5-2024

Neural Network Characterization Methods for Noisy Quantum Hardware Results

Daniel Richard Breslawski
drb3015@rit.edu

Neural Network Characterization Methods for Noisy Quantum Hardware Results

Daniel Richard Breslawski

Neural Network Characterization Methods for Noisy Quantum Hardware Results

Daniel Richard Breslawski

May 2024

A Thesis Submitted
in Partial Fulfillment
of the Requirements for the Degree of
Master of Science
in
Computer Engineering

RIT | Kate Gleason College of Engineering

Department of Computer Engineering

Neural Network Characterization Methods for Noisy Quantum Hardware Results

Daniel Richard Breslawski

Committee Approval:

Sonia Lopez Alarcon *Advisor* Date
Department of Computer Engineering

Andres Kwasinski Date
Department of Computer Engineering

Dongfang Liu Date
Department of Computer Engineering

Acknowledgments

I would like to thank my family for all their support and encouragement on the wild and winding journey that brought me here. Thank you to the Breslawskis, Memelos, Chutes, Endicotts, Dixons, and Caseys.

I would like to thank all the individuals at RIT and MCC that made my time there special. It was a difficult but worthy path to walk, and we got there together.

I would especially like to thank all my professors at both schools for their invaluable advice, support, and care. A special thank you to my advisor Dr. Sonia López Alarcón for her input and support during my unconventional go at post-secondary education, and for her understanding and care during times of personal duress.

I would also like to extend a special thank-you to my colleagues in the Quantum Computing Research Group. Thank you Harrison, Brody, Tony, Aden, Rachel, Andy, and all the others.

Thank you to my friends in the Rochester cycling community, to the Genesee Valley Pathfinders, and to the Rochester rock climbing community. You taught me the value and meaning of community and always allow me a space to push my limits.

Thank you to my coworkers at AMD and D3 for your support during the final push to finish this work. Having so many of you deeply engage with this work and attend my defense meant more than I can say.

Dedicated to Willow, the perfect companion; to Branta canadensis; and to everyone I have been lucky enough to love in this life.

Abstract

As the nascent quantum computing paradigm matures and quantum devices become widely available, discrete "eras" are emerging which characterize evolving quantum computing technologies, much like semiconductor technology node classifications. The current period has been coined the Noisy Intermediate-Scale Quantum (NISQ) era [1], as devices above 50 qubits in size exist but computations using these devices accumulate error quickly from classical interference sources and state decoherence. NISQ machines may surpass the capabilities of modern classical computers in ideal circumstances, but the accumulation of error from physical noise limits the size and implementability of reliable quantum algorithms [2]. Because of these limitations, strategies are under development that can improve the results of computations on NISQ devices or identify characteristics of the accurate solution space that might be preserved in the noisy data. These are known as *error mitigation* strategies [3] [4].

One such method that has shown promise is the use of classical machine learning to extract information about the pre-measurement output of a NISQ device. This work proposes a new use of machine learning to identify the accurate solutions of basis-encoded quantum algorithms in the presence of noise. Methods of encoding the probabilistic solution space of a basis-encoded quantum algorithm are researched to identify the characteristics that represent good ML training inputs. A multilayer preceptron artificial neural network (MLP ANN) was trained on the results of 8-state and 16-state basis-encoded quantum algorithms both in the presence of noise and in noise-free simulation. It is demonstrated using simulated quantum hardware and probabilistic noise models that a sufficiently trained model may identify accurate solutions to quantum applications with over 90% precision and 80% recall on select data. The model makes confident predictions even with enough noise that the solutions cannot be determined by direct observation, and when it cannot, it can identify the inconclusive experiments as candidates for other error mitigation techniques.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

This work is motivated by the need to extract useful information from the output of quantum computational devices in realistic conditions, i.e. in the presence of noise. As devices in the Noisy Intermediate-Scale (NISQ) era continue to scale, new quantum computers are becoming available which feature increasing numbers of qubits. The latest era of superconducting qubit circuits includes Google's 53-qubit *Sycamore* [12], the Chinese 66-qubit *Zuchongzhi* [13], and IBM's 127-qubit *Eagle* and 433-qubit *Osprey* [14]. Quantum computers with more qubits are capable of executing more complex algorithms, including some which have "superclassical" speedup scaling. For some of these problems, 50+ qubit NISQ-era devices have demonstrated *quantum advantage*: the ability to solve a problem that no classical computer can solve in a reasonable amount of time [12] [13]. Demonstrable quantum advantage proves that quantum computers are a useful subject of research.

As the name suggests, a key characteristic of NISQ-era devices is the presence of error-inducing noise. In a superconducting quantum circuit, noise can be introduced through thermal quantum state decoherence (relaxation), interference from outside energy sources, and qubit link decoupling. This noise can alter the system state during the experiment, changing the measured result and causing error in the results

of an experiment which uses multiple repeated trials of a quantum circuit to encode its answer in the resulting probability distribution. In basis-encoded algorithms, noise can cause the result of the experiment to be inconclusive–no output state is measured frequently enough to be confidently identified as the answer to the problem [15]. This work aims to mitigate the issue of noise in NISQ devices by identifying characteristics of the probability distribution of basis-encoded quantum algorithms that are favorable to classical machine learning techniques with the goal of using these techniques to accurately classify the solution space of a problem.

1.2 Objectives

The primary objective of this work is to demonstrate a novel approach for the identification and characterization of solutions of quantum computing problems, employing classical artificial neural networks (ANNs). Simple ANNs with a single hidden layer are known to be capable of discrete categorization tasks [16]. Basis-encoded quantum algorithms have a discrete answer or answers in a finite solution space, encoded as the most frequently observed states in the probability distribution of all observed final system states. This makes the results of basis-encoded algorithms a good candidate as a *multi-label classification problem* for a neural network. This work aims to use data generated from the results of quantum computer experiments as training, validation, and test input for a feed-forward multi-label classification ANN. The ANN is trained using this data and a supervised learning approach to identify the correct solutions to quantum problems, encoded as discrete category labels.

As discussed in the previous section, confounding factors exist which may make it difficult to identify the solutions to a basis-encoded problem. The primary impediment is the presence of noise, which distorts the distribution and makes it less likely to observe the answer states, in some cases making the answer inconclusive. Another impediment is that a quantum algorithm may have more than one answer. Each ad-

ditional answer to a basis-encoded problem reduces the probability of observing each other answer state which amplifies the negative effects of noise on the observability of the answer states.

Since ANNs may provide a statistical confidence for their predictions, a well-calibrated model may be used as part of a heterogeneous quantum-classical error mitigation strategy. Given an arbitrary probability distribution from a basis-encoded problem, the model's predictions and its confidences may be applied back to the input distribution to amplify predicted solutions and diminish other states.

Ultimately, this work provides a pathway for quantum computing researchers to reach more useful conclusions about the results of their real-world experiments. Since ANNs are a robust and mature area of existing research with a great deal of hardware and software implementation support, they are straightforward to apply as a characterization method for QC results. Adopting this approach will allow precious NISQ-era QC hardware resources to be reallocated away from error mitigation or correction and towards more complex algorithms, and future researchers in the NISQ era may build upon this work by applying other ML strategies towards the characterization of noisy quantum computer output.

1.3 Contribution

- A data set was created from the output of simulated quantum systems for a subset of basis-encoded problems. This data set consists of approximately 230,000 experiments solving different problems that leverage the quadratic speedup of Grover's Algorithm for unstructured database search [17]. The data set contains experiments from both ideal (noise-free) simulations and from simulations introducing various types and amounts of noise. This data set is the first-known of its type, aggregating results from different complex quantum circuits that solve real problems with QA and also include noise of different sources and types.

- The results of each experiment in this data set were encoded as discrete probability distributions and the solutions to each problem were supplied as multi-hot encoded multi-label classification categories against which machine learning models can train. Characterizing basis-encoded quantum problems as a multi-label classification problem suitable for ANN training is a novel contribution to neural network-QC research, and a new approach to NISQ error mitigation.

- Multi-label classification models were trained and validated on subsets of this data set. The models were also equipped with a mechanism enabling out-of-distribution (OOD) detection, enabling them to evaluate whether the experimental result was conclusive by determining whether any answer category can be predicted with sufficient confidence to be called an answer to the problem.

- The performance of the model in identifying solutions was evaluated using standard ML classification success metrics. The model's predictive performance was evaluated on various subsets of the full data set, allowing performance to be evaluated for specific noise types and amounts.

- An implementation from the conclusions of this work is proposed which integrates classical ANNs into quantum computing workflows as an error mitigation technique in a new way.

Chapter 2

Background and Related Work

2.1 Quantum Computing and the Qubit Gate Model

Quantum computing is a computational paradigm that leverages probabilistic quantum mechanical effects to solve some problems that are classically intractable [18]. Quantum effects leveraged by this paradigm include superposition, entanglement, and quantum tunneling. The superclassical speedup achieved by the use of these effects is known as *Quantum Advantage* (QA): the demonstrable and measurable success of processing some real-world problem faster or more efficiently on a quantum computer than on a classical computer. While quantum computational hardware has yet to scale to a size where QA may be exhibited for problems of practical interest, QA has been demonstrated for simple algorithms and artificial problems [12, 13, 19]. Practical QA has the potential to revolutionize the world of computing in the near term, and problem domains such as particle modeling, quantum simulation, encryption, and machine learning will be changed forever by future generations of practical quantum computational hardware.

The predominant theoretical model of quantum computation is the qubit gate model, in which computation is decomposed into individual binary-state quantum systems called *qubits*. While a classical bit may exist in exactly one of two possible states, the superposition principle permits a qubit to simultaneously exist in both

states, during which a classical observation of the qubit will probabilistically collapse it into either of the two classically-observable states. Superposition is modelled mathematically as a unitary linear combination of all possible states of the system. Qubits can be combined to compose a quantum register in which an n-qubit system is represented by a superposition state vector in 2^n-dimensional Hilbert space:

$$|\psi\rangle = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_{2^n} \end{bmatrix} \tag{2.1}$$

With vector components $\alpha_1, \alpha_2, ...\alpha_{2^n} \in \mathbb{C}$. Qubits in a register compose the basic information bits of a quantum computer, and during computation the register state is evolved through unitary Hamiltonian energy transformations abstracted as quantum gates, manipulating the register toward a desired final state which corresponds to the answer to a problem. Once this state is achieved, the system is measured via a non-unitary classical transformation which collapses the probabilistic state wavefunction of the qubits into a discrete classical state.

