing, operations are performed on qubits through quantum gates, manipulating their quantum states to process and manipulate information. These gates perform transformations on the quantum state of one or more qubits. Each gate represents a specific operation that alters the quantum state, such as changing the probability

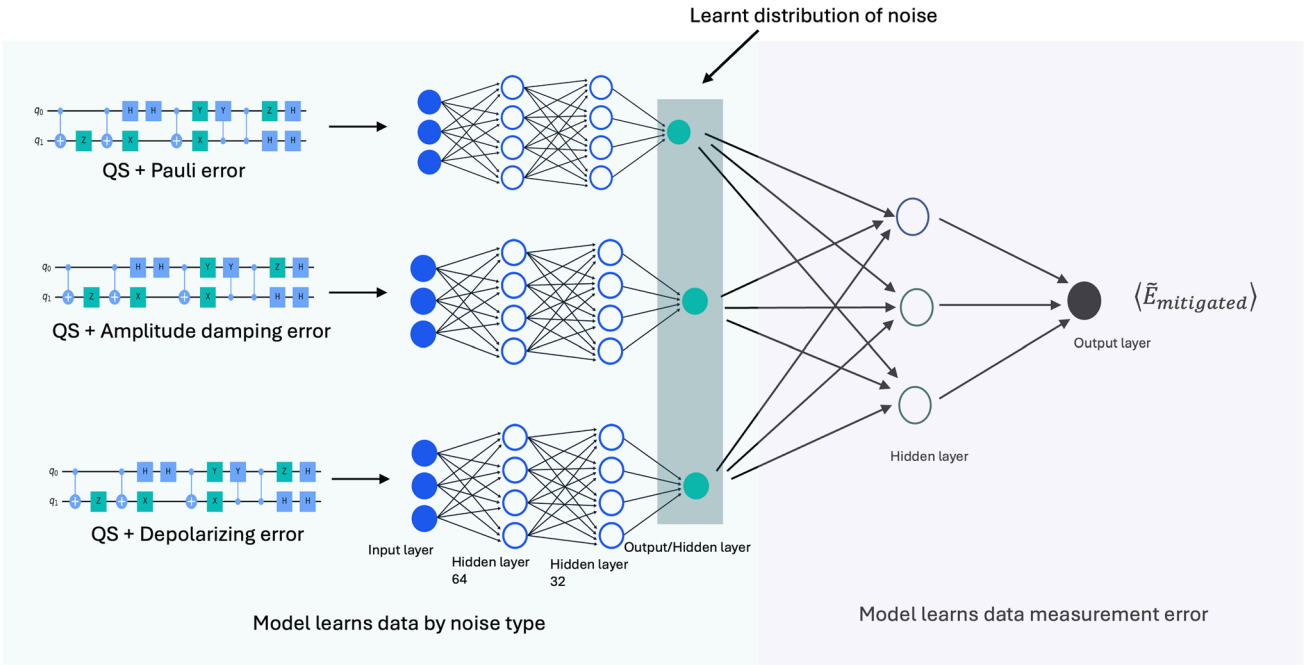

Fig. 3 The architectural diagram of the adaptive neural network. The classifier module includes model that learns data by noise type, and the regression module is the model that learns data measurement error. The architecture depicts an adaptive neural network where both the classifier module and regression module are both neural networks

amplitudes or entangling qubits. A quantum circuit is a step-by-step series of gate operations conducted over time. It starts with the preparation of qubit states, initializing their initial conditions. Subsequently, gate operations within the circuit dynamically modify these states transforming them to the desired configurations. Eventually, the circuit is measured using a measurement gate, capturing the final quantum states for observation or computation (Chatterjee et al. 2023). A quantum gate can be represented by a unitary operator acting on the state vector of a quantum system, denoted as $|\psi\rangle$ (Fig. 3).

The action of a quantum gate U on a quantum state $|\psi\rangle$ can be expressed as Eq. (1).

$$|\psi\rangle_{out} = U|\psi\rangle_{in} \tag{1}$$

where $|\psi\rangle_{out}$ represents the resulting quantum state after the gate operation and $|\psi\rangle_{in}$ represents the initial quantum state before the operation.

In quantum computing, noise models such as Pauli errors, amplitude damping errors, and depolarizing errors commonly encountered can significantly impact the performance and reliability of quantum algorithms. Interestingly, these noise models can be implemented and simulated using quantum gates, providing a direct link between the theoretical understanding of noise and practical quantum computation. Some common quantum gates used to create quantum circuit and implement noise models in this work are as follows:

Hadamard gate (H): Puts a qubit into a superposition of states, making it equally likely to be measured as 0 or 1. In a matrix form, the Hadamard gate is represented as Eq. (2) (Steane Feb. 1998; Spector 2008).

