BACHELOR OF SCIENCE IN COMPUTER SCIENCE AND ENGINEERING

Anomaly Detection in IoT Devices Using Quantum Machine Learning

Shaikh Faiyaz Karim
190041116
Sajid Ahmed Chowdhury
190041140
Md. Shahriar Islam Bhuiyan
190041124

Department of Computer Science and Engineering
Islamic University of Technology
June, 2024

Declaration of Candidate

This is to certify that the work presented in this thesis is the outcome of the analysis and experiments carried out by **Shaikh Faiyaz Karim**, **Sajid Ahmed Chowdhury**, and **Md. Shahriar Islam Bhuiyan** under the supervision of **Dr. Md. Sakhawat Hossen**, Associate Professor, Department of Computer Science and Engineering and co-supervision of **Ali Abir Shuvro**, Lecturer, Department of Computer Science and Engineering, Islamic University of Technology, Dhaka, Bangladesh. It is also declared that neither this thesis nor any part of it has been submitted anywhere else for any degree or diploma. Information derived from the published and unpublished work of others have been acknowledged in the text and a list of references is given.

Dr. Md. Sakhawat Hossen
Associate Professor
Department of Computer Science and Engineering
Islamic University of Technology (IUT)
Date: June 04, 2024

Ali Abir Shuvro
Lecturer
Department of Computer Science and Engineering
Islamic University of Technology (IUT)
Date: June 04, 2024

Shaikh Faiyaz Karim
Student ID: 190041116
Date: June 04, 2024

Sajid Ahmed Chowdhury
Student ID: 190041140
Date: June 04, 2024

Md. Shahriar Islam Bhuiyan
Student ID: 190041124
Date: June 04, 2024

Dedicated to Our Parents

Contents

List of Figures

List of Tables

Acknowledgement

We are profoundly grateful to our supervisor, Dr. Md. Sakhawat Hossen, Associate Professor at the Islamic University of Technology, for his endless patience, insightful critiques, and for always believing in us even in the face of adversity.

A special thanks to our co-supervisor, Mr. Ali Abir Shuvro, Lecturer at the Islamic University of Technology, who provided invaluable support and guidance whenever the need arose. His dedication to helping guide us along our research journey.

Finally, we would like to thank our parents for their unconditional love and support. Thank you for always being there, even when times were tough.

To all of you, we extend our heartfelt appreciation and ask your prayers as we go onwards on the next stage of our journey.

Abstract

The adoption of the Internet of Things in our daily lives has led to an explosion of data. As we stay connected our data is exposed in various networks which can be a huge security concern. It is important to effectively and efficiently detect these anomalies and act accordingly. However, traditional machine learning techniques struggle to process and analyze this voluminous data efficiently. Quantum Machine Learning (QML) offers a promising solution by leveraging its quantum benefits of exponential speedup through superposition. This paper explores the application of QML for anomaly detection in IoT networks, highlighting its potential to significantly improve detection accuracy and speed. Ultimately the goal of this research is to build a foundation for quantum machine learning algorithms showing its potential equipping us for the future of big data.

Chapter 1

Introduction

1.1 IoT Anomaly And Big Data

The present world is all about connectivity. Everything is connected today starting from our phones, smart homes, automobiles, industries etc.[6] At the heart of this connectivity is the Internet of Things. This gives rise to a new concern of security. These IoT devices may be compromised by different attacks like security breaches, network disruption, Distributed Denial of Service (DDOS) etc.[14] The impacts of these anomalies range from data theft to being a threat to life when IoT devices in healthcare or transportation are compromised.

Considering the issue of anomaly it is imminent to detect these anomalies. Several attempts are being made using machine learning and deep learning to counter these anomalies[4]. However, with the immense growth of big data and IoT data classical computation takes a significant amount of time to process. The situation will only get worse as big data has been growing exponentially. To keep up with this growth of data and become future-proof quantum computing is needed. Quantum computing has traits such as superposition and interference which allows for a computational speedup required to accommodate the growth of big data.

1.2 Quantum Computing And Its Advantages

Quantum Computing is performing a computational task using the perks of quantum mechanics. In quantum computing information is stored in quantum bits also known as qubits. Unlike regular bits which store the data in either one of the two states (0 or

1), a qubit can store multiple states at the same time. This is the superposition nature of the quantum bit. As a result of superposition, quantum computing has an inherent parallelism advantage. This sets up the stage for an exponential speedup in computation by quantum computing.

Due to superposition, quantum computing requires fewer resources than classical bits to store the same information. Quantum computing also offers other advantages such as entanglement and interference. This offers lower time complexity compared to classical computing. The boom of quantum computing in the field of Machine learning gives us a new domain of computation. One which can prepare us for the ever-growing big data of the Internet of Things.

1.3 Problem Formulation

The goal is to detect anomalies in IoT by utilizing Quantum Machine Learning (QML) algorithms. Specifically, it is needed to design and fine-tune a Quantum Circuit that is optimized for Quantum Machine Learning and noise mitigated for IoT data.

The Quantum Machine Learning circuit should be well-performing on standardized datasets. For this experiment, testing was done on the KDDCup99 dataset [23], which is widely used for IoT anomaly detection studies. Further experiments were done on various well-known anomaly datasets such as AnoML-IoT[10], Environmental Sensor Telemetry Data, TWTDUS (Texax Wind Turbine) dataset by NREL, IoT_Fridge dataset of the New Generation Dataset of IoT and IIoT for Data-driven Intrusion Detection Systems[1] etc. Testing on various datasets made the QML algorithm more generalized.

Chapter 2

Related Works

2.1 IoT Attacks

IoT devices are cost-effective and easily set up to form a network. This often leaves it vulnerable to various attacks. These attacks range from low-grade attacks to high-intensity attacks [9] that pose a threat to the secrecy of the network. The impacts of these attacks include data theft, unauthorized access, safety risks etc.

One A common type of attack that is often faced by vulnerable IoT devices is Botnet attacks. Once infected with such malware, computers continually search the internet for vulnerable IoT devices and then use known default usernames and passwords to log in, infecting them with malware. These devices were things like digital cameras and DVR players. Examples of Botnet malware are mirai, spike, kaiten etc. Distributed denial of service (DDoS) is caused by these attacks.

2.2 Classical Algorithms

Some notable reasons for IoT devices being attacked are default login credentials' vulnerability [11] unprotected internet protocol which can be accessed easily using Zmap and Nmap information. These attacks also lead to power drainage and abnormality in the devices. Abnormalities in devices such as transport (cars and trucks) and healthcare systems (cardiac devices) may pose serious risks. So it is imminent to find ways to detect these anomalies.

With the advent of big data and the growth of the internet of Things (IoT) many industrial Internet of things (IIoT) devices were also introduced. These IIoTs are also vulnerable to attacks. Several attempts have been made to counter these attacks.

Among these [28] discusses an ensemble model using the Long Short Term Memory (LSTM) Autoencoder architecture is notable. In this method, anomaly detection is made where cyber threat hunting is done due to imbalanced datasets. Two such imbalanced datasets are the secure water treatment (SWaT) [7] and the Gas pipeline system [25] dataset.

The LSTM Autoencoder is used to train the most important features. Due to the nature of the LSTM long-term patterns can be taken into consideration. The input (before encoding) and output (after decoding) are compared. If the difference is significant then it indicates a high reconstruction error resulting in anomaly detection. The proposed model had an accuracy ranging from 90-94 per cent on the datasets depending on the number of LSTM layers. There is still scope for improving the accuracy and for a large enough dataset the model takes longer time to train. Considering the exponential growth of IoT data this is a concern to be addressed.

Another study [19] did something similar where the LSTM model was used for anomaly detection by feature extraction using graph convolution. The detection algorithm was tested on UNSWNB15 and KDDCUP99 Dataset [23], two very well-known datasets for anomaly classification for network security.
The goal for this proposal was to optimize network security and was effective in extracting spacial and temporal data using the convoluted graphs. However, creating a topological graph for a large enough network or dataset takes a huge toll on resources. So an effective way of representation of the states is to be found.