The qubit-gate model formalism uses as its computational basis two states that are represented in Dirac notation as $|0\rangle$ and $|1\rangle$. These states are orthogonal vectors which span the vector space describing all possible qubit states. The state of a one-qubit system may be represented as a unitary linear combination of possible states, as in Equation 2.2:

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle\,; |\alpha|^2 + |\beta|^2 = 1. \tag{2.2}$$

The probability amplitudes α and β are both complex, and the square of the norm of each amplitude $|\alpha|^2$ or $|\beta|^2$ is equal to the probability of observing the qubit in that state. This is known as the *Born rule* [20] and is a fundamental concept of quantum

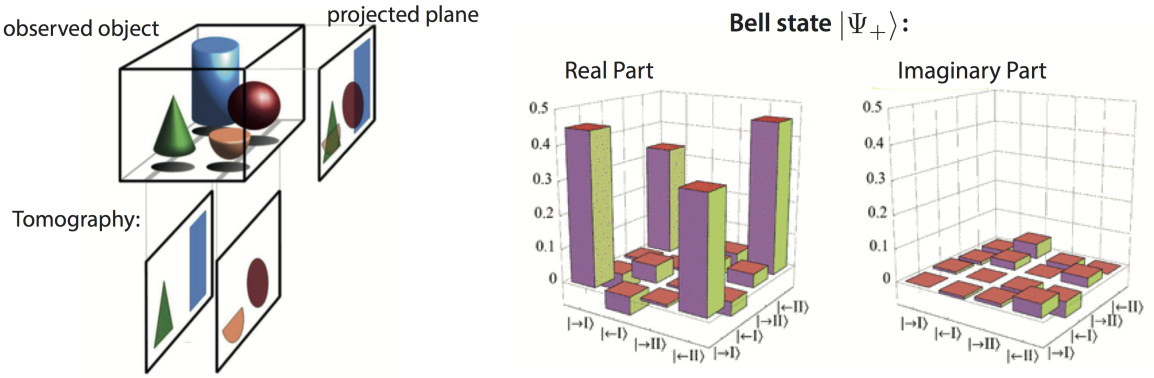

Figure 2.1: Illustration of quantum state tomography. Schematic illustration of the concept of tomography on left; reconstructed density matrix of a neutronic Bell state on right [5] [6].

information science. The solution to a quantum problem may be encoded as the specific state or states which are measured most frequently, known as *basis-encoded states* [21]. This work focuses on the class of problems which are basis-encoded.

2.1.1 Quantum State Tomography

Tomography is the reconstruction of higher-dimensional data by the process of building from lower-dimensional sections or "slices" of the original data. An example of this is a CT (computerized tomography) scan, which combines a series of X-ray images into detailed internal views of body systems which are hidden from view without invasive medical procedures. Similarly, the state of a qubit register before measurement is unknown; without observation, qubits in the register may exist in any complex-valued superposition of their possible basis states. It is well-known mathematically how the energy Hamiltonian (and thus the qubit register's statevector) evolves during a quantum circuit, but measurement is a non-unitary operation which interacts with and disturbs the energy state of a qubit, causing its wavefunction to collapse into one of the basis states. Thus it is impossible to observe qubit state evolution directly and so it must be extrapolated through quantum state tomography (QST).

QST is the process by which the results of a quantum algorithm are determined,

and it provides a view into the state of the qubit register before measurement. Measuring bits in the register causes them to collapse into one of the two basis states $|0\rangle$ or $|1\rangle$. Due to the probabilistic nature of QC, making only a single measurement of the system after a circuit is executed provides an insufficient picture of the results of the experiment. Information is encoded in the probability amplitudes or statevector of the qubit register and making one (or even a few) repeated measurements results in the loss of this information. Executing quantum experiments is thus a *multi-shot* process, in which the qubits are initialized to a known rest state, the circuit is executed, and measurements are taken in a repeated process, hundreds or thousands of times.

2.1.2 Noisy Intermediate-Scale Quantum Computing (NISQ)

Superconducting qubit systems require rigorous physical conditions to maintain the quantum properties of the qubit register. The system must be kept at an extremely low temperature (on the order of millikelvins), which permits low resistance and high conductivity, and it must be completely isolated from unintended environmental interference. These challenges lead to the dual phenomena of *quantum noise* and *state decoherence*, both sources of error which may render the outcome of an experiment invalid [22]. Figure 2.2 demonstrates the effect of noise sources on the outcome of a basis-encoded quantum experiment.

The problem class studied in the scope of this work, called the *basis-encoded class*, encodes the answer to problems in the quantum state or states which, after repeated experiments, are observed most often (with highest probability) [21]. As in Figure 2.2, the probability distribution of outcomes from repeated experiments may be represented graphically by a histogram of measurement results for each possible basis state, which normalizes to a probability distribution of observable states. However, the presence of error from quantum noise and state decoherence corrupt this probabil-

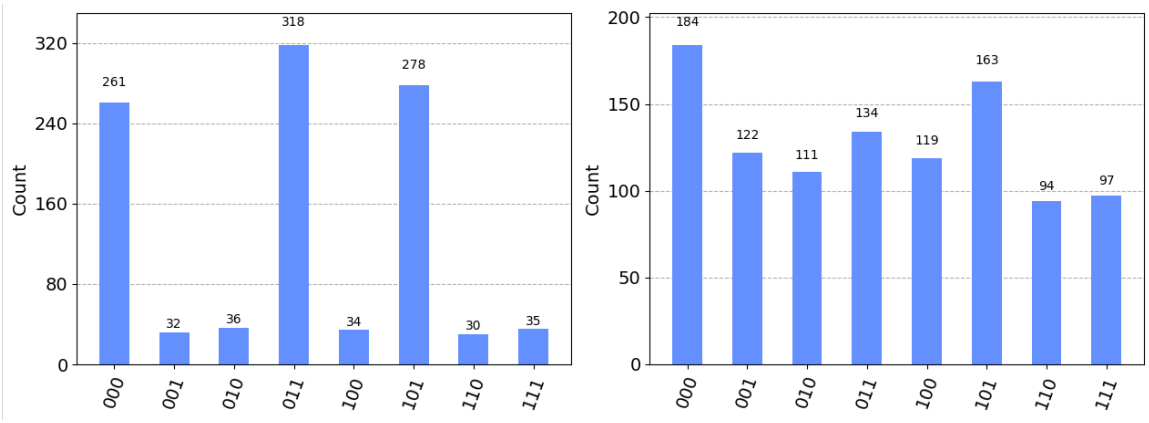

Figure 2.2: Comparison of noise-free (left) and noisy (right) simulation results for a quantum 3-SAT experiment with three solutions. Each simulation performed with 1,024 shots. The solutions {000}, {011}, and {101} are clearly recognizable without error from noise, but the correct solutions (and number thereof) are unclear from the noisy simulation.

ity distribution; with noise, the chance of measuring incorrect states is higher, so the chance of measuring the correct states is lower. Enough noise may corrupt the distribution entirely by causing incorrect states to be measured with higher probability than correct ones, rendering the experimental results invalid.

2.1.3 Error Correction and Error Mitigation

Currently, two strategies exist which attempt to control the influence of error on the outcome of an experiment: *error correction* routines and *error mitigation* strategies. Quantum error correction routines are similar to error detection and correction routines in classical information theory in that they may correct for arbitrary error given enough dedicated resources (ancilla qubits or gate operations) [23]. However, due to the limited hardware capabilities of today's quantum computers, arbitrary-precision error correction is not achievable as qubits are in short supply and gate operations must be limited to preserve qubit coherence. Pre-measurement error mitigation strategies intend to reduce the effect of noise by addressing its sources: reducing the number of gate operations or qubits required for the algorithm to execute [24]. This process may come at the cost of some accuracy in the final theoretical result,

but actual accuracy may increase since the error sources have less effect overall. Error mitigation strategies may also be applied after execution of the quantum circuit, using the experimental results as input to another algorithm or method. The method implemented in this work is an example of a post-measurement NISQ error mitigation strategy. Section 2.3 discusses other works in detail which develop error mitigation strategies, many of which employ machine learning.

2.1.4 Grover's Algorithm

Grover's Algorithm is a fundamental and important quantum algorithm for unstructured database search [17] and is used throughout this work as the primary basis-encoded algorithm for neural network characterization. This algorithm, one of the first theorized to demonstrate quantum advantage, can find an item or items in a database faster than classical search algorithms. Given a database of n elements, the best classical search algorithm may take at worst $O(n)$ time since, on average, half the domain must be checked for an even chance to find the right item. Grover's Algorithm, however, requires $O(\sqrt{n})$ iterations, providing at most a quadratic speedup for the problem. A quantum circuit of k qubits can search a database of at most 2^k elements, and the returned elements are basis-encoded in the output probability distribution of the multi-shot experiment.

The goal of Grover's Algorithm is to perform a search of the database. A single

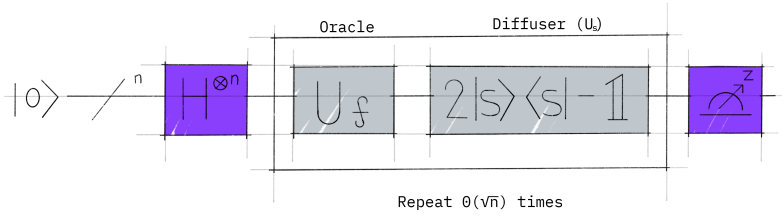

Figure 2.3: Steps of Grover's Algorithm for unstructured database search [7].

iteration has two steps: *phase inversion* and *inversion about the mean*, performed by the *oracle* and *diffusion* operators, respectively. A visual depiction of the algorithm is shown in Figure 2.3. To begin, the register qubits are put in equal superposition of basis states with Hadamard gates. The *oracle* identifies the targets by inverting their phase (and resulting probability amplitude), as shown in Figure 2.4. The oracle defines the search, and proper creation of the oracle within the context of Grover's Algorithm can be used to solve any NP-hard problem which reduces to database search, including cost function minimization and optimization problems.

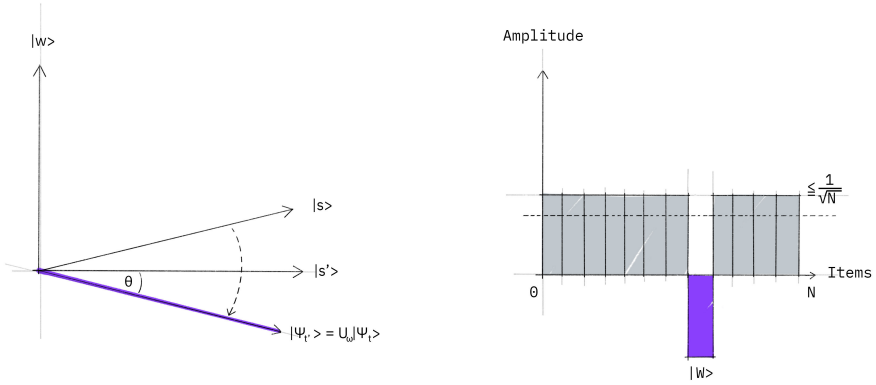

Figure 2.4: Phase inversion step of Grover's Algorithm, performed by the oracle, in geometric and amplitude representations [7].

Following the oracle, the diffuser inverts the probability amplitude of all items about the mean, as shown in Figure 2.5. Since the oracle step decreases the mean of the amplitude, inversion about the mean has the effect of lowering the probability of non-target items and amplifying the probability of the targets. This two-step process is repeated approximately $\sqrt{\frac{N}{m}}$ times when searching for m items in an N-item search space, until the probability of measuring the desired states corresponding with the target items is as high as possible.