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

Pauli gates (X, Y, Z): Similar to classical NOT gate (X), but with additional operations in the Y and Z directions in the Bloch sphere. Equation (3) shows the matrix representation of the identity, Z, Y, and X gates, respectively (Steane Feb. 1998).

$$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \tag{3}$$

CNOT gate (controlled-NOT): Performs an operation on a target qubit based on the state of a control qubit, enabling entanglement between two qubits. Equation (4) is the matrix representation of the CNOT gate.

$$CNOT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{4}$$

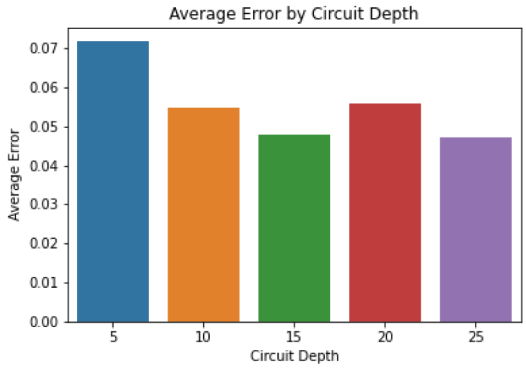

Fig. 4 Average scaled error varied with the circuit depth k

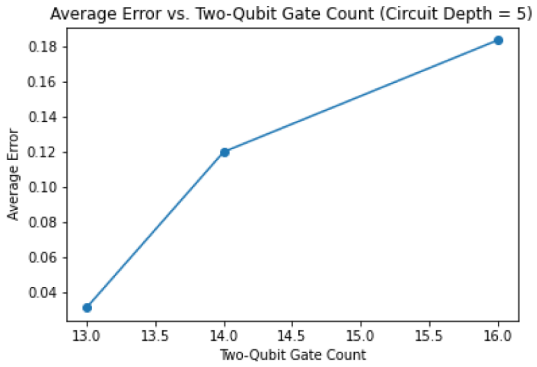

Fig. 5 Average error varied with two qubit gate count for circuit depth = 5

3.2 B. Quantum error models

There are a number of errors and scenarios that induces noise in quantum systems; this research implements three error models: Pauli error, depolarizing error, amplitude damping error, and phase damping error. We considered these error models because they identify as ideal models of errors that capture many of the most important features of the noise occurring in quantum mechanical systems (Figs. 4, 5, 6, and 7).

Pauli errors are characterized by the action of Pauli matrices on quantum states and gates. It encompasses bit flip (X), phase flip (Z), and the combination of both (Y) errors called bit-phase flip.

The Pauli channel error can be represented as shown in Eq. (5).

$$\mathcal{P}(\rho) = (1-p)\rho + \frac{p}{3}\sum_{i=1}^{3} X_i \rho X_i \tag{5}$$

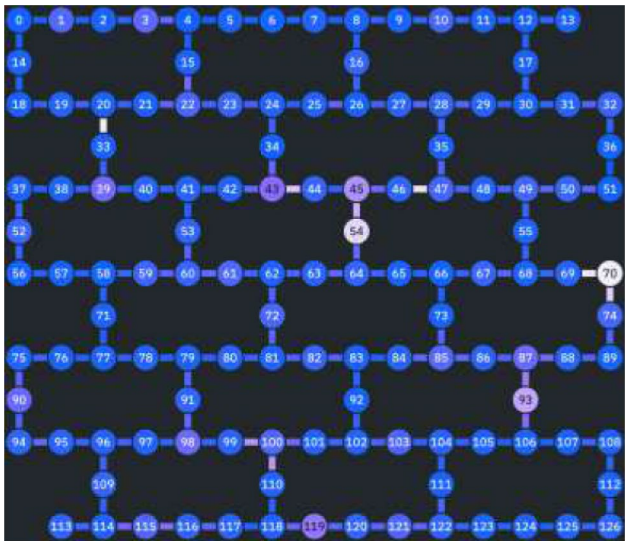

Fig. 6 Layout of 127-qubit IBM Brisbane quantum computer used for the simulation

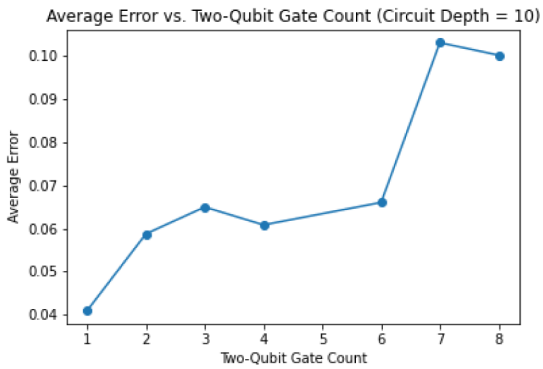

Fig. 7 Average error in expected value varies with two qubit gate count for circuit depth = 10

where $1-p$ represents the probability of an error occurring and X_i denotes the $i-$ th Pauli matrix (with $i = 1 for X, i = 2$ for Y, and $i = 3$ for Z) (Nielsen and Chuang 2011).

Bit-flip error occurs when a qubit's state flips between the computational basis states (0 and 1). For example, a qubit in state $|0\rangle$ might accidentally change to state $|1\rangle$, and it occurs with a probability $1-p$. The bit-flip channel has the operation elements as shown in Eq. (6):

$$E_1 = \sqrt{1-p}X = \sqrt{1-p}\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$
$$E_o = \sqrt{p}I = \sqrt{p}\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{6}$$

Equation (7) gives the operation elements for the phase flip error which describes a phase change flipping the sign of the state.

$$E_1 = \sqrt{1-p}Z = \sqrt{1-p}\begin{bmatrix} 0 & 0 \\ 0 & -1 \end{bmatrix}, \quad E_o = \sqrt{p}I = \sqrt{p}\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{7}$$

The bit-phase flip error has the operation elements given in Eq. (8).

$$E_1 = \sqrt{1-p}Y = \sqrt{1-p}\begin{bmatrix} 0 & i \\ i & 0 \end{bmatrix} \quad E_o = \sqrt{p}I = \sqrt{p}\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{8}$$