2.3 Quantum Algorithms

One common issue that is noticed in the classical models is the constraint on the resources and how it slows down when the dataset is large enough. This problem is only about to get more prominent with the increase of big data. To find a solution to this problem quantum computer with its inherent properties of superposition, entanglement and interference can be used. A review case study [29] explains in detail how these concepts of quantum mechanics can be applied for machine learning i.e. quantum machine learning (QML).

Quantum machine learning algorithms work with quantum bits (qubits). So the classical bits first need to be encoded into qubits. There are many encoding methods such as amplitude encoding, basis encoding, angular encoding and higher-order encoding. The basic computation in quantum machine learning is done in quantum circuits with several gates which are a series of matrix multiplication. When the gates are

parameterized with the input values then a probability distribution is given for that value. Afterwards, we decode to classical bit for evaluation.

The review paper [29] also discusses some fundamental quantum machine learning algorithms such as variational quantum circuit (VQC), quantum support vector machine (QSVM), quanvolutional neural network (QNN) and a comparative analysis was done with the classical counterparts. The performance of the QML models was promising and close to the classical models if not better.

Another paper [17] gives an analysis of the QML algorithm in noisy channels for the classification of IoT devices. In this proposal, they use 3 algorithms namely, UU-Dagger, Variational Quantum Circuit, and QNN and compare with simple classical algorithms such as linear regression and logistical regression. They worked on an unsupervised dataset and got better accuracy using the QML algorithm. This shows a lot of potential for QML algorithms in the future. However, linear regression is a very basic algorithm and a more advanced classification algorithm should be considered to set a benchmark.

2.3.1 Expressibility and Entangling Capability of Parameterized Quantum Circuits

Quantum circuits traverse the Bloch Sphere. This is especially true for variational quantum circuits, or VQCs for short. The paper [20] describes different ansatz circuits about different VQCs and how they manage to traverse the Bloch Sphere and achieve different levels of expressibility. A visual representation can be seen in figure 2.1. A balance has to be achieved between high and low expressibility, as training time can be increased exponentially in the search for good expressibility.

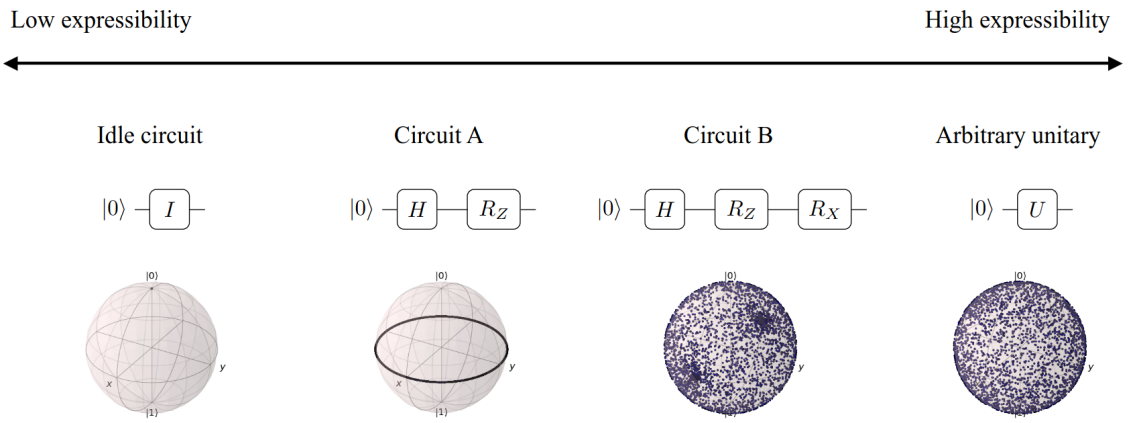

Figure 2.1: Expressibility Chart

2.3.2 A Comparison of Various Classical Optimizers For Variational Quantum Circuits

VQCs work based on optimisers, which are responsible for fine-tuning parameters. This study [15] tested out different optimisers over different scenarios. They found out that for most average case scenarios, COBYLA performed best whilst optimisers like ADAM lagged significantly. Another paper [21] discusses ways of finding the optimal gradient by setting certain conditions from the begining.

2.3.3 Finding The Optimal Feature Map and Ansatz

Nowadays, many studies have focused on finding the optimal feature map and ansatz for VQCs, as the optimal circuit configuration would result in the best classification performance possible whilst greatly reducing the time needed to train the VQC. The articles in [8], [24] and [5] discuss different strategies for finding the optimum ansatz and feature map. [24], in particular, talks about a Pauli gate-based feature map that can be optimised, whilst [5] proposes an algorithm called QAS.

In the case of optimal ansatz design, [16] suggested looking at certain criteria to maximise: **accuracy** and **efficiency**. From the results in the paper, it is readily apparent that accuracy can be increased by reducing noise in the circuit and using better circuit configurations with less depth. In the case of efficiency, data pruning and certain optimisation models can help. More fine-tuning of these aspects will result in either a general solution or a specialised solution.

In the case of efficient circuit design, *VQE-generated Quantum Circuit Dataset for Machine Learning* talks about different quantum circuits that are good at classification tasks. Similar circuits have also been discussed in [20], which are shown in figures 2.2 and 2.3.