As mentioned, Grover's Algorithm forms the basis of a broad class of quantum applications. One such problem is the maximum clique problem for finding the largest fully-connected subgraph (clique) of a given graph [11]. In that work, an oracle is

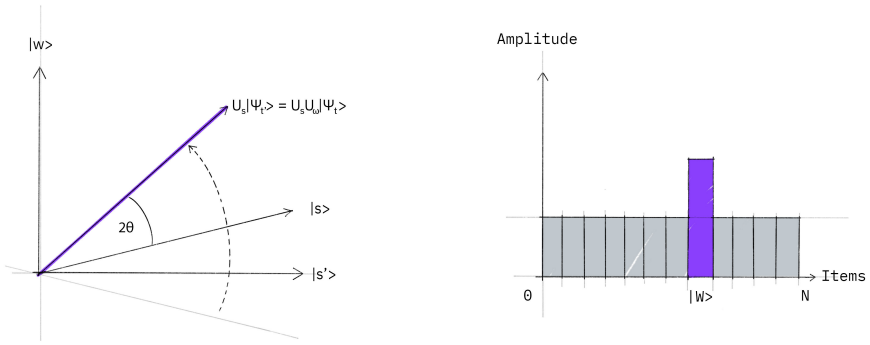

Figure 2.5: Inversion about the mean in Grover's Algorithm, performed by the diffuser, in geometric and amplitude representations [7].

developed which flips a target qubit if the state represents a subgraph that is larger than the minimum possible number of vertices for the clique. The diffuser reads this target to identify cliques over the minimum acceptable size. A second use of Grover's algorithm is the Boolean satisfiability problem (SAT), the problem of identifying a combination of Boolean variables (true or false) which cause a given Boolean formula to evaluate truthfully [25] [26]. SAT has been shown to be NP-complete [27]. Grover's Algorithm is observed to provide speedup for NP-complete problems [28], indicating that Grover's can speed up SAT and any problem which is proven to reduce to it.

2.2 Artificial Neural Networks

Neural networks or artificial neural networks (ANNs) are a category of machine learning architecture which are intended to model the electrical connections found between neurons in a biological brain [29]. Individual *nodes* or *neurons* are computational units which apply an activation function to the sum of their inputs and, based on the activation results of the function, may send the result to one or more nodes deeper in the network. Each input to a node has an associated weight applied to the input value, and inside the node, the resulting weighted values from each input are added

together, as in a sum-of-products equation:

$$s = b + \sum_{n=1}^{N} a_n w_n \tag{2.3}$$

with n node inputs a_n, corresponding weights w_n, and a tunable skew or bias value b. In the node, the sum s is given to an activation function which determines whether the node will send the value across its outputs to any connected nodes. The most commonly used activation functions are the sigmoid and rectified linear unit (ReLU). A model of an n-input neuron is shown in Figure 2.6.

ANN topology is a robust area of research; many topologies exist, but the most common network used in machine learning is the *feed-forward network*, in which nodes are organized into *layers*. In a feed-forward ANN, there are at least three layers: an input layer, a hidden layer, and an output layer. The connections between layers, called *edges*, link the output of nodes in one layer to the input of nodes in the next layer. A single-hidden-layer feed-forward ANN is shown in Figure 2.7. At the output layer, the activated neurons can be interpreted to form a classification or category label which corresponds to the solution to a problem. The training process for an ANN occurs through gradient descent. An error is determined between the ANN's output and the true solution to the problem, called the *loss*, and the loss is used to modify the neuron weights for the next round of classification. Learning occurs in

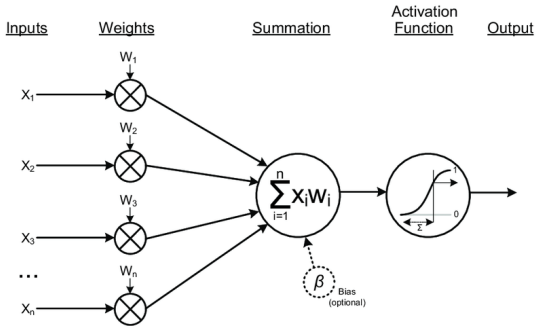

Figure 2.6: n-input neuron in an artificial neural network with a sigmoidal activation function [8].

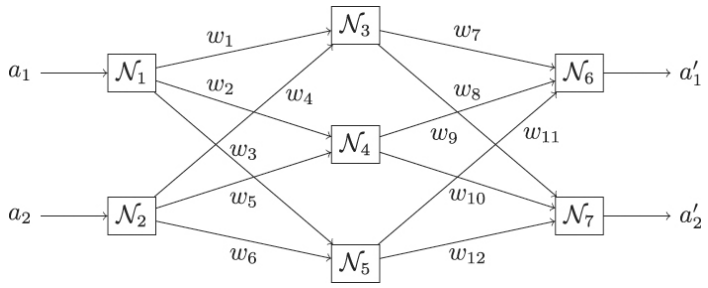

Figure 2.7: Single-hidden-layer feed-forward ANN with two inputs, two outputs, and seven neurons [9]. This is a *fully-connected network* with all neurons connected to the inputs of every neuron in the next layer.

the ANN over these rounds, called *epochs*, through the process of adjusting weights and thresholds in the network with the ultimate goal of minimizing loss.

Determining appropriate neuron weights through training is the most computationally intensive part of using ANNs. Training requires a large set of data, with many distinct input data samples and a corresponding expected output or label for each. By working through a set of data across multiple epochs, the ANN can be trained to answer questions which are similar to those in the input data set.

2.2.1 Multi-Label Classification

ANNs are useful for two types of predictive tasks: classification and regression. Classification tasks seek to predict a *discrete* identity or class label based on the input data, while regression tasks seek to predict a *continuous* value, often for fitting or identifying a trend in the data. In this work, the data under analysis by the ANN is a set of discrete probability distributions resulting from normalized multi-shot QC experiments. The ANN is tasked with predicting characteristics of the accurate solution set of such an algorithm. Since the ultimate goal of this work is the identification of discrete solution states of basis-encoded algorithms, it is representative of a *classification* task.

Classification tasks may be *binary*, *multi-class*, or *multi-label* depending on the type of category labels and how they may be applied to the data. *Binary* tasks

are those with just two possible disjoint labels for each sample: a sample may be categorized as in exactly one of the two classes. *Multi-class* tasks are those that have more than two labels, but each label is disjoint: any sample still belongs to exactly one of the categories. *Multi-label* problems are more complex, since in these tasks there may be many possible labels but they are no longer disjoint: each sample may take more than one label (or possibly none). A visual representation of these three types of classification problem is shown in Figure 2.8. Since the problem being answered by a quantum computer may have more than one answer and these answers coexist in a basis-encoded problem's probability distribution, the answer class labels are not disjoint and it is a characteristic **multi-label classification** task.

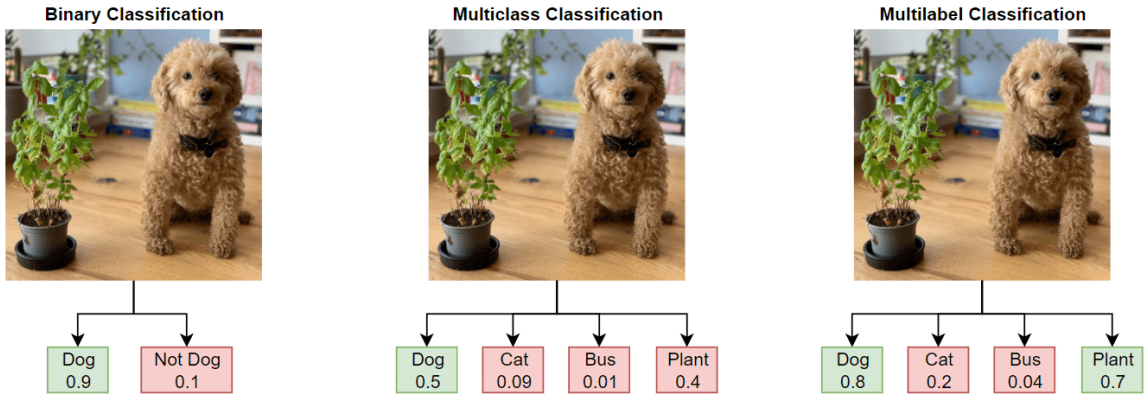

Figure 2.8: Types of ML classification task. Binary classification separates data into two disjoint categories. Multi-class classification decides between many disjoint categories. In multi-label classification, multiple categories may be assigned simultaneously [10].

2.3 Related Work

Research of ML applications to QC is a developing area of study and this work enhances the existing body of research by representing the task of identifying characteristics of a quantum probability distribution as a multi-label classification problem. The majority of existing research applying ML to QC focuses on the study of autoregressive models and their use as a QST technique [30]. As discussed in Section

2.1.1, QST is the process of reconstructing the qubit register's statevector before measurement [31].

Much existing research [32, 33, 34, 35] explores tomographic schemes which implement machine learning for the purpose of identifying the quantum statevector at some point in time before measurement, a regression task. For example, Torlai *et al.* [32] approach this task by using a tensor network encoding of quantum states and unsupervised learning (with no reinforcement or data labeling). Yu *et al.* [35] use a semiquantum reinforcement learning algorithm to reconstruct an unknown photonic quantum state in the continuous-variable QC formalism (as opposed to the qubit-gate model). Schmale *et al.* [34] use the density matrix representation of states as an encoding to choose a favorable convolutional neural network architecture for QST, and contrast the results to the most popular tomographic tool, maximum likelihood estimation (MLE).

Ahmed *et al.* [33] implement a deep convolutional neural network for classification tasks in QC and demonstrate its efficacy in in the presence of multiple noise sources, but focus specifically on states of continuous-variable photonic quantum systems, categorizing the Wigner state of the system as belonging to one of eight discrete representative classes. This research does not attempt to classify basis-encoded quantum algorithms or predict their results.

Mundada *et al.* [36] implement a complete automated error-suppressing workflow for quantum algorithms, which is evaluated on quantum circuits of up to 16 qubits implementing real algorithms. This process includes a post-experiment measurement error mitigation step designed to suppress readout errors, which constructs a confusion matrix of bit-flip error probabilities from groups of qubits in the circuit. This matrix is used as input to a neural network which is tasked with restoring correlations neglected between the groups and accounting for small nonlinear effects not captured by the confusion matrix formalism. This step applies ML to QC error mitigation, but does

so as an refinement to an existing error mitigation strategy based on the tensored confusion matrix protocol presented by Nation *et al.* [37]. The machine learning models are of the deep *ResNet* architecture, rather than the simple single-hidden-layer ANN models of this work, and they are not trained directly on the circuit results or used to predict answers to the problem.

Bennewitz *et al.* [38] present a neural network error mitigation scheme which is used to improve estimates of ground states and ground-state observables of quantum systems, a problem class called *variational algorithms*. This involves the application of the neural network quantum state tomography process by Torlai *et al.* [32]. These works focus on the application of ANNs to QST, reconstructing the pre-measurement quantum states using unsupervised machine learning approaches. These approaches differ from this work in both the problem class studied in [38] and the type of ANN and the ultimate goal of its predictive capabilities in [32].

As mentioned, many of these works focus on regression tasks by ANNs, using them to gain information on the pre-measurement quantum states, and their application to variational algorithms which may achieve near-term quantum advantage. By keeping the focus of this work exclusively on basis-encoded algorithms using the qubit gate model and their tomographic post-measurement solution spaces, the difficult problem of accurately predicting the *continuous* probability distribution of the system (characterizing the real probability of measuring each possible state of the register) can be reduced to a *discrete* prediction problem on the output of real quantum systems (identifying the states which represent to the answer to a basis-encoded problem).