Depolarizing errors cause a qubit to deviate from its intended state, typically by randomly altering its state in an unpredictable manner. This can be considered a generalization of bit-flip and phase flip errors. Depolarizing channel $\mathcal{D}_p$ introduces stochastic noise, leading to the degradation of quantum information. This error acts independently on each qubit, replacing the quantum state with a completely mixed state $\frac{I}{2}$ with probability $1-p$. Equation (9) shows the state of the system after the depolarizing noise.

$$\mathcal{D}_p = \frac{p}{2}I + (1-p)\rho, \tag{9}$$

The depolarizing channel can often be parametrized in different ways given as Eq. (10):

$$\mathcal{D}_p = (1-p)\rho + \frac{p}{3}\begin{pmatrix} X\rho X + avb \\ Y\rho Y, ZVZ \end{pmatrix} \tag{10}$$

State ρ is left alone with probability $1-p$, and the operators X, Y, and Z applied each with probability $p/3$ (Nielsen and Chuang n.d.; Jagadish and Petruccione 2018).

Amplitude damping error occurs when a qubit transitions from an excited state to a ground state due to energy dissipation to the environment. This interaction with the environment causes decay in the amplitudes of the quantum states. Trancing over the environment gives the quantum operation described in Eqs. (11) and (12) (Nielsen and Chuang n.d.; Jagadish and Petruccione 2018).

$$\varepsilon_{AD}(\rho) = E_o\rho E_0^\dagger + E_1\rho E_1^\dagger \tag{11}$$

$$E_o = \begin{bmatrix} 1 & 0 \\ 0 & \sqrt{1-\gamma} \end{bmatrix}, E_1 = \begin{bmatrix} 0 & \sqrt{\gamma} \\ 0 & 0 \end{bmatrix} \tag{12}$$

The operation elements for amplitude damping $\gamma = sin^2\theta$ can be thought of as the probability of losing a photon.

3.3 C. The adaptive mechanism (AM) for the proposed approach

The adaptive learning mechanism (Miao and Li 1992; Park et al. 1991) in this research forms a pivotal component, aiming to dynamically adjust/update a neural network parameter based on identified errors. The adaptive learning mechanism operates as a recalibrator within the quantum error mitigation system. It is designed to continuously refine the neural network's ability to recognize and mitigate errors specific to various quantum circuits. The mechanism leverages the results of previous classifier modules on simulated quantum circuits identifying errors, enabling the preceding neural network regression module to adapt its parameters and responses to different error characteristics. This updated rule ensures that the neural network learns and adjusts its parameters to minimize the discrepancy between predicted and target outcomes, enhancing its error mitigation capabilities.

The classifier module is trained on input quantum circuit data X and the corresponding error labels y. The classifier module is represented in Eq. (13) using a supervised learning framework, where we seek to learn a mapping $f_c(X)$ that predicts the error label $\hat{y}$ given the input data X (Kotsiantis 2007).

$$f_c(X) = softmax(XW_c + b_c) \tag{13}$$

W_c and b_c represent the weights and biases of the parametrized function $f_c(X)$.

After supervised machine learning identifies the type and characteristics of errors in a quantum circuit simulation, the predictions from the classifier module, along with the original input data, are then passed to the regression module, which is then employed to adjust the output probabilities or expectation value of the quantum circuit of interest, compensating for the identified errors. The regression module predicts $\hat{\varepsilon}$ based on X and $\hat{y}$.

Let $f_r(X, \hat{y})$ represents the output of the regression module, where f_r is a function parameterized by weights W_r and biases b_r. Mathematically, the regression module can be represented as Eq. (14) (Doan and Kalita 2015).

$$f_r(X, \hat{y}) = XW_r + b_r \tag{14}$$

The adaptive learning mechanism operates iteratively, refining the neural network's error mitigation capabilities over multiple cycles of error identification, neural network training, and adjustment. This iterative refinement process ensures that the neural network becomes increasingly adept at handling a wide range of errors, contributing to the overall improvement of quantum circuit fidelity. Both the classifier module and the regression module form the adaptive neural network for quantum error mitigation (ANN-QEM) designed for adaptive mitigation in this study.

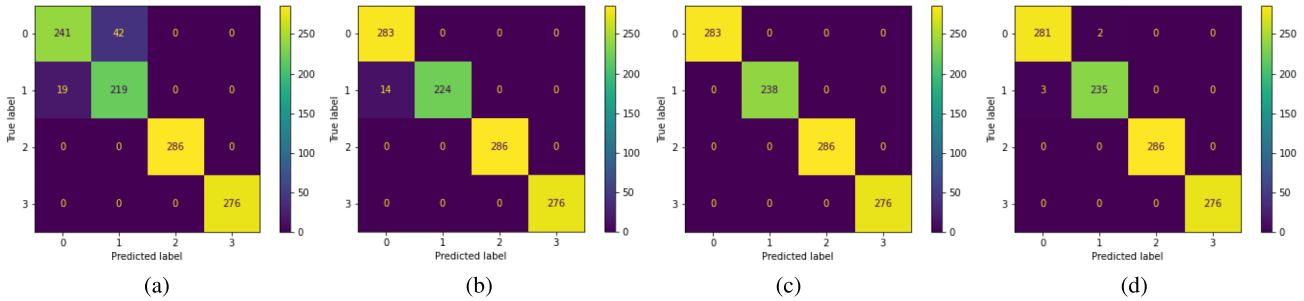

Fig. 8 Result for the classifier module classifying 1081 quantum circuit data based on amplitude damping error, depolarizing error, no error, Pauli error, and as 0, 1, 2, 3, respectively. **a** The confusion matrix from logistic regression classifier, **b** the confusion matrix from support vector classifier, **c** the confusion matrix from the random forest classifier, and **d** confusion matrix from the neural network classifier

Performance metrics, such as the fidelity improvement and error mitigation accuracy, are employed to validate the efficacy of the adaptive learning mechanism. These metrics quantify the neural network's ability to adapt to different error characteristics and improve the fidelity of quantum circuit outputs.