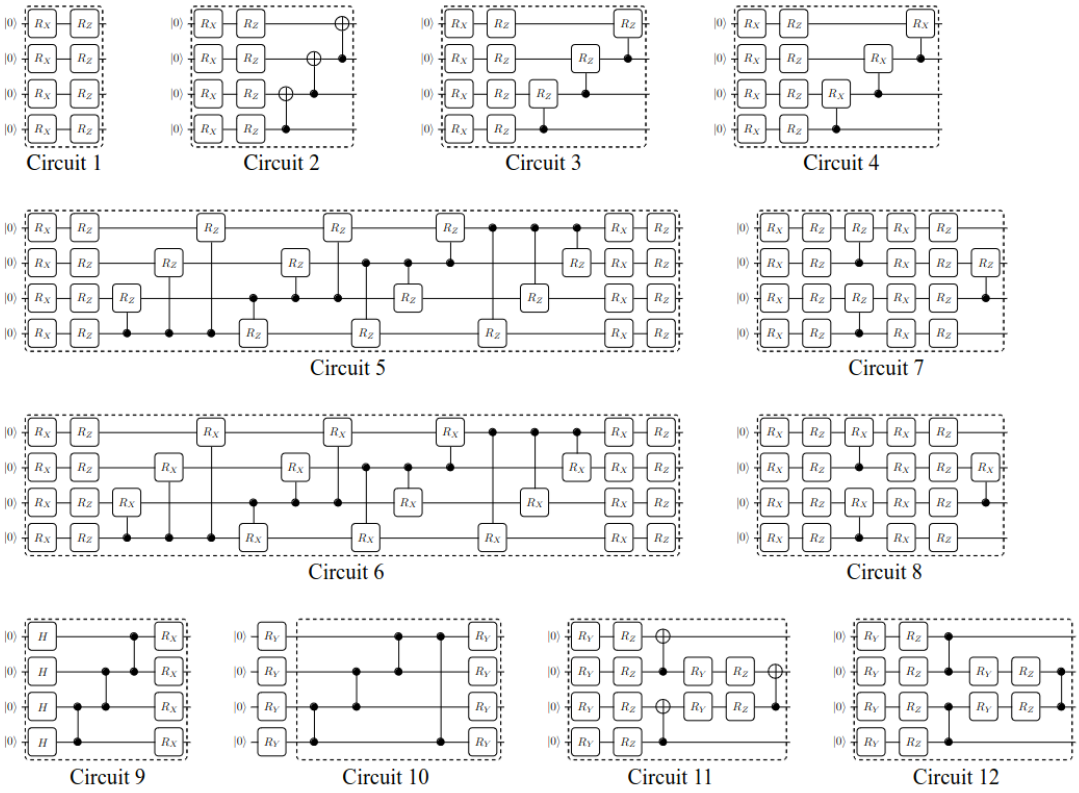

Figure 2.2: Popular Ansatz 1

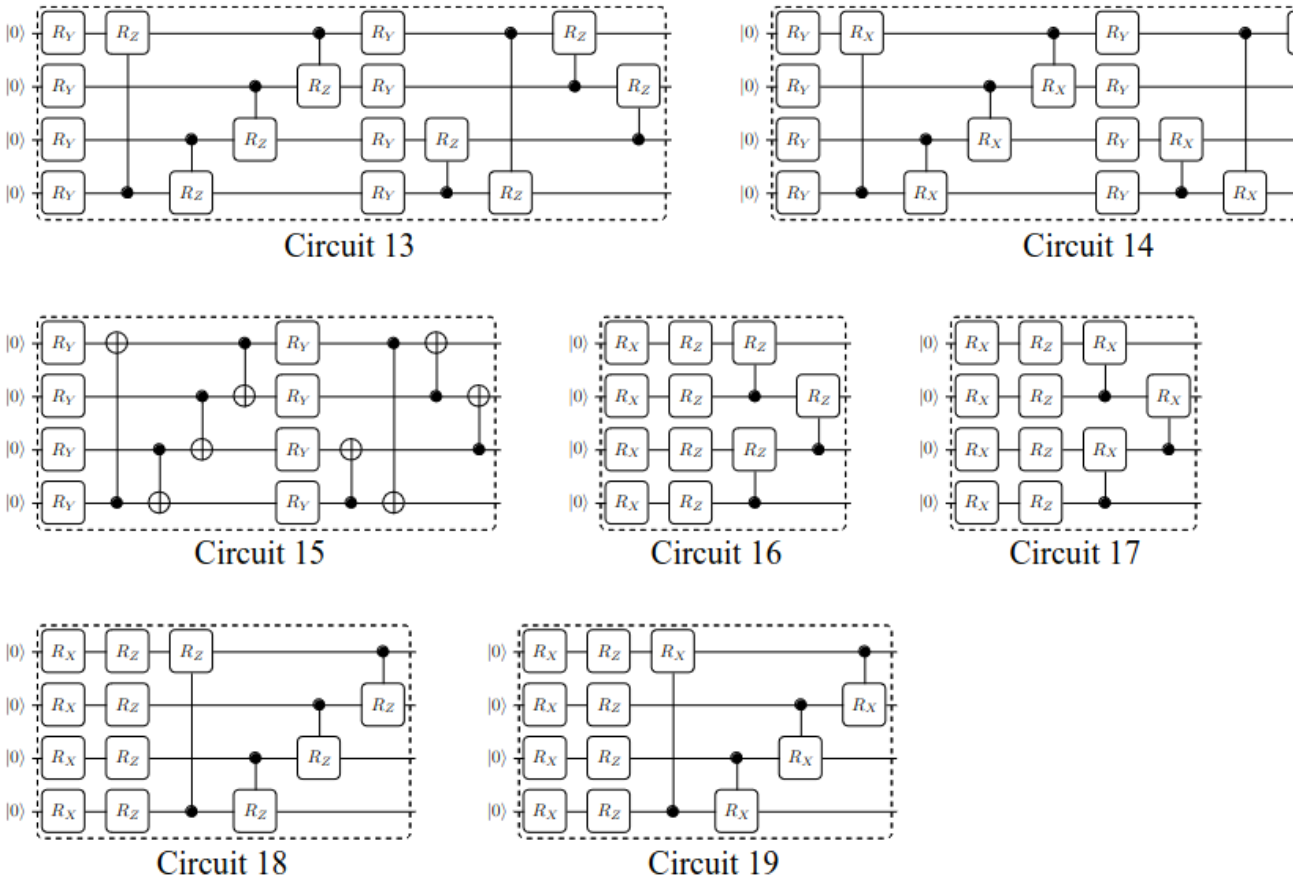

Figure 2.3: Popular Ansatz 2

Lastly, in the paper [2], the authors proposed an auto feature map generation algorithm based on a genetic algorithm. The overall flow of the algorithm is shown in figure 2.4.

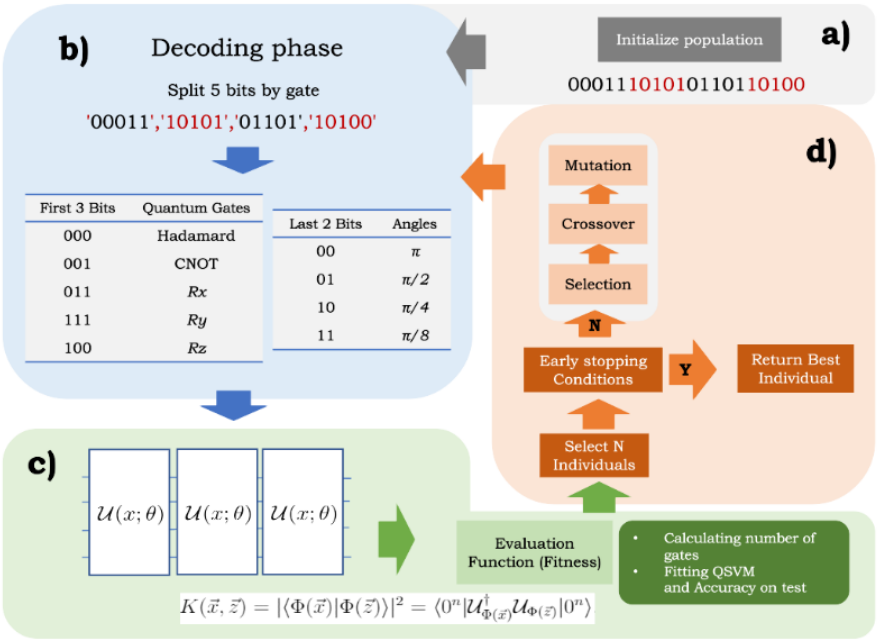

Figure 2.4: Genetic Algorithm From [2]

In this approach, a Support Vector Machine (SVM) is used on the dataset to extract meaningful features and transform them into a bitstream. Thus, the bitstream showcases the extracted features from the dataset in a digital form.

Based on the encoding scheme given in figure 2.4, the bitstream is mapped onto the gates of a quantum circuit. This helps to showcase the extracted features in the quantum space and also explains how each bit or sequence of bits in the bitstream corresponds to a particular quantum gate configuration along with its specific parameters.

After the mapping process comes the optimisation process. This is done using a genetic algorithm, where the best bitstreams are selected from the parent generation. These bitstreams are selected based on the criteria of classifying the input dataset correctly. The data from the parent bitstreams can then be propagated to the next generation. Additionally, bitstreams are also crossed over to get children bitstreams which will contain bit sequences from both parents. This is done to introduce the concept of inheritance and increase the scope for potential improvement in the overall circuit configuration. Since the genetic algorithm will work over several successive generations, the process will be iterative. This will allow the algorithm to eventually arrive at an optimal solution, where the bitstream showcases a quantum circuit that will act as the feature map for the given dataset. By putting more emphasis on quantum rotation gates, the genetic algorithm tries to find circuit configurations that contain more simple operations instead of complex operations like those carried out in entanglement

gates, such as CNOT gates.

Ultimately, the goal of the algorithm proposed here is to find an optimal feature map for the given dataset, which will help in the creation of the overall quantum circuit that will successfully classify data.

Chapter 3

Methodology

Analysis of the problem made us consider a quantum approach, where quantum circuits are used for the classification of unsupervised and supervised data. This was due to Quantum Computing being able to potentially handle complex data structures in comparison to classical computing.