Chapter 3

Methodology

This work proposes a new approach to identify characteristics of the true solution set of basis-encoded quantum algorithms, even in the presence of error from approximation or from hardware noise. At a high level, this approach involves training single-hidden-layer MLP ANNs using a back-propagation supervised learning strategy to perform multi-label classification tasks on a dataset consisting of the output probability distributions of basis-encoded quantum algorithms, simulated both with and without noise, and including output from emulated NISQ hardware. The neural networks trained in this manner are evaluated based on standard ANN accuracy metrics such as training and validation accuracy and loss and test prediction precision and recall, and compared to each other and to their own results on different algorithms and with different sources and amounts of input error.

As demonstrated by the results in Section 4, a MLP ANN can accurately identify the solution set of an arbitrary basis-encoded quantum algorithm when implemented on noisy hardware, given an appropriate set of training data and trained for multi-label classification. This approach applies supervised learning techniques to the probability distributions from both ideal and noisy quantum simulations: the data is tagged with the correct solutions to the problem during training in order to increase accuracy. Characteristics of the discrete probability distributions are evaluated to determine what ANN input data results in the most accurate identification

of solutions.

- A data set was created from the output of simulated quantum systems for a subset of basis-encoded problems. This data set consists of approximately 230,000 experiments of different problems which leverage the quadratic speedup of Grover's Algorithm for unstructured database search [17]. Additionally, The data set contains experiments from both ideal (noise-free) simulations and from simulations introducing various types and amounts of noise. Each experiment was performed with at least 1,000 shots to create an accurate tomographic reconstruction of the experimental results.

- The experimental results in this data set were normalized into discrete quasi-probability distributions and the solutions to each problem were supplied as multi-hot encoded multi-label classification categories against which the models are able to train. Characterizing basis-encoded quantum problems as a multi-label classification problem suited to ANN training is a novel contribution to neural network-QC research.

- Multi-label classification models were trained and validated on a subset of this data set. The models were also equipped with a mechanism enabling confidence-based out-of-distribution (OOD) detection, enabling them to evaluate whether the experimental result was conclusive by determining whether any answer category can be predicted with sufficient confidence to be called an answer to the problem.

- The performance of the model in identifying solutions was evaluated using standard ML classification success metrics. Evaluating the success of multi-label classifiers is an active area of study, and recent work on the evaluation and visualization of multi-label classification performance is explored [39]. The model's

predictive performance was evaluated on various subsets of the full data set, allowing performance to be evaluated for specific noise types and amounts.

- An implementation from the conclusions of this work is proposed which integrates classical ANNs into quantum computing workflows as an error mitigation technique in a new way.

The quantum circuits for each problem were generated using the Qiskit SDK and executed using the Qiskit Aer simulator [40]. Each single experiment was comprised of 1,000 shots and the shot counts were normalized to create discrete probability distributions separated by solution bit-string, on four-bit-wide solution problems, from $|0000\rangle$ through $|1111\rangle$. Each experiment's quasi-probability distribution was recorded along with its list of solutions as an entry in a CSV file, an ideal format for neural network training input. This process was repeated for all experiments shown in Table 3.2 until the entire data set of 229,800 experiments was generated.

3.1 Multi-Label Classification ANN Training

The data set was split into disjoint training, validation, and test subsets. 90% of the experiments were used for training, while 5% each went to validation and test. The measured shot probability of each possible state comprises the independent variable input data. The solutions were encoded as a binarized one-hot vocabulary of target values, making the single-experiment solution lists suitable targets for multi-label binary classification.

TensorFlow [41] and Keras [42] were used to construct the ANN trained for the classification task. The neural network architecture used in this work was a simple feedforward MLP ANN with a single hidden layer composed of 128 rectified linear unit (ReLU) neurons. The output layer uses sigmoid activation, which is standard for multi-label classification models in order to map the output values to probabilities

in $[0, 1]$. These output probabilities represent the confidence level of the model when applying the given label to the input data, and the model is defined as identifying a solution to the problem if its confidence level for that state is greater than 0.5.

This work employs a supervised learning approach. In this method, the model is supplied with the independent data against which it makes a prediction, along with the correct answer or answers to predict. A *loss function* is used to calculate the error or loss, which is based on the distance between the predicted solution and the actual solution in the n-parameter hyperspace of the problem. The models in this work employ the **binary cross-entropy** loss function, which is standard for multi-label classification problems (those that have multiple non-exclusive binary solutions). The model makes a prediction, calculates the loss, and changes its weights based on the process of *optimization*, whereby patterns in the change of loss per epoch give information as to the amount and direction to change weights. Over epochs, this approach trends towards a globally minimized loss, which means the model is making its most accurate possible predictions.

To perform optimization on itself, the model employs an *optimizer*, an algorithm that informs the model's changes to its weights and parameters. Based on the optimizer, there is a chance that the model may overshoot during optimization and get stuck in an oscillating pattern, never actually reducing its absolute loss. Basic first-order optimization, called *gradient descent*, may suffer from this phenomenon. As machine learning has matured, higher-order optimizers have been introduced which avoid the oscillation problem by considering rates of change of loss. The optimizer employed in this work is one such called Adaptive Moment Estimation, or ADAM. The ADAM optimizer requires no manual hyperparameter selection, has minimal variance between models, and converges very quickly, at the cost of high computational complexity compared to simpler optimizers. ADAM is a popular choice for supervised learning approaches for these reasons.

Overall, this model is easy to implement and has little training overhead, allowing future researchers the ability to iterate rapidly or adapt the model to more complex classification tasks on quantum circuit output. Data collection, training, and validation was performed using the resources from RIT Research Computing [43].

Two models were trained with these hyperparameters. One was trained on exclusively Grover's algorithm-based applications, defined in the following section. A second was trained on a mixture of all quantum circuit data, including the quantum integer multiplier implemented in [44], which is basis encoded but not Grover's-based or oracle-based. This provides an opportunity to determine how the model performs when identifying the solutions to a basis-encoded problem class it has not encountered.

3.2 Benchmark applications

Grover's Algorithm [17], discussed in Section 2.1.4, is at the core of the applications that have been targeted in this work, as it is a very versatile quantum algorithm with many possible applications. Even though this algorithm only offers polynomial speedup ($O(\sqrt{N})$ vs. $O(N)$), the goal of this work is to offer a proof of concept of the potential for post-measurement error mitigation of ML training and classification process in basis-encoded applications, and Grover's algorithm serves its purpose well. Other basis-encoded algorithms can potentially benefit from this approach, such as period finding in Shor's algorithm which can potentially have multiple solutions, or arithmetic operations with integers, such as integer quantum multipliers [44]. The cases in this paper, however, are sufficient to demonstrate the efficacy of the approach, since the key component is how the noise affects the ideal probability distribution rendering the actual solutions undecided, and how ML classification can help extract information nonetheless. This section introduces the test cases that were used for training, validation, and testing: problems that are basis-encoded and with

Table 3.1: Circuit features

Application	Width	Depth	Size
3-SAT	[3]	[8-137]	[15-182]
4-SAT	[4]	[8-329]	[18-426]
Max Clique (Approx.)	[18]	[25-34]	[55-68]
Max Clique (Exact)	[18]	[33-242]	[66-403]
QBNN	[12]	[13439-89609]	[13464-109657]

potentially multiple valid solutions. A table of relevant benchmark circuit features by application including circuit width (number of qubits), circuit depth (number of operational stages), and total gate count is shown in 3.1.

3.2.1 Grover's-based applications

Grover's implementation at a higher level can be broken down into three steps: initialization, oracle and diffuser, as summarized in Figure 2.3. Although the initialization and diffuser steps are the same for every application of Grover's algorithm, the oracle holds the core of the specific applications. Any optimization problem can be re-framed as a search problem, as long as the optimization conditions are expressed in the oracle component of the search algorithm, and as long as the output can be expressed as a binary string, as is the goal in the basis-encoded applications targeted here. This Oracle holds all the versatility of Grover's algorithm and offers enough varying parameters to test this approach on different applications. Three different applications of Grover's algorithm to computationally difficult problems are tested: *Boolean satisfiability*, *maximum clique*, and *training of binary neural networks.*

Satisfiability problem: The Boolean satisfiability problem (SAT, or k-SAT for k literals) is the NP-complete problem if determining if there exists a combination of Boolean variables (True or False) that may satisfy a given Boolean formula, causing the formula to evaluate to True. An example of a three-variable satisfiability problem

(3-SAT) is given in (3.1).

$$F = (\overline{A} \vee B \vee C) \wedge (A \vee \overline{B} \vee C) \wedge (\overline{A} \vee B \vee C) \wedge (\overline{A} \vee B \vee \overline{C}) \tag{3.1}$$

A formula may have multiple satisfactory combinations of variable assignments. Satisfiability problems are a well-known application of Grover's algorithm. The variable assignments that satisfy the formula are basis-encoded by assigning a qubit to each variable and specifying a Boolean encoding, such as 0 to False and 1 to True. A Grover's oracle is constructed in the 2^k-value search space of the k variables in the formula and the algorithm is performed to find the solutions. Satisfiability is an ideal problem for dataset generation, as it is fast to execute and modify to generate SAT problems with different solutions and numbers thereof. While SAT circuits offer excellent programmability and the lowest circuit width, they suffer from a high circuit depth relative to others and thus a larger potential accumulation of error from noise.

Maximum clique problem: The maximum clique problem is the problem of finding the largest fully connected subgraph of a graph. A graph can have more than one maximum clique, as shown in Figure 3.1. The possible solutions may be basis-encoded when the graph's edges are expressed as one-hot in the connectivity matrix of the graph. Grover's algorithm was used in a previous work to solve this

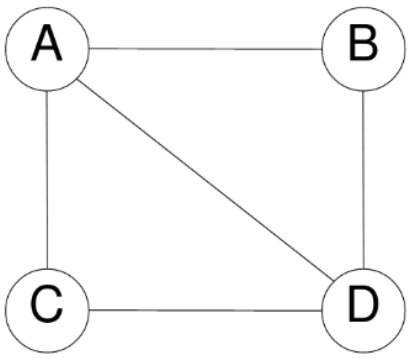

Figure 3.1: Four-node graph demonstrating the maximum clique problem [11]. The graph shown has two maximal cliques, ABD and ACD.

problem [11]. In this work, three-node graphs with every possible combination of edges were used for data generation. This resulted in three cases with a single answer and three cases with two answers. Two implementations from [11] are used as a test case here: one which solves the maximum clique problem exactly, and a second one in which the implementation is approximate to reduce the depth of the circuit. Approximate algorithms such as this are an example of a pre-measurement error mitigation strategy, costing some experimental accuracy in ideal simulation but with the potential to increase actual accuracy on NISQ hardware by reducing the ability of noise to accumulate in the circuit.

Training Binary Neural Networks:

In *Improved Grover's Implementation of Quantum Binary Neural Networks* [45], a quantum circuit is introduced which trains a binary neural network (BNN), providing optimal BNN weight strings as a basis-encoded output. The execution of the QBNN has 2^N possible weight strings, where N is the number of weighted edges in the network. Since the cost function associated with the BNN training data may have multiple minima, the QBNN may provide multiple optimal weight strings as answers. In this work, a simple implementation is used with a single neuron with three inputs and one output. The dataset has a single best weight string, "000". This allows the inputs to enter the neuron unchanged, and the neuron's activation function $th = 2$ outputs a 1 when the inputs sum greater than or equal to 2. The MC and QBNN implementations in this work are less programmable than SAT and use many more qubits, but are good examples of recent complex quantum circuits.