4 Methodology

This section provides a detailed, step-by-step explanation of the workflow depicted in Fig. 2. The methodology outlines the integration of quantum circuit design, execution, and machine learning–based quantum error mitigation.

4.1 C. Quantum circuit design, simulation, compilation, and execution

This section describes part a(i), a(ii), b, and c in the research workflow as shown in Fig. 2, including the quantum circuit design, noiseless and noisy simulations, compilation for hardware execution, and final quantum computation.

The workflow begins with the systematic design of quantum circuits, incorporating diverse configurations of quantum gates. To ensure the uniqueness of this project and the dataset used, the dataset for the analysis is simulated entirely from scratch. The quantum circuit configuration selection process entails the careful construction of quantum circuits comprising a series of quantum gates and operations to embody valid quantum operations. The circuits are constructed with single-qubit gates: , Rx,Ry,Rz,X,Y, Z used to manipulate individual qubits. And the two-qubit gate CX used for entanglement generation and multi-qubit operations. Each circuit represents a quantum operation that evolves the state $|\psi_{final}\rangle$ to$U|\psi_{initial}\rangle$. U is the unitary matrix corresponding to the sequence of gate applied. The design process considers various configurations by systematically setting the number of qubits $q = 2$ qubit with circuit depths ranging from$K = 1, \ldots, 30$.

Simulations to generate dataset were implemented in Python 3 using essential libraries such as Qiskit (Javadi-Abhari et al. 2024) and scikit-learn. Qiskit is IBM's open-source toolkit for quantum computing. The configuration involved setting up Qiskit and related dependencies, ensuring compatibility with available backends (quantum simulators and real quantum devices) for circuit execution. To establish a baseline for error mitigation, noiseless simulations $\langle\widehat{\varepsilon}_k\rangle$ were also performed using IBM quantum simulators. These simulations captured ideal, error-free circuit behaviors. The noisy quantum circuit data, in contrast, were generated using the 127-qubit IBM Brisbane system, which incorporates a realistic noise model. The layout is shown in Fig. 4. IBM's cloud quantum services (IBM-Q) provided access to the hardware, enabling robust experimentation and simulation (Quantum 2021). The noise models employed in this study are based on IBM's realistic noise framework, which replicates the key sources of decoherence and gate infidelity present in IBM quantum processors. This shared framework supports the assumption that models trained on one processor can generalize to others within the IBM network (Fig. 8).

The compilation process [b] in the workflow for real hardware involved two critical steps: translation, where logical circuits were converted into hardware-compatible instructions, and optimization, where circuit gates were fine-tuned to minimize execution errors. Experiments were conducted using Google Colaboratory, leveraging the computational capabilities of NVIDIA Tesla T4 GPUs for building quantum circuits. A comprehensive dataset was constructed, encompassing $n = 5600$ circuits that reflected various noise scenarios (e.g., depolarizing noise, amplitude damping, and Pauli errors). These circuits provided a detailed representation of quantum state behavior under realistic noise conditions.

To ensure robust and representative data, quantum circuit configurations were selected carefully. Variations in gate sequences, qubit arrangements, and circuit depths were systematically introduced to produce a diverse array of quantum circuits. The combination of these configurations ensured comprehensive coverage, generating datasets for both noisy and noiseless circuits. This process established a unique foundation for training and evaluating the proposed adaptive quantum error mitigation approach.

4.2 D. Quantum data description

This section describes part [d] of the architecture of the research workflow as shown in Fig. 2, focusing on the generation and processing of quantum datasets for error analysis and mitigation.

The datasets include 5600 noisy random quantum circuit data, capturing the three error scenarios considered in this research which are Pauli error, depolarizing error, and amplitude damping error which explains the complex environmental interactions of quantum states. The richness of the datasets lies in their ability to represent the real-world noise challenge faced by quantum computers, providing a comprehensive training ground for the classifiers module to identify quantum errors in quantum states and regression module to mitigate error in expectation values.

The data primarily consists of quantum circuit configurations and their corresponding simulation results. Each circuit evolves the state $|\psi_x\rangle$ based on the applied gate sequence U_x where

$$|\psi_x\rangle = U_x|\psi_{initial}\rangle \tag{15}$$

Here, U_x is the unitary operator representing the specific gate sequence.

The feature generated for each quantum circuit $|\psi_x\rangle$ includes the mean given by Eq. (16):

$$\mu_x = \frac{1}{n}\sum_{i=1}^{n} m_i \tag{16}$$

And variance

$$\sigma_x^2 = \frac{1}{n}\sum_{i=1}^{n} (m_i - \mu_x)^2 \tag{17}$$

where m_i represents an individual measurement outcome and n is the total number of measurements. ρ_x = noise models, k = circuit depth, ϵ_x = error deduction, ϕ = probability amplitudes, $\langle\epsilon_x\rangle$ = noisy expectation value, and $\langle\widehat{\varepsilon}_k\rangle$ = true expectation value. The error vector is calculated as $\epsilon_x = \widehat{\varepsilon}_k - \varepsilon_k$. The expectation value of an observable $\widehat{\varepsilon}$ in quantum mechanics is given by the formula in Eq. (18):

$$\langle\widehat{\varepsilon}_k\rangle = \langle\psi|\widehat{\varepsilon}_k|\psi\rangle \tag{18}$$

4.3 E. Data preprocessing

The data preprocessing steps include three processes: data cleaning, feature extraction/engineering, encoding and normalization.