The crux of our approach lies in encoding classical data to quantum data, where quantum gates come into play. The dimensionality of data also comes into play here, where data is converted from a lower dimension into a higher dimension in the quantum space. This helps in making the task of classification easier.

Using quantum algorithms, it is possible to tackle both supervised and unsupervised problems. However, preprocessing is needed to make the data suitable for undergoing these quantum algorithms. The better the preprocessing, the more effective our quantum algorithms will be. Techniques such as feature scaling and normalisation will play a big role in the preprocessing stage.

To get our results, we plan on employing a range of quantum algorithms which would employ different strategies to properly label data in a classification task. Figure 4.3 will give a better idea of the algorithms we plan to implement.

The outcomes from these quantum algorithms will help us to get a good idea about the effectiveness of quantum computing in classification tasks in terms of accuracy and computational efficiency. Comparisons with classical methods can also help us to better gauge the limitations of quantum computing in retrospect to classical computing in certain tasks.

As such, the goal here is to explore the effectiveness of quantum computing in classification tasks and provide results that back up said effectiveness.

3.1 Experiment Flow

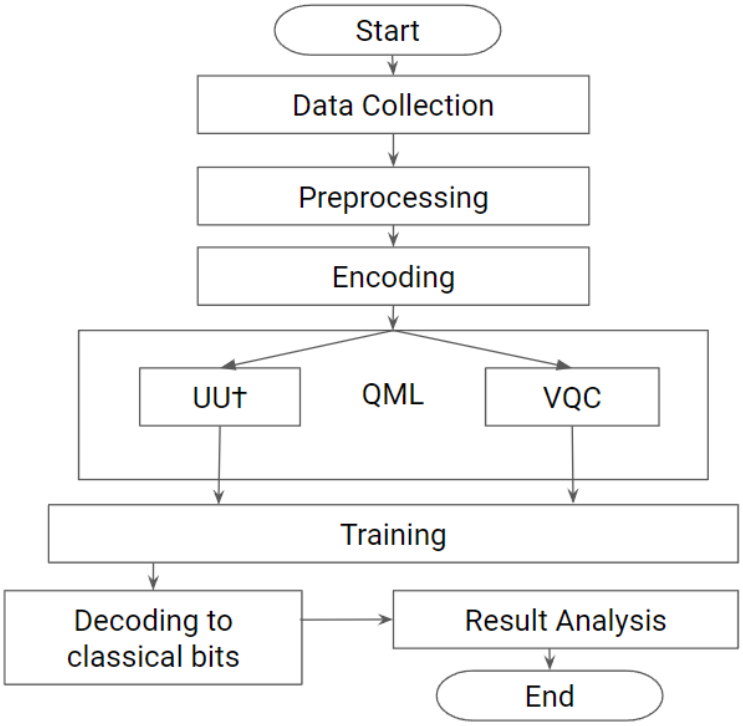

Figure 3.1: Flow chart of the experiment

Based on figure 4.3, we plan to collect data from a host of datasets and then carry out preprocessing on the collected data. Preprocessing is done using standardised feature selection methods available today such as forward selection and backward elimination. The preprocessed data is then encoded onto qubits, which are passed into different quantum circuits. The quantum circuits make up what is known as quantum machine learning models. We plan to work with models called UU-† and the Variational Quantum Circuit (VQC) [18]. Based on the quantum model chosen, the chosen dataset will be used to train the circuit. After training, the qubits will be decoded into classical bits for easier measurement. The decoded bits will have probabilities associated with them, which will be used in analysing the results of the quantum circuit. The results from the experiment could then be used in comparisons with classical models of the same type.

3.1.1 UU-† Circuit

Overview

The UU-† circuit is a very popular quantum circuit that works based on the age-old quantum bra-ket notation [17], which is used to find the dot product between two quantum states:

$$\langle\psi|\phi\rangle = |\psi\rangle \cdot |\phi\rangle$$

The cross-product between two different quantum states helps to find the probability of measuring a system in a given state. This is the modus operandi of the famous Born's Rule [22]:

$$P(\phi) = \langle\phi|\psi\rangle^2$$

In the case of the UU-† circuit, this property of computation allows the classification of data after data is encoded onto the $|0\rangle$ quantum state by passing them as parameters to a Unitary gate, which takes the form of a matrix:

The thresholds that we are measuring the data against are also encoded onto another quantum state by inputting their conjugate transposes as parameters to another Unitary gate. For each feature, we will need a qubit, resulting in the number of features being proportional to the number of qubits. This is the U-† gate. The block diagram for the UU-† circuit is given in figure 3.2.

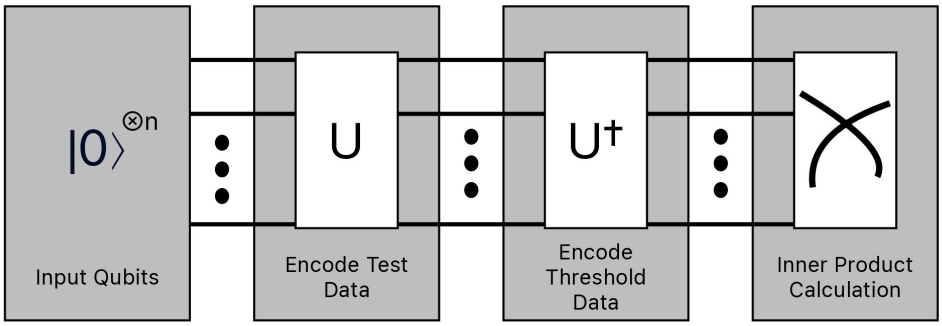

Figure 3.2: UU-† Block Diagram

When we measure the circuits, we will get probability values for an array of states. In the case of a two-qubit system, we can expect to get the distribution for the states: $00, 01, 10, 11$. As the starting qubits were $|0\rangle$, the probability of 00 will give us an estimation of how close the two states are to each other. This is useful in the case of classification because if we take features as parameters for the first U-gate and then take appropriate threshold values as parameters for the U-† gate, we will be able to

find out the probability of an entry being above or below a certain threshold value. For the threshold calculation, we used k-means to find the centroids of the dataset, which in turn acted as the parameters for the U-† gate. The equation for the UU-† circuit can be given as [2]:

$$\mathrm{Re}\langle\Phi(x)|\Phi(x')\rangle = \mathrm{Re}\langle 0^n|U(x;\theta)^\dagger U(x';\theta)|0^n\rangle$$

The gist of the matter is that the UU-† circuit showcases the most intrinsic value of quantum computation, that being the principle of superposition. As such, simplicity aside, the UU-† circuit gives us a good idea about the future of quantum computing.

3.1.2 Variational Quantum Circuit

Overview

The Variational Quantum Circuit (VQC) is the circuit that can be said to be a true machine learning model, where we will see the implementation of a circuit with trainable parameters. The VQC consists of three parts: the **encoder**, the **ansatz** and the **measurement unit**. The ansatz also consists of an **optimiser**, which is responsible for fine-tuning our parameters.