3.2.2 Non-Grover's-based application

Quantum Array Multipliers (QAM) were proposed by Crimmins *et al.* [44]. Taking inspiration from classical array multipliers, they implement integer multiplication taking advantage of rotations in the phase domain through the Quantum Fourier

Transform. The answer is basis-encoded, making tThere is only one correct result per experiment: the resulting product of the two input integers, but noise and approximations contribute to a less definite probability distribution. In this case, since for the sake of simulation cost the largest output in training and testing is four qubits wide, only $2 \times 2 = 4$ qubit outputs are implemented and tested with different noise levels. Therefore, the classes in this case are limited to $1 \times 1 = 1$, $1 \times 2 = 2$, $1 \times 3 = 3$, $2 \times 2 = 4$, $2 \times 3 = 6$ and $3 \times 3 = 9$.

Table 3.2: Total experiment count for each problem.

8 State

Problem Type	Solutions	Experiment Count
3-Sat	0, 3, 5	11,000
	3, 5	11,000
	2, 4, 6	11,000
	1, 2	11,000
	0, 1, 3, 5, 6	11,000
	7	11,000
	2, 4	11,000
	1, 7	11,000
Max. Clique (exact)	3, 5	3,000
	3, 6	3,000
	5, 6	3,000
	3	3,000
	5	3,000
	6	3,000
Max. Clique (approx.)	3, 5	1,100
	3, 6	1,100
	5, 6	1,100
	3	1,100
	5	1,100
	6	1,100
QBNN	0	1,200
QAM	0	300
	1	300
	2	300
	3	300
Total		**115,000**

16 State

Problem Type	Solutions	Experiment Count
4-Sat	0, 15	11,000
	7, 11, 12	11,000
	2	11,000
	1, 5	11,000
	4, 6, 9	11,000
	13	11,000
	3, 14	11,000
	2, 8, 10, 12	11,000
	0, 4, 7, 9	11,000
	11	11,000
	5, 13, 15	11,000
	1, 10, 11	11,000
	3, 6, 8, 14	11,000
QAM	0	300
	1	300
	3	300
	4	300
	6	300
	9	300
Total		**114,800**

Chapter 4

Results

4.1 Method, experiments and metrics

Classification of each probability distribution was performed according to 16 labels. Each label corresponds to the binarized encoding of the measured answer state for each basis encoded problem, e.g. $|0000\rangle$ as the answer or label "0" and $|1011\rangle$ as the answer or label "11". These classes are used for a direct one-to-one correspondence between the solutions to each quantum algorithm and the classification labels for the machine learning model. A supervised learning approach was used to train two different models of the architecture described in Section 3.1. *Model 1* was trained using only Grover's-based application data (SAT, Max. Clique, QBNN). *Model 2* was trained using Grover's-based *and* non-Grover's based (QAM) application data. Three different experiments evaluate the performance of each of the two models on novel data. The first two experiments assessed *Model 1*, while the third one assessed *Model 2*.

Experiment 1 evaluates *Model 1*'s performance using a 1,000-sample subset of the Grover's Algorithm dataset which is disjoint from the training and validation data and includes all noise levels. Benchmarks in this test include satisfiability, maximum clique, and training a binary neural network. This test provides a baseline performance level on a wide variety of distributions with either 8 or 16 possible solutions

and with various amounts of noise.

Experiment 2 evaluates *Model 1*'s performance in finding the solutions to the quantum array multiplier circuit, QAM [44]. QAM has only one label in each sample distribution, the product of the two input integer values. The test set included various noise levels, to distort the probability distributions to different degrees. The quantum multiplier is a basis-encoded circuit not based on Grover's algorithm. This test is useful to evaluate the model's performance on a problem type that it has not encountered during training.

Experiment 3 is performed with *Model 2* and the test evaluates its performance on a 1,000-sample subset of a test set comprised of data from all benchmarks and all noise levels. This set includes results from the QAM.

4.1.1 Noise models

In order to generate the data set used for the ANN classification task, Qiskit noise models were developed which emulate different sources of hardware errors. First, a configurable probabilistic Pauli error (bit-flip) model was written which takes as input a probability percentage, and constructs a noise model that will cause the qubits to flip states randomly during reset, measurement, multi-axis rotation (U) gates, and control-X (CX) gates. Next, a time-based single-qubit thermal relaxation noise model was written which models the expected execution times of various gates and randomly "relaxes" qubits during the gate operations to their rest $|0\rangle$ state. These noise models provide a degree of control over the noise on the basis-encoded quantum circuits considered in this work while accounting for the probabilistic nature of noise experienced by real quantum hardware.

In addition, five different simulated hardware noise models were used to generate data. These are simulated versions of the IBM Hanoi, Johannesburg, Guadalupe, Essex, and Almaden processors, implemented by applying the circuit noise models

and coupling maps to an ideal Aer simulator and transpiling the quantum circuits to hardware gates. Simulated hardware noise models provide insight into how the algorithms perform on real hardware and how the ANN performs when analyzing the output of real quantum computers.

4.1.2 Metrics

Evaluating the performance of multi-label classifiers is not straightforward when compared to binary or multi-class classification, and is itself the subject of active research [46] [47]. For example, standard multi-class accuracy metrics tend to inflate the accuracy of a multi-label classifier by counting every no-match category as accurate, even if the correct categories are not predicted accurately or the positive categories are sparse. This work uses the multi-label performance indicators introduced by Heydarian *et al.* [39] including the precision, recall, F1-score, and weight metrics, and the multi-label confusion matrix schema.

To evaluate multi-label classifiers, the metrics *precision*, *recall*, *F1-score* and *weight* are introduced and calculated separately for each class. Precision gauges the model's accuracy of positive predictions: the percentage of its optimistic guesses that are correct. Recall gauges the model's positive predictions against all instances of the class: how much of a given class label the model identified. F1-score is a specific F-score measure which gives the harmonic mean of precision and recall. Weight is the total count of a given class' labels in the data set. Equations 4.1, 4.2, and 4.3 give precision, recall, and F1-score for a class c, respectively [39].

$$P_c = \frac{TP_c}{TP_c + FP_c} \tag{4.1}$$

$$R_c = \frac{TP_c}{TP_c + FN_c} \tag{4.2}$$

$$F_c^1 = \frac{2TP_c}{2TP_c + FN_c + FP_c} \tag{4.3}$$

The *confusion matrices* in Figures 4.1, 4.2 and 4.3 show true positive predictions on the main diagonal. These are cases where the true solution was also predicted as a solution by the ANN. Entries off the diagonal show the number of inaccurate predictions of other solutions when the true label was a solution to the given problem. The final "NL" column shows cases where the model could not make a satisfactory prediction (that is, with confidence ≥ 0.5) when the True Label was present. The final "NL" row catches out-of-distribution data, or data with labels the model could not recognize. No data of this type was present in the data set as all data was tagged with a solution in $[0, 15]$.

4.2 Experimental results

Results for each experiment presented in Section 4.1 are presented separately, and a full summary follows. In all cases, tests were performed on a 1,000-experiment subsample of data which was reserved for test: the models did not encounter this data during training or validation. Each test set contains a randomized mixture of all noise levels used for dataset generation.

4.2.1 Experiment 1: Grover's trained model (Model 1) on Grover's test cases.

This experiment tested Model 1 against Grover's Algorithm test cases. Since the training and test data sets included three and four qubit output cases, the datasets are unbalanced with a higher number of cases in the 0-7 range than 8-15 range. If multi-label classification is meant to be effective, this pairing of training and testing on the same application type (Grover's) should show best results. As shown in Figure 4.1, the majority of predictions are true positive, reflected in the diagonal of the confusion matrix. The model is presented with a significantly higher workload of cases with answers between 0 and 7 due to the imbalanced nature of the dataset,

which is reflected in the class weights in Table 4.1 and in the upper left quadrant of the confusion matrix.

It is evident from these results that incorrect classifications are scarce despite dataset imbalance. False positives, columnar entries of a given class off the diagonal, are very low for every class although notably higher for class 3 and 5 again due to dataset imbalance causing training bias. False negatives, row entries of a given class off the diagonal, are similarly low throughout, although higher for classes 6 and 7. Weighted averages of precision and recall for this experiment are 95% and 84% respectively, which indicate the model is exceptional at being conservative with its positive predictions and adept at classifying the majority of cases it is presented with. F1 scores range from 0.77 to 0.94 across all classes.

Table 4.1: Overall MLC ANN Performance Metrics for Experiment 1.

Class	Precision	Recall	F1-score	Weight
0	0.93	0.77	0.84	197
1	0.93	0.92	0.93	214
2	0.99	0.90	0.94	222
3	0.88	0.86	0.87	258
4	0.98	0.87	0.92	181
5	0.91	0.83	0.87	241
6	0.94	0.80	0.86	234
7	0.99	0.76	0.86	204
8	1.00	0.78	0.88	91
9	0.94	0.78	0.85	97
10	0.95	0.83	0.89	89
11	0.97	0.80	0.88	124
12	1.00	0.70	0.82	93
13	0.97	0.67	0.79	84
14	0.80	0.74	0.77	86
15	1.00	0.68	0.81	95
Micro Avg.	0.84	0.84	0.81	2510
Macro Avg.	0.94	0.82	0.86	2510
Weighted Avg.	0.95	0.84	0.87	2510

Inconclusive cases make up the largest group of misclassified results, appearing in the final “NL” column in Figure 4.1. The inconclusive cases are also reflected

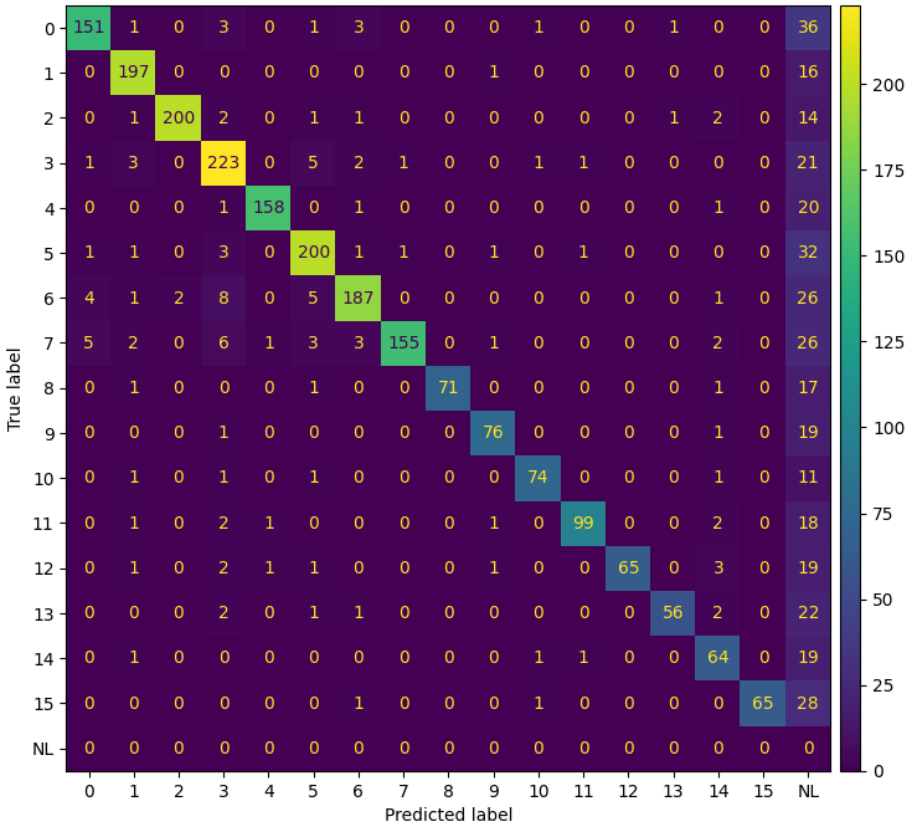

Figure 4.1: Multi-label confusion matrix for performance evaluation, Experiment 1.

in the Recall metric in Table 4.1, as no-label results are considered false negatives. Since inconclusive cases are counted as false negatives, they impact Recall and F1-score negatively. The model was not presented with as many cases of the lower-weight classes during training, and it is clear that this caused the model to be less confident making positive predictions of these solutions. On average across all classes, inconclusive cases represent 13.7 % of all tests. A review of the lower-weight labels shows that the model is particularly weak here; for labels 12, 13, 14, and 15, the model could not make conclusive predictions in over 20% of cases, peaking at 29.5% inconclusive predictions for problems with 15 as the answer. These percentages are also shown in Figure 4.4, along with the same metric from Experiment 3. These cases are the reason Recall scores are lower than Precision scores.