At initial step, all null values were filled with 0 with redundant features dropped. The data was also balanced by resampling low classes. Features were extracted from the simulated dataset. The extracted measurement was combined (probabilistic features, gate sequence representation, error labels) for both noiseless and noisy circuits into a structured dataset. The dataset is complex; the complexity arises from the complexity of quantum system data and the need to simulate a four error conditions, capturing the nuanced interplay of errors in quantum. For the gate sequence, each gate type was extracted and represented as separate feature. The original output of the counts which is the measurement of the quantum state was also transformed to separate features. Each of these separate features represents the number of final quantum state observed after measurement. The project uses one-hot encoding to represent each gate type present in the circuit, a binary label indicating whether the circuit had injected errors or error-free was used for the noise model column for training the classifier module of the adaptive NN. Normalization was employed to bring all values within a standardized range, typically between 0 and 1. This helps ensures that the model is not influenced by varying feature value scales, promoting convergence and stability during training. The dataset was then split to training and testing set using 20% of the dataset as the training set.

4.4 F. Machine learning models for classifier module

This section explains part [e] of the workflow shown in Fig. 2, detailing the machine learning models used to classify quantum error types. Quantum states in error-prone environments exhibit nuanced features, demanding machine learning models that can discern subtle patterns and adapt to evolving error landscape. Consideration was given to machine learning algorithms capable of accommodating high-dimensional data and adapting to the dynamic nature of error-prone quantum systems. Ensemble learning using random forest, support vector classifier (SVC), logistic regression, and neural network are the models considered for the classifier module in the experiment. The machine learning training process of the classifier module also implements boosting and cross-validation strategies to ensure robustness and generalizability of the models across different error scenarios. Bagging, also known as Bootstrap Aggregating, was used. This involves

training multiple models on different subsets of the dataset to improve overall performance. Bagging is employed here by repeatedly training the random forest as a base model on different subsets of the dataset (sample data) obtained by random sampling with replacement. This approach helps in reducing variance and overfitting by introducing diversity in the training data for each model. The final ensemble prediction is made by aggregating predictions from individual models using a simple averaging method. The architecture of the neural network used for the classifier module has its first layer as a fully connected layer consisting of 64 neurons followed by a hidden layer of 32 neurons; this layer connects to that final layer predicting the classes of each quantum data represents; cross-entropy loss also known as log loss is used.

4.5 G. Neural network for regression module

The regression module, representing step [f] in Fig. 2, is tasked with predicting the error mitigated expectation values ($\langle\widehat{\varepsilon}_{mitigated}\rangle$) of noisy quantum states. During training, the true noiseless measurements $\langle\widehat{\varepsilon}_k\rangle$ are used as the ground truth to supervise the learning process. The regression module leverages the noisy measurements, along with features derived from the quantum circuit and the noise classifier module, to learn the relationship between the noisy data and the corrected expectation values. Once trained, the regression module does not rely on the true measurements and instead uses the noisy data and extracted features to predict the mitigated expectation values, enabling error mitigation during inference. The architecture begins with a fully connected layer consisting of 64 neurons and connects to the input layer that combines multiple features such as the mean (μ_x) and variance (σ_x), gate sequences (U_x), noise classifications (ρ_x), and circuit depth (k). The input feature in this layer is a combination of the generated data and the output data from the preceding classifier module identifying the kind of error in each quantum data representation. Each neuron in this layer receives input from all features and performs a linear transformation followed by a rectified linear unit (ReLU) activation function. This layer facilitates the extraction of complex feature representations from the input data. The second fully connected layer comprises 32 neurons and follows the same pattern as first fully connected layer. It further refines the feature representations learned by the previous layer, capturing higher-level abstractions in the data. The final fully connected layer consists of a single neuron, responsible for producing the regression output. It performs a linear transformation of the features learned by the preceding layers without applying any activation function. This layer outputs the predicted expectation value $\langle\widehat{\varepsilon}_{mitigated}\rangle$ of the quantum system.

4.6 H. Training and optimization for ANN-QEM

The training process involved iteratively adjusting the network's parameters to minimize a predefined loss function, first a cross-entropy loss for the classifier module and then a mean squared error (MSE) loss for the regression module facilitated by the backpropagation algorithm. This MSE loss function for the regression module quantified the disparity between predicted and actual expectation values, guiding the optimization process towards minimizing error. This is given by Eq. (19).

$$\frac{1}{n}\sum_{1=1}^{n}\left(\langle\widehat{\varepsilon}_k\rangle - \langle\widehat{\varepsilon}_{mitigated}\rangle\right)^2 \tag{19}$$

To enhance the performance of the adaptive neural network, several optimization strategies are implemented. Firstly, appropriate hyperparameters for both modules were instantiated, such as learning rates $lr = 0.001$, batch sizes = 32, and the number of epochs = 100 and 500 for the classifier and regression modules, respectively. These hyperparameters are carefully selected through experimentation and cross-validation techniques to optimize model convergence and prevent overfitting. Additionally, regularization techniques such as dropout or weight decay were incorporated to mitigate the risk of overfitting. Furthermore, model performance is continuously monitored during training, with validation datasets used to assess generalization capabilities and prevent potential issues such as underfitting or overfitting.