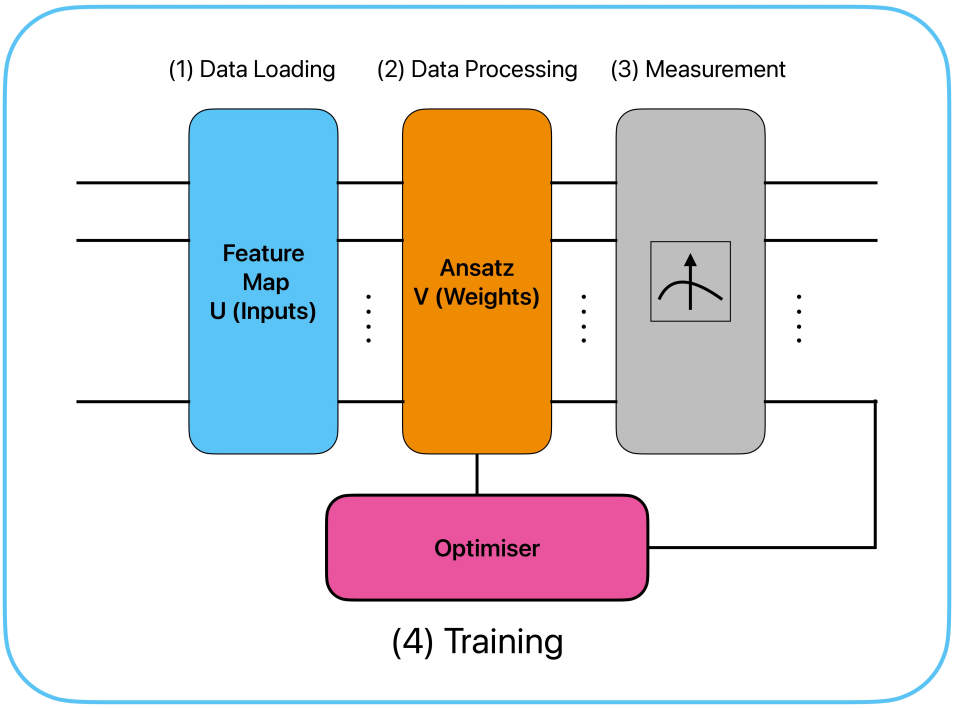

Figure 3.3: VQC Block Diagram

Phases

The VQC can be broken down into phases based on the block diagram shown in figure 3.3. The data loading phase consists of an encoder, better known as a feature map. The feature map will help in encoding the feature values to the input qubits, which in this case will be a set of $|0\rangle$ qubits for each feature value. The feature map can be created using pre-defined feature maps such as the ones given in the Qiskit library, with the **PauliFeatureMap** being a popular example. Another way would be to create feature maps based on the relationships inferred from the dataset. We planned on using a combination of both, with us relying heavily on customised versions of the **PauliFeatureMap** and also on creating our feature maps based on the algorithm discussed in [2]. A variety of gates are responsible for encoding the data to the qubits, with Pauli-gates, Hadamard gates and X-gates being focused on single qubit embedding. CNOT-gates, on the other hand, are responsible for higher-order encoding, which means that the state of one intermediate qubit will depend on the state of another intermediary qubit. In this way, feature values can be joined together so that more information can be extracted from them. Throughout the many encoding strategies used, we will focus mainly on angle encoding, which uses Rotation gates to apply x,y or z-plane rotations to the qubit.

The data processing phase consists of an ansatz, which is a circuit that can be repeated multiple times. The anstaz also consists of a combination of popular quantum gates with the CNOT gate being responsible for making use of one of the most powerful properties of quantum computing: quantum entanglement. Through quantum entanglement, we manage to traverse more of the 3-dimensional space of a quantum space, the Bloch Sphere as discussed in [20]. These gates, along with the Bloch Sphere, have been discussed in section 5.

The measurement section consists of classical registers, which are used to measure the data and hence allow us to obtain probability measurements for the different states. These states are binary and contain encoded information about the different features.

For the learning part of the model, we have to rely on reliable methods of parameter tuning to arrive at parameter values which work well for the dataset we will choose. We could say that this is the classical part of the mode, focusing heavily on classical optimisation techniques based on gradient descent. Some popular optimisers available today are ADAM, COBYLA, SLSQP, etc. For our experimentation, we decided on COBYLA due to the influence of the results of the study from [15].

For training, we created our quantum circuits by the methodology proposed in sec-

tion 3. Our two quantum circuits, shown in figures 4.6 and **??**, took in two qubits, q_0 and q_1 as inputs. The UU-† circuit then carried out two matrix multiplications on the input qubits via a unitary gate (U-gate) and a U-† gate, which was the conjugate transpose of the U-gate. The centroid data of the dataset was inputted into the U-† gates and this allowed the circuit to output classical bits on measurement with probabilities associated with them. The probabilities showed the likelihood of a particular input belonging to a certain class or label.

3.1.3 Evaluation Metrics

Accuracy

$$Accuracy = \frac{TruePositive + TrueNegative}{TruePositive + FalsePositive + TrueNegative + FalseNegative}$$

For the evaluation metrics, we decided to use accuracy to judge the number of labels that match the data from the supervised data. This is because accuracy is a good metric in determining the amount by how much the labels predicted by our models match the labels in the test set. As such, an accuracy score stands to provide a clear and straightforward representation of our results.

From the equation given above, it can be seen that an accuracy score takes into account both false positives and false negatives. This will help to alleviate concerns regarding such outlier values and make sure that a balanced assessment can be achieved. In the field of anomaly detection, such outliers can be paramount in classifying anomalies.

Generalization is another aspect that can be tested when we find out the accuracy score for our models. Another term for this would be robustness, which determines a model's ability to work well on test sets in addition to the training set. As discussed in [27], a mix between the two is always preferred. In our case, by finding out accuracy scores for both our training data and test data, we will be able to see how well the model works on data outside of the data it was trained on. The main goal here is to create a model that is both accurate and robust.

In the case of our evaluation, we used two frameworks. Firstly, we used the **VQC** class supplied by the **qiskit-machine-learning** library to carry out an accuracy measurement. This was done using the **score** method supplied by said class. A plus point of using this class is the fact that it is optimised for quantum circuits. For our second framework, we used the **accuracy_score** method from the **sklearn** library. This is one of the most popular methods of measuring accuracy, used broadly by researchers

in the field of traditional machine learning. All in all, both methods allowed us to set the benchmark in terms of accuracy in predicting labels from a wide array of datasets.

Chapter 4

Results and Discussion

4.1 Experiment

4.1.1 Environment And Configuration

Table 4.1: Environmental Specifications

Setup	Specifications
Operating System	Linux
Cloud Environment	IBM Quantum, Google Colab
Number of Qubits	3-7

Initially, we started working on IBM Quantum Lab, a cloud quantum environment provided by IBM. However, the project was sunset on May 15 due to IBM focusing on a more hands-off approach to quantum computing. This resulted in our thesis moving towards local environments and other cloud environments like Google Colab. Due to Qiskit being easy to set up in any environment, the change did not turn out to be that big of an issue.

4.2 Implementation

The algorithms discussed in section 3.1 have been implemented in this section, accompanied by circuit diagrams of our proposed circuits. The overall flow of the experiment follows the flow chart shown in section 3.1. The datasets used in our experiments have also been highlighted in this section.

4.2.1 Datasets

Datasets related to IoT devices and network traffic were considered for this research. The goal was to establish a foundation for quantum machine learning in anomaly detection. As such datasets with anomalies were considered a high priority. Many datasets were explored and experimented on to solidify the anomaly detection algorithm.

KDD CUP 1999 Data

This dataset [23] was used for The Third International Knowledge Discovery and Data Mining Tools Competition. The competition was held in conjunction with KDD-99 The Fifth International Conference on Knowledge Discovery and Data Mining. Since then KDDCUP99 has become one of the most widely recognized and used datasets for anomaly detection [12]. The KDDCUP 99 is a supervised data i.e. the outcome (label) is known to us. In general, the data is significantly large with 42 features and nearly 5,00,000 records.

Preprocessing:
To clean the data firstly unnecessary columns such as protocol_type, services, flag etc. were dropped. Then the label "normal" was replaced with 0 and any "anomaly" was replaced with 1. This was done as our goal was the detection of anomalies. There was still a huge amount of features and due to the lack of qubits, we needed to select the best feature. Feature selection using Mutual Information[26] on KDDCUP99 was done [3]. The top seven features with the descending Mutual Information score were selected as our training feature.