Under further inspection, these inconclusive cases were found to be mostly from

the Johannesburg hardware noise model and the 5% and 10% bit-flip probability models. While it might be the case that inconclusive answers come mostly from higher noise levels, it is also the case that the noise level is controlled for in the random test sample as there are equal numbers of experiments of each noise level. Since the number of conclusive predictions is still higher for the higher-weight classes even controlling for noise with random sampling, it can be concluded that both noise level and sample weight have independent effects on model confidence.

4.2.2 Experiment 2: Grover's trained model (Model 1) on QAM test cases

Table 4.2: MLC ANN Performance Metrics for Experiment 2. Classes with no samples are omitted.

Class	Precision	Recall	F1-score	Weight
0	0.75	0.49	0.59	252
1	0.86	0.47	0.61	261
2	0.84	0.37	0.51	117
3	0.70	0.52	0.60	249
4	0.78	0.45	0.57	100
6	0.56	0.50	0.53	114
9	0.94	0.45	0.61	98
Micro Avg.	0.75	0.47	0.58	1191
Macro Avg.	0.77	0.46	0.57	1191
Weighted Avg.	0.77	0.47	0.58	1191

This experiment tested Model 1 against non-Grover's Algorithm-based test cases, namely those from QAM. The goal of this experiment was to test how the applications used for training transpire into the ML model. In other words, these results mean to prove that quantum noise from a circuit is not just random noise but structured on the circuit's internal rules, and that a ML model can find patterns in this structure to make predictions. The models with this level of training are not universal; they will need to be trained for specific applications or groups of applications with the same structure or based in the same principles.

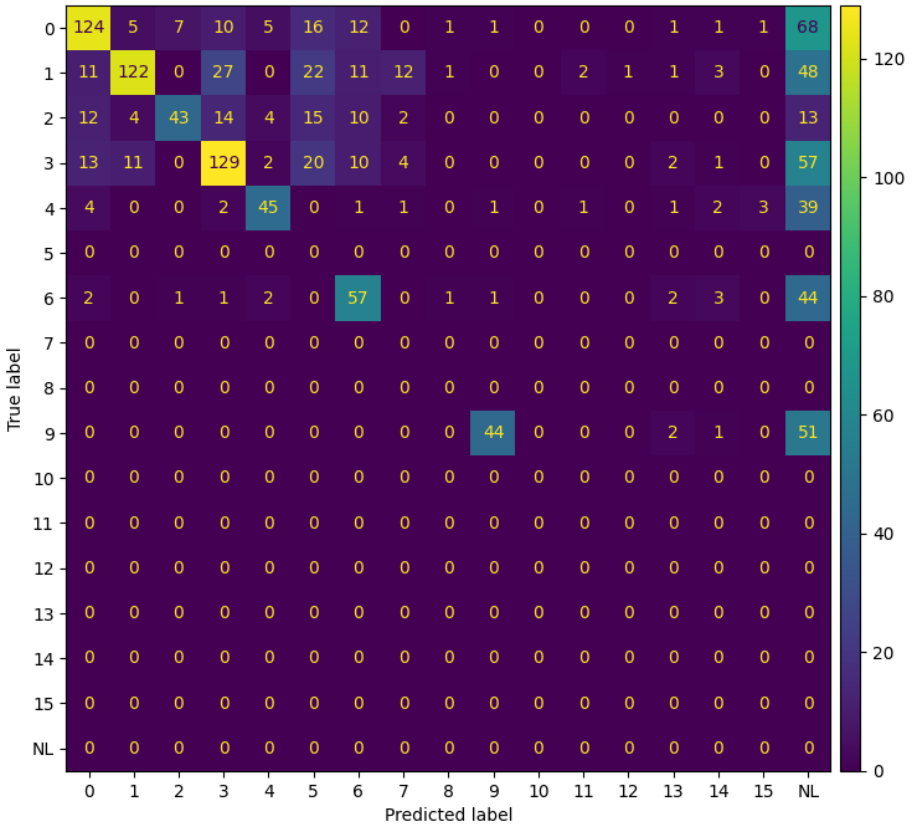

Figure 4.2: Multi-label confusion matrix for performance evaluation, Experiment 2.

The QAM data was not generated using the noise models described in Section 4.1.1. QAM with small input cases as discussed in these experiments (2 qubits $\times$ 2 qubits = 4 qubit result) are highly accurate under standard noise models. For that reason, the simulations used a unique noise model applied to all qubits which produced highly noisy results to test the accuracy of this ML model. Simulations were run at three levels: no noise (perfect results), depolarizing error with 10^{-5} and $5 \cdot 10^{-4}$ single-qubit and two-qubit gate error respectively, and a modified depolarizing error with 10^{-2} single qubit and two qubit gate error. Results from this test are outlined in Table 4.2 and Figure 4.2. Table 4.2 omits classes which were not present in the dataset.

As expected, it is observed that the model makes significantly more misclassifications and inconclusive classifications. Only between 37% and 52% of the tests are

correctly classified across all classes with an average of 47%. The high noise level cases represent 33% of the total number of test cases, but the misclassifications go well beyond that percentage. The reason is that the ML model was not trained on data coming from this circuit implementations. The model does not know how to handle the test cases or what to expect except for about 48-63% of the cases, which is an overall poor result.

This test has a secondary benefit of highlighting bias in the model's weights: due to the dataset imbalance during training, the model tends to falsely predict solutions 3, 5, and 6 more often than other categories. While Model 1 is much less confident making predictions on this data, it also makes many more erroneous predictions than in the other experiments. This is due to the skew toward solutions 3, 5, and 6 in the solutions of the training data.

4.2.3 Experiment 3: Grover's and non-Grover's trained model (Model 2) on all test cases (Grover's and QAM)

This experiment was conducted on Model 2 against a subset of data from all test cases, including both Grover's and non-Grover's (QAM) problems. 1,000 experiments were sampled out of the full test set, which contains data from all noise models. Thus Model 2 may encounter data from any problem of any noise level during this experiment. This experiment is meant to second the previous point. The ML model now has learnt the circuit implementation of a new application and is able to better classify cases from that instance. Further, the new results show that two plus two does not always equal four. In this case, the additional training on QAM not only improves upon Experiment 2, but outperforms Experiment 1 as well.

Model 2 displays good performance at classifying solutions from Grover's and QAM probability distributions, as shown in Table 4.3 and Figure 4.3, and in contrast with Experiment 2 demonstrates the importance of training the model on data sim-

ilar to that which it will encounter during testing or deployment. Overall weighted averages for precision and recall are 0.95 and 0.84 respectively, identical to those from Experiment 1. But the average F1 score is slightly higher in Experiment 3 which indicates a better rate of attempted classification when compared to Model 1. The extra training data that Model 2 saw made it a more confident predictor of the solutions to basis-encoded quantum problems. The reason for this change is that the number of inconclusive cases has dropped slightly in this case, when compared to Experiment 1, and the distribution has also changed. Figure 4.4 displays these results for inconclusive cases. Experiment 1 had an average 13.7% number of inconclusive cases while the average has dropped to 11.8% in Experiment 3. The probability distribution has for Experiment 3 also seems to be more uniform, less skewed towards the higher end of the figure. In a number of classes, the rate of inconclusive cases is lower in this new experiment than in Experiment 1 (0,4,5,7, 8,9,10,12, 13, 14, 15). It appears that QAM added diversity to train the model has a positive impact on the classification as a whole, but further analysis should be performed to reach solid conclusions.

4.2.4 Noise Model Analysis

It is important to gain understanding of the effect of noise on the models' predictive confidence. Randomizing the training and test data in previous experiments effectively controlled for the influence of noise by presenting equal amounts of each noise level to the models for training and testing, with the goal of assessing performance across solution categories and types of quantum circuits. By analyzing the model's performance categorized by noise level, it can be determined whether there is some threshold above which the model can no longer make predictions confidently.

To perform this analysis, the complete data set was split up by noise type, percentage, and hardware backend, and 1,000-sample groups were pulled from each. Model 2 was used in this analysis, as it had the best performance across the previous

Table 4.3: Overall MLC ANN Performance Metrics for Test 3.

Class	Precision	Recall	F1-score	Weight
0	0.94	0.82	0.87	195
1	0.98	0.86	0.91	215
2	0.96	0.89	0.93	196
3	0.93	0.86	0.89	267
4	0.99	0.88	0.93	159
5	0.93	0.84	0.89	264
6	0.89	0.84	0.86	217
7	0.96	0.84	0.90	186
8	0.96	0.84	0.90	90
9	1.00	0.79	0.88	92
10	1.00	0.88	0.93	73
11	0.96	0.76	0.85	126
12	0.97	0.87	0.92	75
13	0.94	0.78	0.85	100
14	0.95	0.84	0.89	88
15	0.91	0.71	0.80	83
Micro Avg.	0.83	0.84	0.84	2426
Macro Avg.	0.95	0.83	0.89	2426
Weighted Avg.	0.95	0.84	0.89	2426

trials. QAM data was omitted due to the inconsistent noise models used for those experiments. Precision and Recall for each noise type is shown in Figures 4.5 and 4.6.

Scoring metrics indicated in Figure 4.5 demonstrate the large effect that noise has on model prediction accuracy and completeness. Predictions are very accurate and recall is excellent for most noise levels, staying above 90% for all noise models except Johannesburg and the 5% and 10% Pauli error models. For these three categories, the model performs much worse. Especially for 10% noise, the model's recall is 0.33 or 33% of which is very poor. Precision is better at 0.62 which indicates that at this noise level the model is making many misclassifications, but is mostly unable to guess any answer states with significant confidence.