4.7 I. Evaluation metrics

We evaluated the classifier module in terms of various evaluation metrics such as accuracy, precision, recall, and F1 score for each of the dataset considered. These metrics will represent how accurate the predictions are, percentage of quantum states that are correctly classified, and the true positive number.

$$\text{Precision} = \frac{N_{\text{TP}}}{N_{TP} + N_{FP}} \tag{20}$$

$$\text{Recall} = \frac{N_{\text{TP}}}{N_{TP} + N_{FN}} \tag{21}$$

$$\text{Accuracy} = \frac{N_{\text{TP}} + N_{\text{TN}}}{N_{\text{TP}} + N_{\text{TN}} + N_{\text{FP}} + N_{FN}} \tag{22}$$

$$\text{F1 score} = \frac{2 \times \text{precison} \times \text{recall}}{\text{precision} + \text{recall}} \tag{23}$$

where N_{TP} is the number of true positives, i.e., number of images correctly labeled; N_{FP} is the number of false positives

Table 1 Experimental result for classifier module

Classifier module	Avg. F1 score	Avg. precision	Avg. recall	Avg. accuracy
RFC	1.0	1.0	1.0	1.0
NN	0.996	0.995	0.995	0.995
SVM	0.949	0.949	0.9487	0.9487
Log Reg	0.948	0.949	0.949	0.944

which mean the number of images detected but not corresponding to the ground truth; and N_{FN} is the number of true negatives which means number of correct images that could not be identified. Table 1 shows the accuracy of each of the model tested for the classifier module.

5 Experimental results

Four ML models were considered for the classifier module. The classifier module was individually evaluated how it can predict the four classes which include the three different kinds of noise models scenarios considered and a noiseless condition. Subsequently, the system undergoes regression analysis within a neural network to predict the correct expectation value, thereby mitigating errors induced by the identified noise. In evaluating the performance of the various classifiers, namely random forest classifier (RFC), neural network (NN), support vector machine (SVM), and logistic regression (Log Reg), several metrics were employed to assess their effectiveness in classifying data instances. The F1 score, precision, recall, and accuracy serve as crucial indicators in measuring the classifier's predictive capability and overall performance and is presented in Table 1. The RFC exhibited exceptional performance across all metrics, achieving perfect scores of 1.0 for F1 score, precision, recall, and accuracy. This signifies the RFC's robust ability to correctly classify instances across all the four classes with high precision and recall rates, resulting in optimal overall accuracy. The NN exhibited slightly lower but still excellent performance, with an average F1 score, precision, recall, and accuracy of 0.996. Conversely, the SVM and Log Reg models displayed slightly lower but still impressive performance, with F1 scores of 0.949 and 0.948, respectively. The classifier module of the adaptive neural network successfully identified different types of noise in quantum systems with high accuracy. The results suggest that RFC and NN are particularly effective classifiers for this task, with SVM and Log Reg also providing reasonable performance.

The adaptive mechanism (AM) integrated with various classifier modules and a NN regression module presents a comprehensive approach to error mitigation in quantum computing. By leveraging different classifier models in conjunction with the NN regression module, the AM adapts to the diverse error characteristics present in quantum circuits, thereby enhancing accuracy and reducing MSE. The results presented in Table 2 depict the performance of the adaptive neural network in mitigating errors in the expectation value of quantum systems. The adaptive mechanism (AM) employed in the regression module aims to predict the correct expectation value, thereby compensating for errors induced by different noise types identified by the classifier module.

Table 2 Experimental result for adaptive QEM

Adaptive mechanism (AM)	Testing accuracy	Testing MSE
Fully NN AM	99.99	0.00005
RF + NN AM	99.17	0.00354
SVM + NN AM	99.06	0.00401
Log Reg + NN AM	98.27	0.00739
ZNE (traditional QEM method)	88.68	0.01530

The experimental setup involved testing various adaptive mechanisms, including fully neural network (NN) AM, Random Forest (RF) + NN AM, SVM + NN AM, and Log Reg + NN AM, alongside a baseline fully NN AM. Each adaptive mechanism was evaluated based on testing accuracy and MSE on the testing dataset.

The fully NN AM achieved the highest testing accuracy of 99.99% and the lowest MSE of 0.00005, indicating its exceptional performance in accurately predicting the expectation value and effectively mitigating errors induced by noise. In this model, both the classifier module that predicts the kind of error in the system and the regression module that mitigates the expected value error are neural network architecture. This result underscores the effectiveness of utilizing a fully neural network approach without relying on additional classifiers for noise identification.

The RF + NN AM and SVM + NN AM achieved slightly lower testing accuracies of 99.17% and 99.06%, respectively, along with higher MSE values of 0.00354 and 0.00401. While these adaptive mechanisms still demonstrate impressive performance, the addition of classifiers (RF and SVM) in conjunction with the neural network introduces some computational overhead and may slightly degrade performance compared to the fully NN AM.

Similarly, the Log Reg + NN AM exhibited a testing accuracy of 98.27% and a higher MSE of 0.00739, indicating a slightly lower performance compared to the other adaptive

mechanisms. The reliance on logistic regression as a classifier may have contributed to the reduced accuracy and increased MSE compared to the fully NN AM.