Table 4.2: Feature Selection Based on Mutual Information Score in KDDCUP99 Dataset

Feature Name	Mutual Information Score
src_bytes	0.444144
count	0.432390
dst_bytes	0.370577
dst_host_same_src_port_rate	0.273433
logged_in	0.265738
srv_count	0.246803
dst_host_count	0.205202

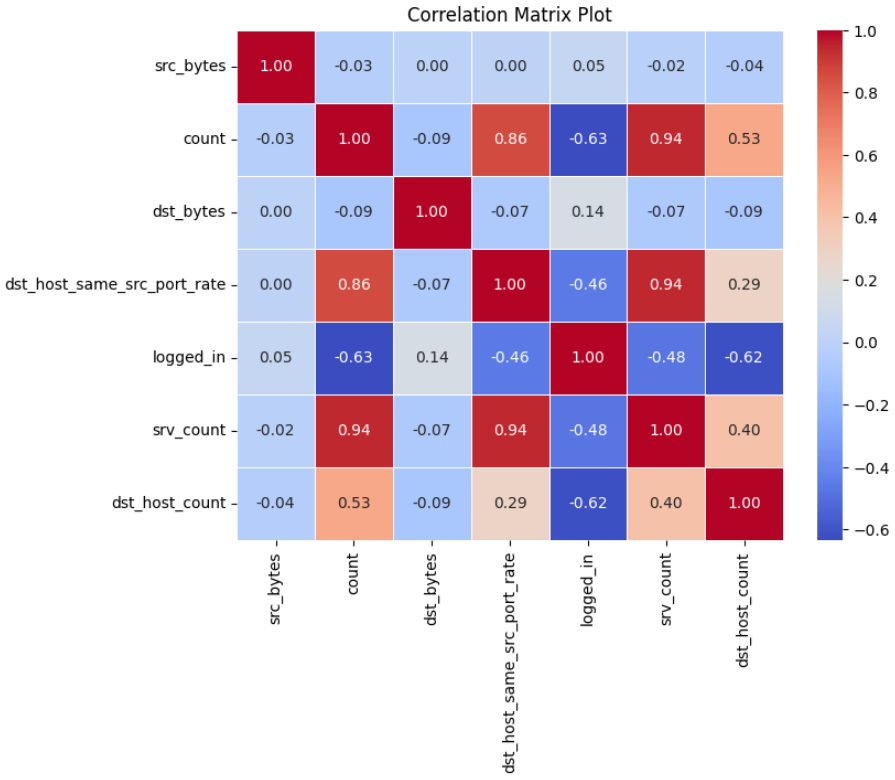

Figure 4.1: Correlation Matrix Plot of KDDCUP99 dataset

Then the label and the data were split and the train test split was done. Furthermore, the data was scaled and then encoded to quantum data with higher-order encoding.

AnoML-IoT

The anoML-IoT dataset[10] is a specialized time-series dataset designed for anomaly detection in Internet of Things (IoT) environments. The data was primarily collected from various sensor data, network traffic data, system logs of IoT devices etc. The anoML-IoT dataset is an unsupervised dataset having 5 features and over 6,500 records. Compared to the KDDCUP99[23] this is significantly smaller which helps the model to converge faster.

Table 4.3: Sample data of prominent features of anomML-IoT Dataset

Temperature	Humidity	Loudness	Light
37.94	28.94	106	644
37.94	29.00	145	645
37.88	28.88	146	644
37.72	28.94	139	646
37.69	29.19	155	644

Preprocessing:
anoML-IoT dataset contains information regarding the IoT devices such as Humidity, Temperature, Air Quality, Light and Loudness. For preprocessing, we dropped the time feature and found out the correlation matrix. This allowed for a smarter selection of features to train more efficiently.

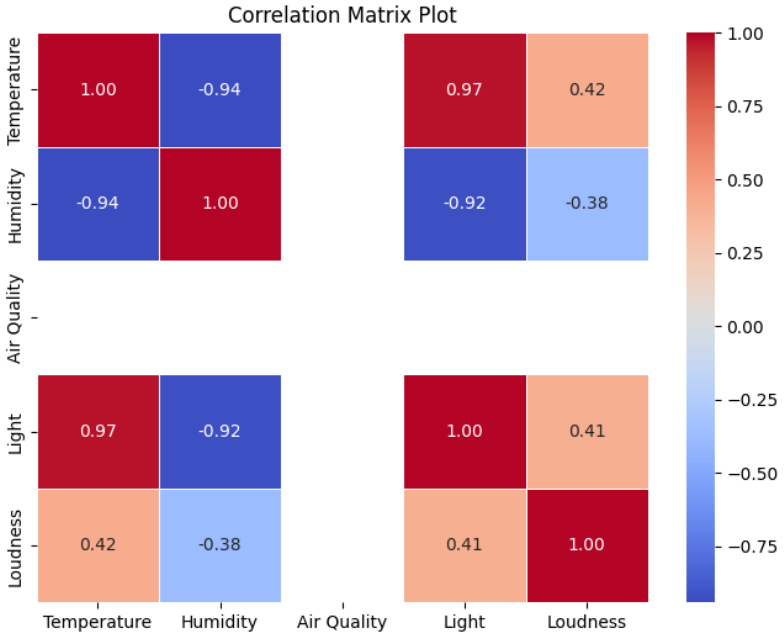

Figure 4.2: Correlation Matrix Plot of anoML-IoT dataset

Environmental Sensor Telemetry Data

The Environmental Sensor Telemetry Dataset was published on Towards Data Science. In the Environmental sensor telemetry dataset, data was generated from a series of three identical, custom-built, breadboard-based sensor arrays. Each of the three IoT devices was placed in a physical location where environmental conditions would vary. Each IoT device collected a total of seven different readings from the four sensors at regular intervals. Including device ID and timestamp, the number of features in the dataset was 9.

Table 4.4: Features along with their unique frequencies in the Environmental sensor telemetry dataset

Feature Name	Number of Unique Feature Values
ts	212739
device	3
co	4615
humidity	474
light	2
lpg	4611
motion	2
smoke	4810
temp	205

Table 4.5: Sample data of the prominent features in the telemetry dataset

co	humidity	light	lpg	smoke	temp
0.003230	77.099998	0	0.005613	0.014662	19.900000
0.005154	51.100000	0	0.007871	0.021040	22.000000
0.005769	49.500000	0	0.008540	0.022954	22.400000
0.005148	47.800000	0	0.007865	0.021021	23.100000
0.003782	52.299999	1	0.006292	0.016563	29.900000

Preprocessing:

During preprocessing of the time-series data, the time stamp i.e. ts and device column were dropped. The light and motion feature was labelled encoded to 1 and 0. The missing values were filled by the mean value. And finally, the correlation Matrix was found to get an idea about which features will be more beneficial to the training process.