Looking at the inconclusive predictions across each noise model, the results of this analysis are shown in Figure 4.6. As the noise level increases in the quantum simulation, the solution states are measured less frequently and the resulting probability

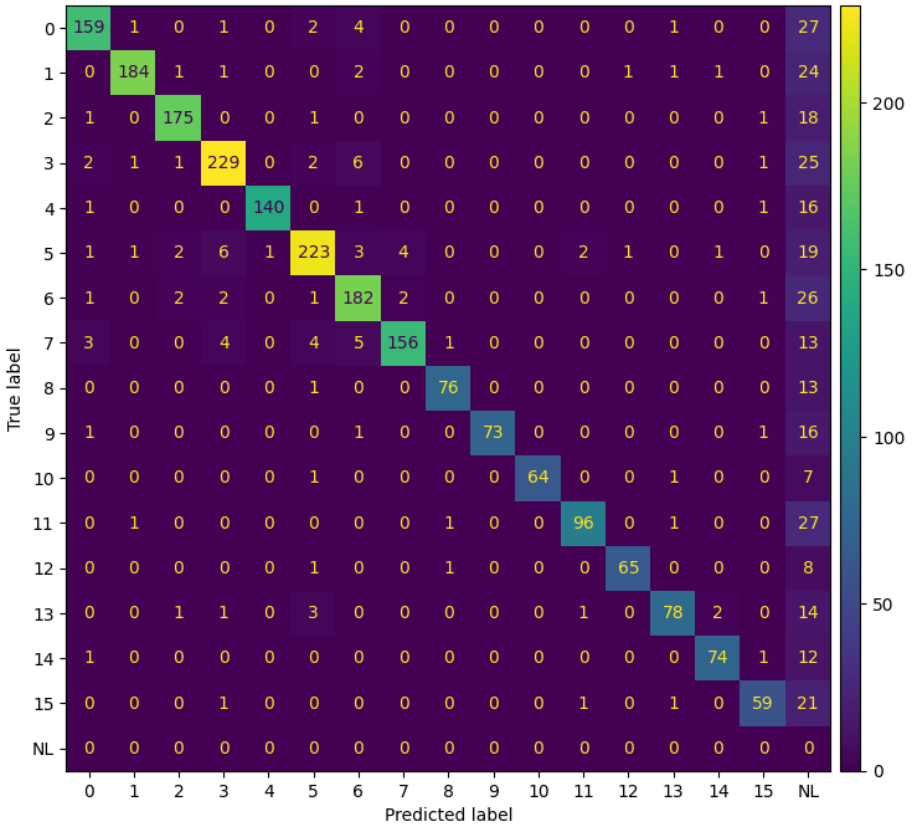

Figure 4.3: Multi-label confusion matrix for performance evaluation, Experiment 3.

distribution increasingly resembles a uniform distribution, making it more difficult for the ANN to classify the distribution. The ANN becomes less confident in its predictions and at 10% bit-flip noise, eventually cannot classify approximately 50% of the samples it is presented on a subset of all 10% noise data.

Knowledge of the distributions which resulted in inconclusive ANN output is a useful research outcome. These samples can be separated from the dataset and re-weighted for further training, or omitted from future training data altogether. These samples are also identified as good candidates for cleaning or other post-measurement error mitigation, a process that is not required for accurately classified data. Generally, when the model makes a prediction on an experiment it has been trained on, it makes an accurate one; with a confidence threshold of 0.5, numbers of inconclusive predictions are far higher than false positives.

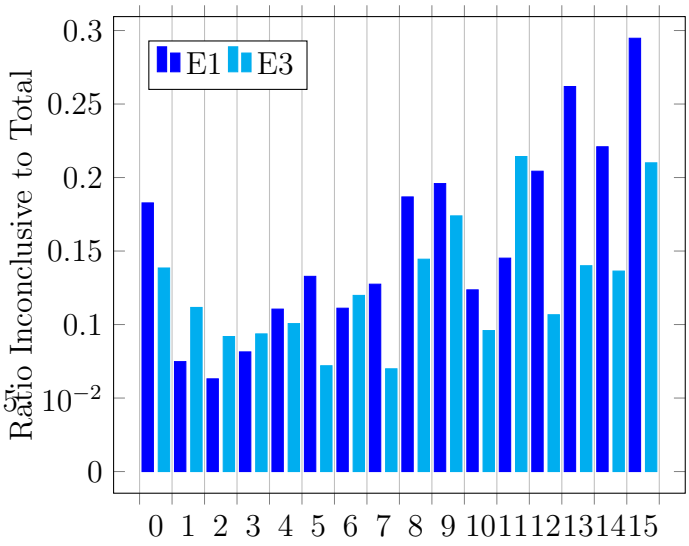

Figure 4.4: Fraction of inconclusive cases per label for Experiment 1 and Experiment 3 compared.

Of interest in this analysis is the poor performance on the data generated on the IBM Johannesburg quantum system noise model. At the time the noise model was extracted, this device was experiencing high error in both single qubits and two-qubit links. Identification of this system as particularly noisy when compared to the other hardware backends could indicate another use of machine learning analysis on quantum computer outputs. A well-trained model looking at the real-time running output of hardware quantum systems could be used as a preventative system monitoring tool, assessing when system noise rises above some threshold during operation.

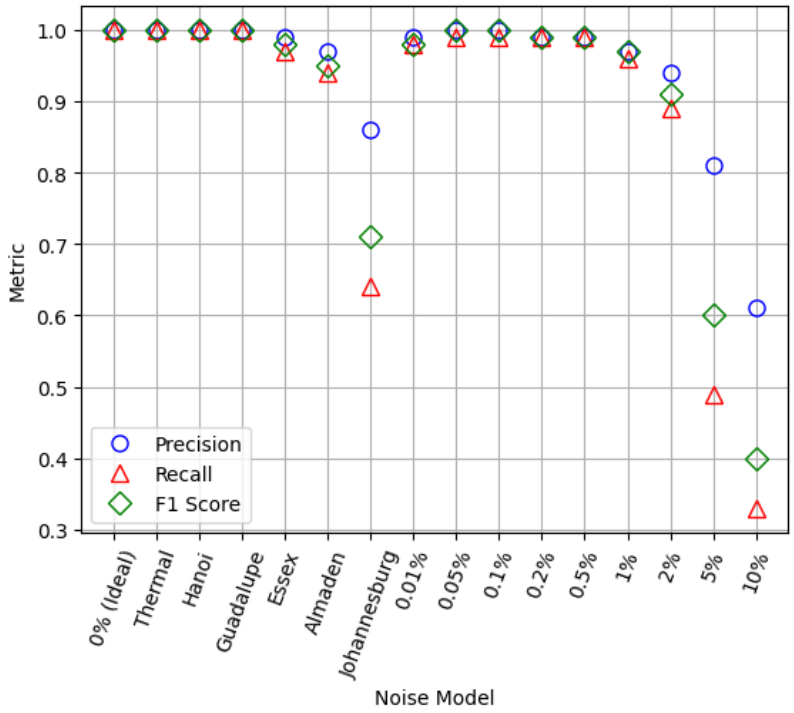

Figure 4.5: Weighted average Precision, Recall, and F1 score of sampled data sets segmented by noise model.

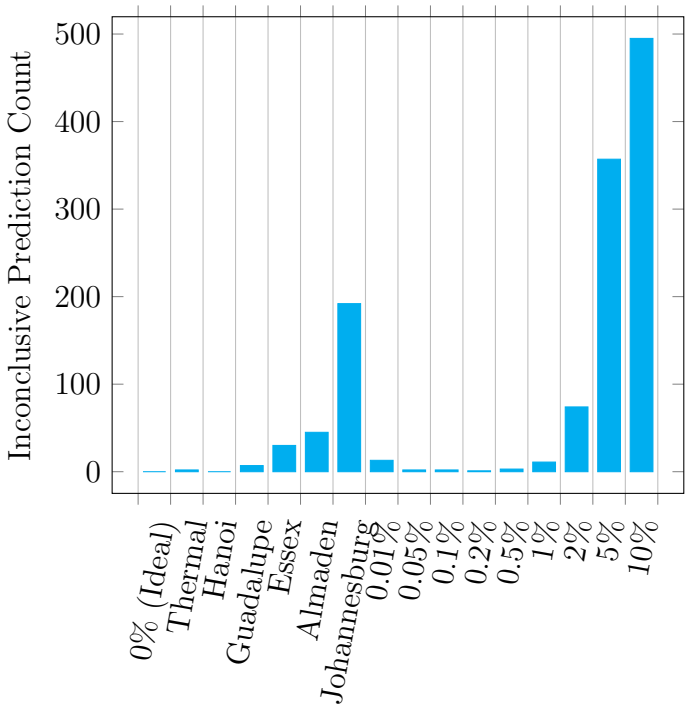

Figure 4.6: Number of Zero-Prediction items by Model 2 in 1,000 samples of each noise model dataset, with P(answer) ≥ 0.5.

Chapter 5

Conclusion

This work uses ML approaches to classify probability distributions generated out of quantum computing experiments, specifically approaching the problem of identifying solutions to a noisy quantum experiment as a multi-label classification problem. Simple multi-label classification models were used to identify the correct solutions of basis-encoded quantum applications. This process was successful despite noise that would otherwise result in inconclusive outcomes and particularly when the number of valid solutions to the problem is unknown beforehand.

The approach was tested on Grover's Algorithm-based applications (Satisfiability, Maximum Clique and Quantum Binary Neural Network training) and non-Grover's based applications (Quantum Array Multiplier). Due to resource and time constraints for simulation, training, validation, and test were limited to three- and four-qubit application outcomes. From the efficacy of the approach, these proof of concept experiments showed that applying discrete classification strategies to data generated on NISQ machines is a viable option with precision, recall and F1-score around 0.95, 0.84 and 0.89, respectively. As expected, the trained models had a large increase in the number of inconclusive cases when noise raised above 5%. These same models held up well with noise at 2% or under, or when noise was modeled after data taken directly from hardware, except in the case of the IBM Johannesburg processor, which was a particularly noisy example.

In addition to the successful classification of solutions to quantum experiments, another useful research outcome of this work is the identification of experiments which are good candidates for further noise mitigation. This work may serve as the first step toward an integrated quantum-classical workflow to identify and clean noisy quantum probability distributions post-measurement, similar to the results in [36]. This would save valuable NISQ hardware resources for more complex algorithms. It is evident that machine learning approaches can reduce or eliminate the need for costly quantum error correction routines and compliment other methods of error mitigation, providing a new pathway for quantum computing researchers to reach more useful conclusions about the results of their real-world experiments.

5.1 Future Work

Due to the highly black-boxed nature of neural networks, it is difficult to gain insight into the patterns the model learns or extract the learned trends from the model. It is clear from the results that the noise in quantum computers is not completely random; there is some deterministic pattern which is carried into the ML model through the training process and which becomes part of the model used for classification. This information could possibly be reflected in the network's edge weights. Patterns in the trained model's weights may provide insight to the corruption that the noise causes to the experiment and point toward how to best separate this noise from the ideal probability distribution. The problem of separating out the noise is similar to separating a mixture of probability distributions, and this strategy has not been explored in existing research in quantum error mitigation.

While the efficacy of simple multi-label classification models in accurately predicting the solutions to noisy quantum applications is demonstrated here, more specialized and complex classifiers are worth exploring to build upon this work. More complex models may be able to make more confident predictions in the difficult-to-classify

experiments with high noise. The dataset developed in this work also demonstrates bias toward a few categories due to the solutions to the problems used to develop it. An enhanced dataset would have approximately equal quantities of experiments with each possible solution, and could be expanded to contain more non-Grover's Algorithm basis-encoded problems.