To provide a more detailed analysis of the adaptive mechanisms (AM) employed in this study, we include the number of trainable parameters for each configuration. The fully neural network adaptive mechanism (fully NN AM), which integrates neural networks for both the classifier and regression modules, consists of 3265 trainable parameters. In comparison, the hybrid models—RF + NN AM, SVM + NN AM, and Log Reg + NN AM—have slightly fewer trainable parameters, totaling 3201. This difference arises because the hybrid models utilize traditional machine learning classifiers, which do not contribute to the trainable parameter count, alongside the neural network regression module.

In conclusion, the experimental results underscore the significance of the adaptive neural network approach in mitigating errors in quantum systems. The fully NN AM emerges as the most effective adaptive mechanism, offering superior accuracy and MSE performance compared to hybrid approaches involving classifiers.

In this study, the ML models were trained and evaluated on data generated from the IBM Brisbane quantum processor. The dataset included a variety of simulated noise models that mimic noise characteristics common to IBM quantum processors, which were intended to support generalization across different IBM hardware. Although direct testing on data from other IBM processors (e.g., IBM Melbourne or IBM Lagos) was not conducted in this research, the inclusion of diverse noise models was aimed at ensuring that the models are not overly dependent on the specific noise profile of IBM Brisbane.

We compared results with zero noise extrapolation (Temme et al. Nov. 2017) implemented on Mitiq ("Mitiq 0.34.0 documentation". n.d.) which facilitates noise mitigation on IBMQ backends using the default settings for noise scaling and extrapolation. MSE is used to evaluate the performance of zero noise extrapolation (ZNE) by comparing the mitigated expectation values obtained using ZNE with the true expectation values in noiseless simulations. First, we performed noiseless simulations of the quantum circuit for each k, U_x, and ρ_x to obtain $\langle \widehat{\varepsilon}_k \rangle$. These expectation values serve as the ground truth against which the ZNE mitigated values will be compared. Then, we computed the mean squared error between the ZNE mitigated expectation values and the true expectation values obtained from noiseless simulations.

In comparison, the ANN-QEM demonstrates a far lower overhead over four different types of noise and random circuit combinations than the digital noise extrapolation. Quantum processing time (QPU) to generate data plus mitigation time for ANN-QEM was about 8 h, 20 min for 5000 circuits, 8 h for the QPU processing time, and 20 min for the ANN-QEM to mitigate the error. For ZNE, mitigation for about 1800 circuits took over 12 h. In effect, ANN-QEM reduces the runtime better than traditional QEM methods. In all the machine learning experiments perform in this research, the accuracy for ML models was better, producing lesser MSE values. This indicates a significant improvement in efficiency over digital ZNE.

6 Future work

The current research lays the groundwork for advancing error mitigation techniques in quantum computing, yet several avenues remain unexplored for future investigation. One promising direction involves further refinement and optimization of the ANN-QEM integrated with different classifier and regression modules. Fine-tuning the ANN-QEM architecture and exploring alternative combinations of classifier models and regression techniques could potentially enhance error mitigation efficacy and improve overall quantum circuit fidelity.

Additionally, extending the scope of the study to encompass a broader range, larger quantum systems, and noise models would provide valuable insights into the generalizability and robustness of the proposed ANN-QEM approach. As the methodology is scaled to circuits with more qubits, adjustments will be necessary to manage the exponential increase in data dimensionality and computational requirements. For instance, incorporating sparsity-aware neural networks or leveraging distributed training techniques could enable efficient handling of high-dimensional quantum datasets. In this context, assessing the performance of the ANN-QEM on quantum circuits with increasing qubit counts and complexity would provide valuable insights into its scalability and applicability to future quantum computing architectures.

Investigating the ANN-QEM performance across various quantum computing platforms and experimental setups could offer comprehensive validation of its efficacy in real-world applications. This includes tailoring the approach to account for hardware-specific noise models and constraints. This will also involve validating the ML models on data from multiple IBM quantum processors to empirically confirm their generalization capabilities. This includes training on one processor and testing on others to account for processor-specific noise characteristics and calibration routines. Furthermore, understanding the computational overhead and scalability challenges becomes critical when addressing systems with higher complexity. The parameter count in this research provides valuable insights into the model's complexity. For example, the fully neural network adaptive mechanism (3265 parameters) contrasts with hybrid models such as RF + NN AM, SVM + NN AM, and Log Reg + NN

AM (3201 parameters), highlighting the trade-offs in computational efficiency and model capacity.

A roadmap for future research focused on adapting ANN-QEM to larger quantum systems includes optimizing the methodology for circuits with higher qubit counts, increasing depth, and leveraging advanced techniques to manage computational complexity. Incorporating these strategies into future iterations of ANN-QEM could significantly improve its utility and scalability, paving the way for more robust and practical quantum error mitigation techniques in diverse quantum computing environments.

7 Conclusion

In conclusion, this research presents a novel adaptive neural network model for quantum error mitigation (ANN-QEM), demonstrating its efficacy in removing error from quantum circuit data measurement across diverse noise models and circuit configurations. Through the integration of supervised machine learning techniques and adaptive learning mechanisms, the ANN-QEM dynamically adjusts output probabilities and expectation values based on identified error characteristics, thereby learning errors and mitigating them in real time with high accuracy. Experimental validation on simulated quantum circuits on and empirical testing on the IBM quantum computing platform showcase the versatility and scalability of the proposed approach. By leveraging machine learning and neural network methodologies, this research contributes to the ongoing efforts in overcoming the challenges posed by noise in quantum computing, paving the way for more reliable and efficient quantum computations in practical applications.