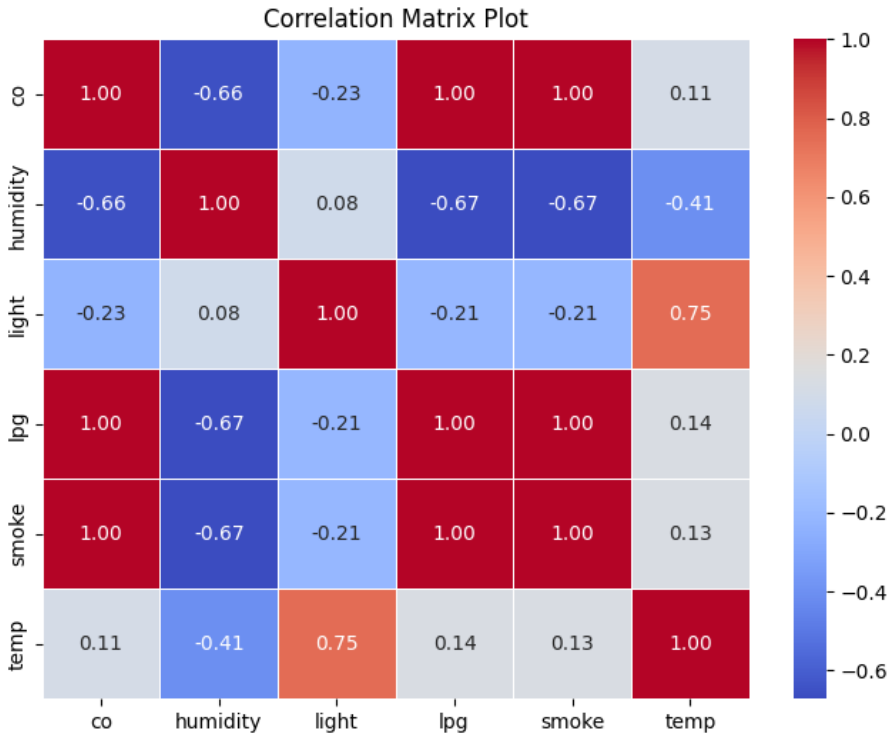

Figure 4.3: Correlation Matrix Plot of Environmental sensor telemetry dataset

TWTDUS dataset

The other dataset used for experimentation was the Texas Wind Turbine dataset. The dataset consisted of full-year hourly time-series data simulated using the National Renewable Energy Laboratory (NREL). The dataset consists of 6 features with nearly 9,000 records which can be analysed to be used as predictors in classification tasks and unsupervised learning.

Table 4.6: Sample of prominent features in TWTDUS dataset

System power generated (kW)	Wind speed (m/s)	Wind direction (deg)	Pressure (atm)	Air temperature (°C)
1766.64	9.926	128	1.000480	18.263
1433.83	9.273	135	0.999790	18.363
1167.23	8.660	142	0.999592	18.663
1524.59	9.461	148	0.998309	18.763
1384.28	9.184	150	0.998507	18.963

Table 4.7: Features along with their unique frequencies in the Texas Wind Turbine dataset

Feature Name	Number of Unique Feature Values
Time stamp	8760
System power generated | (kW)	7857
Wind speed | (m/s)	1433
Wind direction | (deg)	361
Pressure | (atm)	2173
Air temperature | ('C)	428

Preprocesssing:
Firstly, all the data regarding the time stamp was dropped and the records having no System power generated were also discarded. This was done as we wanted to examine only the active turbines. No further preprocessing was needed so the correlation Matrix to find the relationship between the features.

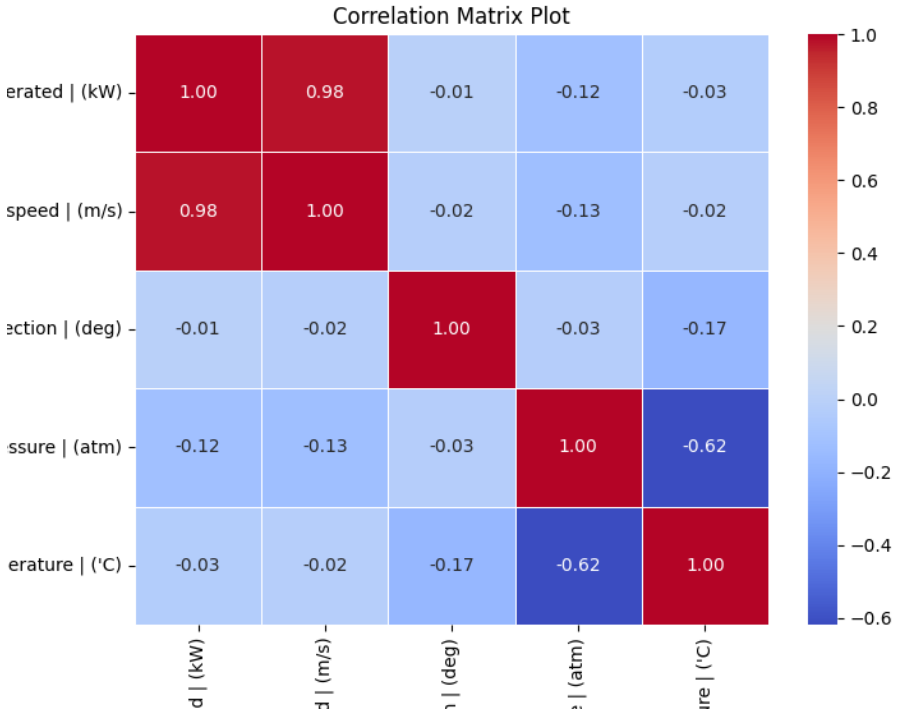

Figure 4.4: Correlation Matrix of the IoT_Fridge dataset

IoT_Fridge dataset

Finally, experiment was conducted on the IoT_Fridge dataset which is part of the New Generation Dataset of IoT and IIoT for Data-driven Intrusion Detection Sys-

tems[1]. This is a supervised time-series data for anomaly detection with close to 60,000 records. There is an extensive record of the timestamps and temperature conditions, labels and types of anomaly.

Table 4.8: Sample data for prominent features in IoT_fridge dataset

Fridge Temperature	Temp Condition	Label	Type
9.00	high	1	ddos
9.25	high	1	ddos
12.65	high	1	ddos
4.65	low	1	ddos
12.65	high	1	ddos

Table 4.9: Unique Value Counts of Features in IoT_Fridge dataset

Feature	Unique Value Count
ts	18910
fridge_temperature	151
temp_condition	6
label	2
type	7

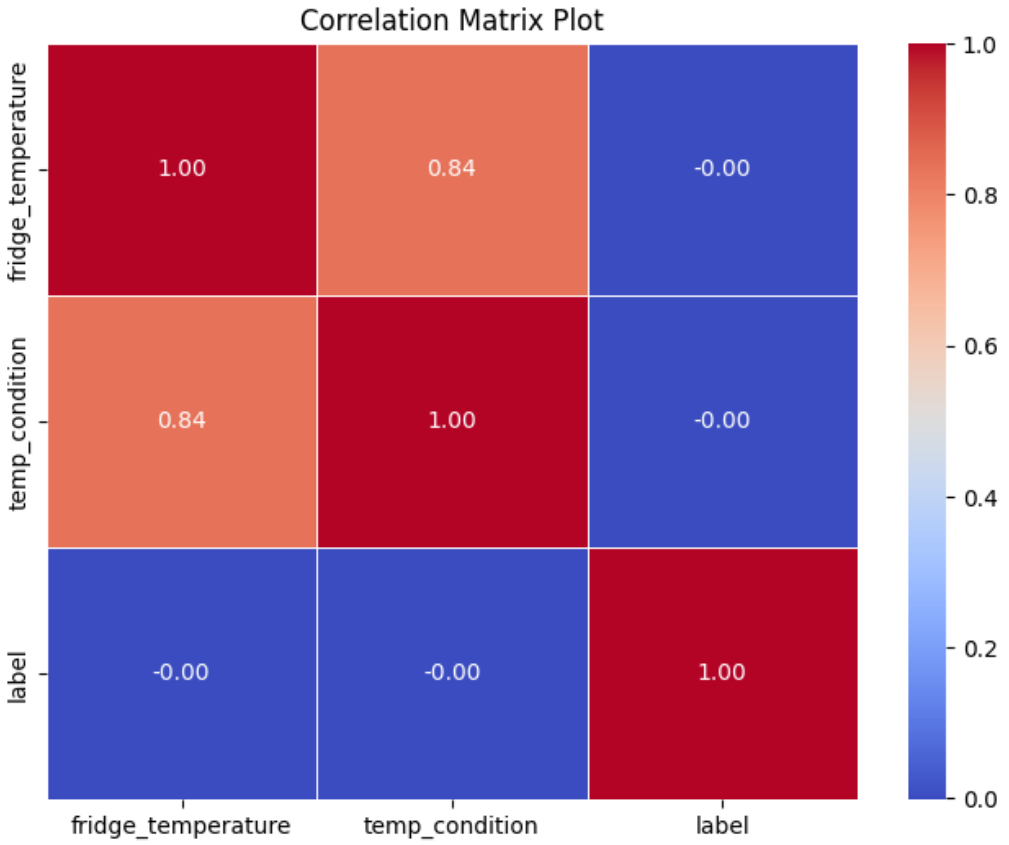

Figure 4.5: Correlation Matrix of the IoT_Fridge dataset

Preprocessing:
All the details regarding the timestamp, date, time etc. were dropped. The type of anomaly was also discarded. Temperature condition was label encoded and any missing value was handled. After cleaning the data we had only 2 features mainly to work with i.e. fridge_temperature and temp_condition. This proved to be disadvantageous as more features would have given us a better insight. The correlation matrix shows us the relationship between the features of the IoT_Fridge Dataset.