Another enhancement to the dataset would be the inclusion of other characteristics of each probability distribution, such as shape parameters like skewness or kurtosis. This could be coupled with characteristics which contrast the information contained in two probability distributions, such as Hellinger distance or Kullback-Leibler divergence. This information could be used as an additional training weight or a separate input category in addition to the measurement probabilities.

Bibliography

[1] J. Preskill, "Quantum Computing in the NISQ era and beyond," *Quantum*, vol. 2, p. 79, Aug. 2018. [Online]. Available: https://doi.org/10.22331/q-2018-08-06-79

[2] E. Knill, R. Laflamme, and L. Viola, "Theory of quantum error correction for general noise," *Physical Review Letters*, vol. 84, no. 11, pp. 2525–2528, mar 2000. [Online]. Available: https://doi.org/10.1103/physrevlett.84.2525

[3] A. Rolander, A. Kinos, and A. Walther, "Quantum error correction in the noisy intermediate-scale quantum regime for sequential quantum computing," *Physical Review A*, vol. 105, no. 6, jun 2022. [Online]. Available: https://doi.org/10.1103/physreva.105.062604

[4] N. Cao, J. Lin, D. Kribs, Y.-T. Poon, B. Zeng, and R. Laflamme, "Nisq: Error correction, mitigation, and noise simulation," 2021. [Online]. Available: https://arxiv.org/abs/2111.02345

[5] S. Sponar. Quantum state tomography. [Online]. Available: http://www.neutroninterferometry.com/background/quantum-state-tomography

[6] Y. Hasegawa, R. Loidl, G. Badurek, S. Filipp, J. Klepp, and H. Rauch, "Evidence for entanglement and full tomographic analysis of bell states in a single-neutron system," *Phys. Rev. A*, vol. 76, p. 052108, Nov 2007. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevA.76.052108

[7] I. Qiskit. Grovers algorithm. [Online]. Available: https://github.com/Qiskit/textbook/blob/main/notebooks/ch-algorithms/grover.ipynb

[8] J. C. Patterson, "Managing a real-time massively-parallel neural architecture," Ph.D. dissertation, The University of Manchester, 01 2012.

[9] Y. Liao, D. Ebler, F. Liu, and O. Dahlsten, "Quantum speed-up in global optimization of binary neural nets," *New Journal of Physics*, vol. 23, no. 6, p. 063013, jun 2021. [Online]. Available: https://dx.doi.org/10.1088/1367-2630/abc9ef

[10] MathWorks. Multilabel image classification using deep learning. [Online]. Available: https://www.mathworks.com/help/deeplearning/ug/multilabel-image-classification-using-deep-learning.html

[11] A. R. Haverly, "A comparison of quantum algorithms for the maximum clique problem," Master's thesis, Rochester Institute of Technology, May 2021.

[12] F. Arute *et al.*, "Quantum supremacy using a programmable superconducting processor," *Nature*, vol. 574, no. 7779, pp. 505–510, oct 2019. [Online]. Available: https://doi.org/10.1038/s41586-019-1666-5

[13] Y. Wu *et al.*, "Strong quantum computational advantage using a superconducting quantum processor," *Phys. Rev. Lett.*, vol. 127, p. 180501, Oct 2021. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.127.180501

[14] "Ibm unveils 400 qubit-plus quantum processor and next-generation ibm quantum system two," Nov 2022. [Online]. Available: https://newsroom.ibm.com/2022-11-09-IBM-Unveils-400-Qubit-Plus-Quantum-Processor-and-Next-Generation-IBM-Quantum-System-Two

[15] F. Leymann and J. Barzen, "The bitter truth about gate-based quantum algorithms in the nisq era," *Quantum Science and Technology*, vol. 5, no. 4, p. 044007, sep 2020. [Online]. Available: https://dx.doi.org/10.1088/2058-9565/abae7d

[16] G.-B. Huang, Y.-Q. Chen, and H. Babri, "Classification ability of single hidden layer feedforward neural networks," *IEEE Transactions on Neural Networks*, vol. 11, no. 3, pp. 799–801, 2000.

[17] L. K. Grover, "A fast quantum mechanical algorithm for database search," in *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, ser. STOC '96. New York, NY, USA: Association for Computing Machinery, 1996, p. 212–219. [Online]. Available: https://doi.org/10.1145/237814.237866

[18] ——, "Quantum mechanics helps in searching for a needle in a haystack," *Phys. Rev. Lett.*, vol. 79, pp. 325–328, Jul 1997. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.79.325

[19] H.-S. Zhong *et al.*, "Quantum computational advantage using photons," *Science*, vol. 370, no. 6523, pp. 1460–1463, 2020. [Online]. Available: https://www.science.org/doi/abs/10.1126/science.abe8770

[20] M. Born, "Zur quantenmechanik der stoßvorgänge," *Zeitschrift für Physik*, vol. 37, no. 12, pp. 863–867, Dec 1926. [Online]. Available: https://doi.org/10.1007/BF01397477

[21] M. Weigold, J. Barzen, F. Leymann, and M. Salm, "Data encoding patterns for quantum computing," in *Proceedings of the 27th Conference on Pattern Languages of Programs*, ser. PLoP '20. USA: The Hillside Group, 2022.

[22] V. P. Belavkin, "Quantum noise, bits and jumps: Uncertainties, decoherence, trajectories and filtering," 2005.

[23] P. W. Shor, "Scheme for reducing decoherence in quantum computer memory," *Phys. Rev. A*, vol. 52, pp. R2493–R2496, Oct 1995. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevA.52.R2493

[24] Z. Cai, R. Babbush, S. C. Benjamin, S. Endo, W. J. Huggins, Y. Li, J. R. McClean, and T. E. O'Brien, "Quantum error mitigation," *Rev. Mod. Phys.*, vol. 95, p. 045005, Dec 2023. [Online]. Available: https://link.aps.org/doi/10.1103/RevModPhys.95.045005

[25] D. Fernandes and I. Dutra, "Using grover's search quantum algorithm to solve boolean satisfiability problems: Part i," *XRDS*, vol. 26, no. 1, p. 64–66, sep 2019. [Online]. Available: https://doi.org/10.1145/3358251

[26] D. Fernandes, C. Silva, and I. Dutra, "Using grover's search quantum algorithm to solve boolean satisfiability problems, part 2," *XRDS*, vol. 26, no. 2, p. 68–71, nov 2019. [Online]. Available: https://doi.org/10.1145/3368085

[27] S. A. Cook, "The complexity of theorem-proving procedures," in *Proceedings of the Third Annual ACM Symposium on Theory of Computing*, ser. STOC '71. New York, NY, USA: Association for Computing Machinery, 1971, p. 151–158. [Online]. Available: https://doi.org/10.1145/800157.805047

[28] M. Fürer, "Solving np-complete problems with quantum search," in *LATIN 2008: Theoretical Informatics*, E. S. Laber, C. Bornstein, L. T. Nogueira, and L. Faria, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 784–792.

[29] S. Agatonovic-Kustrin and R. Beresford, "Basic concepts of artificial neural network (ann) modeling and its application in pharmaceutical research," *Journal of Pharmaceutical and Biomedical Analysis*, vol. 22, no. 5, pp. 717–727, 2000. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0731708599002721

[30] G. M. D'Ariano, M. G. A. Paris, and M. F. Sacchi, "Quantum tomography," 2003.

[31] J. B. Altepeter, D. F. James, and P. G. Kwiat, *4 Qubit Quantum State Tomography*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 113–145. [Online]. Available: https://doi.org/10.1007/978-3-540-44481-7_4

[32] G. Torlai, G. Mazzola, J. Carrasquilla, M. Troyer, R. Melko, and G. Carleo, "Neural-network quantum state tomography," *Nature Physics*, vol. 14, no. 5, pp. 447–450, May 2018. [Online]. Available: https://doi.org/10.1038/s41567-018-0048-5

[33] S. Ahmed, C. Sánchez Muñoz, F. Nori, and A. F. Kockum, "Classification and reconstruction of optical quantum states with deep neural networks," *Phys. Rev. Res.*, vol. 3, p. 033278, Sep 2021. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevResearch.3.033278

[34] T. Schmale, M. Reh, and M. Gärttner, "Efficient quantum state tomography with convolutional neural networks," *npj Quantum Information*, vol. 8, no. 1, p. 115, Sep 2022. [Online]. Available: https://doi.org/10.1038/s41534-022-00621-4

[35] S. Yu, F. Albarrán-Arriagada, J. C. Retamal, Y.-T. Wang, W. Liu, Z.-J. Ke, Y. Meng, Z.-P. Li, J.-S. Tang, E. Solano, L. Lamata, C.-F. Li, and G.-C. Guo, "Reconstruction of a photonic qubit state with reinforcement learning," *Advanced Quantum Technologies*, vol. 2, no. 7-8, p. 1800074, 2019. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/qute.201800074

[36] P. S. Mundada, A. Barbosa, S. Maity, Y. Wang, T. Merkh, T. Stace, F. Nielson, A. R. Carvalho, M. Hush, M. J. Biercuk, and Y. Baum, "Experimental benchmarking of an automated deterministic error-suppression workflow for quantum algorithms," *Physical Review Applied*, vol. 20, no. 2, Aug. 2023. [Online]. Available: http://dx.doi.org/10.1103/PhysRevApplied.20.024034

[37] P. D. Nation, H. Kang, N. Sundaresan, and J. M. Gambetta, "Scalable mitigation of measurement errors on quantum computers," *PRX Quantum*, vol. 2, p. 040326, Nov 2021. [Online]. Available: https://link.aps.org/doi/10.1103/PRXQuantum.2.040326

[38] E. R. Bennewitz, F. Hopfmueller, B. Kulchytskyy, J. Carrasquilla, and P. Ronagh, "Neural error mitigation of near-term quantum simulations," *Nature Machine Intelligence*, vol. 4, no. 7, pp. 618–624, Jul 2022. [Online]. Available: https://doi.org/10.1038/s42256-022-00509-0

[39] M. Heydarian, T. E. Doyle, and R. Samavi, "Mlcm: Multi-label confusion matrix," *IEEE Access*, vol. 10, pp. 19 083–19 095, 2022.

[40] Qiskit contributors, "Qiskit: An open-source framework for quantum computing," 2023.

[41] M. Abadi *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/

[42] F. Chollet *et al.*, "Keras," https://keras.io, 2015.

[43] Rochester Institute of Technology, "Research computing services," 2022. [Online]. Available: https://www.rit.edu/researchcomputing/

[44] A. Crimmins, "Efficient quantum multiplication in the quantum fourier transform domain," Master's thesis, Rochester Institute of Technology, May 2024.

[45] B. A. Wrighter, "Improved grover's implementation of quantum binary neural networks," Master's thesis, Rochester Institute of Technology, May 2023.

[46] M.-L. Zhang and Z.-H. Zhou, "A review on multi-label learning algorithms," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 8, pp. 1819–1837, 2014.

[47] R. B. Pereira, A. Plastino, B. Zadrozny, and L. H. Merschmann, "Correlation analysis of performance measures for multi-label classification," *Information Processing & Management*, vol. 54, no. 3, pp. 359–369, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306457318300165