Author contribution Adeniyi conducted the experiments and prepared the manuscript. Kumar provided the design feedback and updated the manuscript.

Data availability No datasets were generated or analyzed during the current study.

Declarations

Competing interests The authors declare no competing interests.

References

Acampora G, Vitiello A (2021) Error mitigation in quantum measurement through fuzzy C-means clustering. In: 2021 IEEE international conference on fuzzy systems (FUZZ-IEEE), pp 1–6. https://doi.org/10.1109/FUZZ45933.2021.9494538

Acampora G, Grossi M, Vitiello A (2021) Genetic algorithms for error mitigation in quantum measurement. In: 2021 IEEE congress on evolutionary computation (CEC), pp 1826–1832. https://doi.org/10.1109/CEC45853.2021.9504796

Brun TA (2020) Quantum error correction. Oxford Research Encyclopedia of Physics. https://doi.org/10.1093/acrefore/9780190871994.001.0001/acrefore-9780190871994-e-35

Cai Z, Babbush R, Benjamin SC, Endo S, Huggins WJ, Li Y, McClean JR, O'Brien TE (2023) Quantum error mitigation. Rev Mod Phys 95(4):045005. https://doi.org/10.1103/RevModPhys.95.045005

Chatterjee A, Phalak K, Ghosh S (2023) Quantum error correction for dummies. In: Proceedings - 2023 IEEE international conference on quantum computing and engineering (QCE), pp 70–81

Chiaverini J et al (2004) Realization of quantum error correction. Nature 432:7017

Czarnik P, Arrasmith A, Coles PJ, Cincio L (2021) Error mitigation with Clifford quantum-circuit data. Quantum 5:592

Doan T, Kalita J (2015) Selecting machine learning algorithms using regression models. In: 2015 IEEE international conference on data mining workshop (ICDMW), pp 1498–1505. https://doi.org/10.1109/ICDMW.2015.43

IBM Quantum (2021) IBM Quantum platform. Retrieved from https://quantum.ibm.com/

Jagadish V, Petruccione F (2018) An invitation to quantum channels. Quanta 7:1

Javadi-Abhari A, Treinish M, Krsulich K, Wood CJ, Lishman J, Gacon J, Martiel S, Nation PD, Bishop LS, Cross AW, Johnson BR, Gambetta JM (2024) Quantum computing with Qiskit. arXiv. https://doi.org/10.48550/arXiv.2405.08810

Kim C, Park KD, Rhee J-K (2020) Quantum error mitigation with artificial neural network. IEEE Access 8:188853–188860

Kim J, Oh B, Chong Y, Hwang E, Park DK (2022) Quantum readout error mitigation via deep learning. New J Phys 24(7):073009

Kotsiantis SB (2007) Supervised machine learning: a review of classification techniques. In: Emerging artificial intelligence applications in computer engineering: real world AI systems with applications in eHealth, HCI, information retrieval and pervasive technologies, pp 3–24. IOS Press. https://doi.org/10.5555/1566770.1566773

H Liao, DS Wang, I Sitdikov, C Salcedo, A Seif, and ZK Minev (2023) "Machine learning for practical quantum error mitigation." arXiv, Sep. 29, 2023

Miao YF, Li ZM (1992) Adaptive prediction using neural networks. In: Proceedings of the 1992 IEEE international symposium on circuits and systems, pp 340–343. https://doi.org/10.1109/ISCAS.1992.229944

Mitiq 0.34.0 documentation (n.d.). Available at: https://mitiq.readthedocs.io/. Accessed 20 Jan 2024

Nielsen MA, Chuang IL (2011) Quantum computation and quantum information: 10th anniversary edition. Cambridge University Press

Park DC, El-Sharkawi MA, Marks RJ (1991) An adaptively trained neural network. IEEE Trans Neural Netw 2(3):334–345

Pokharel B, Lidar DA (2023) Demonstration of algorithmic quantum speedup. Phys Rev Lett 130(21):210602

Preskill J (2018) Quantum computing in the NISQ era and beyond. Quantum 2:79

Reed MD (2013) Entanglement and quantum error correction with superconducting qubits. Lulu.com
Schindler P et al (2011) Experimental repetitive quantum error correction. Science 332(6033):1059–1061
Schumacher B (1995) Quantum coding. Phys Rev A 51(4):2738–2747
Seif A, Landsman KA, Linke NM, Figgatt C, Monroe C, Hafezi M (2018) Machine learning assisted readout of trapped-ion qubits. J Phys B at Mol Opt Phys 51(17):174006
L Spector (2008) "Quantum computing," in Proceedings of the 10th annual conference companion on Genetic and evolutionary computation, Atlanta GA USA: ACM, 2865–2894
Steane A (1998) Quantum computing. Rep Prog Phys 61(2):117–173
A. Steane (2003) "Quantum computing and error correction," ArXiv Quantum Phys., Apr. 2003
Temme K, Bravyi S, Gambetta JM (2017) Error mitigation for short-depth quantum circuits. Phys Rev Lett 119(18):180509
Wittek P (2014) Introduction. In: Quantum machine learning. Academic Press, pp 3–10. https://doi.org/10.1016/B978-0-12-800953-6.00001-3