In most of these data feature scaling was done to scale the data in the range of $[0, \pi]$. This helped to normalise the data and make it easier for processing. In the case of supervised data we split the label and the feature as X and y and then train-test split was done accordingly. In the case of very large data such as the KDDCUP99 and Environmental Telemetry data which had over 4,90,000 and 4,00,000 records, appropriate down-sampling was done to make training easier and faster. The data were also converted to a numpy array before training to make the training faster and more efficient.

4.2.2 Circuits

UU-† Circuit

As stated before, the UU-† circuit is the representation of the basic quantum operation of the dot product. As such, here we encoded our feature values into U gates (U gate method in qiskit) and then entered the conjugate transpose of the centroid values in the succeeding U gates. Then we measure them using classical registers. Figure 4.6 shows the implementation of the UU-† circuit in qiskit. This implementation will remain the same regardless of the dataset being used, with the only change coming if more features are added. If more than two features are added, then an extra qubit will be added to the circuit for each feature.

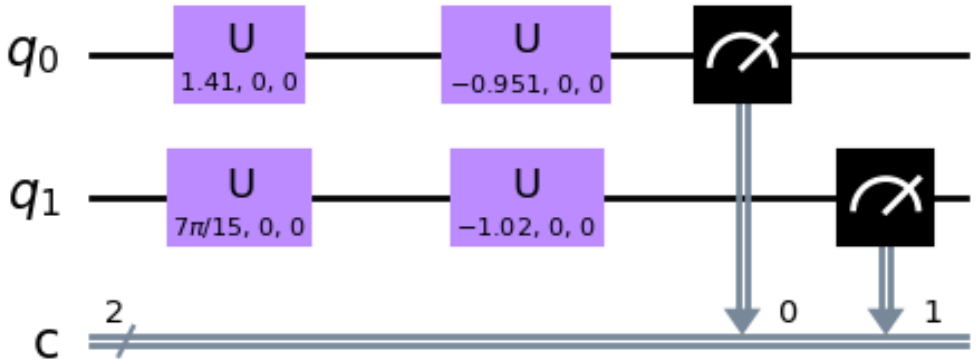

Figure 4.6: UU-† Circuit Diagram

Variational Quantum Circuit (VQC)

The parts of the VQC have already been discussed in section 3. For each of our datasets, we used a VQC, with there being two implementations for VQC for the KDD CUP dataset. The following parts of the section will showcase the circuits created during the experiment, with a brief explanation as to why we chose them.

VQC For IoT_Anomaly Dataset:
Figure 4.7 showcases the feature map of the VQC circuit used for the IoT Anomaly dataset, where 4 qubits are used to encode the feature information. The Pauli gates help to rotate the qubits with information about the features, whilst the CNOT gates help in entanglement. The circuit goes on until CNOT gates have been used between all qubit pairs. Between the CNOT gates, random rotations were carried out to traverse more of the Bloch Sphere.

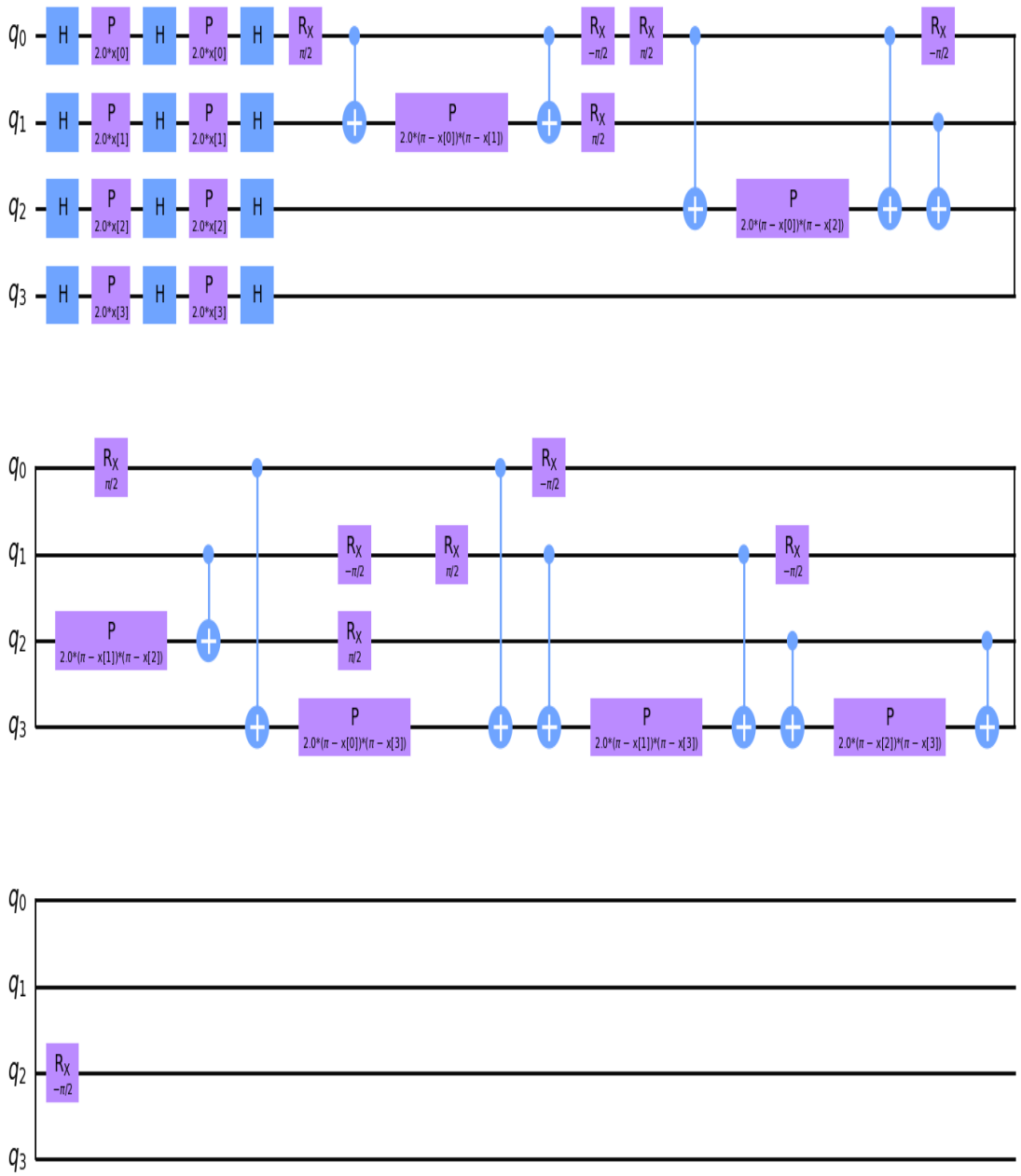

Figure 4.7: Feature Map For IoT Anomaly Dataset

Figure 4.8 an ansatz consisting of rotations via Pauli-Y gates was carried out on each qubit, followed by CNOT gates on all qubit pairs. This ansatz was repeated 4 times to get a proper depth to the circuit. This depth of 4 was found by testing the circuit until a middlepoint between expressibility and trainability was reached.

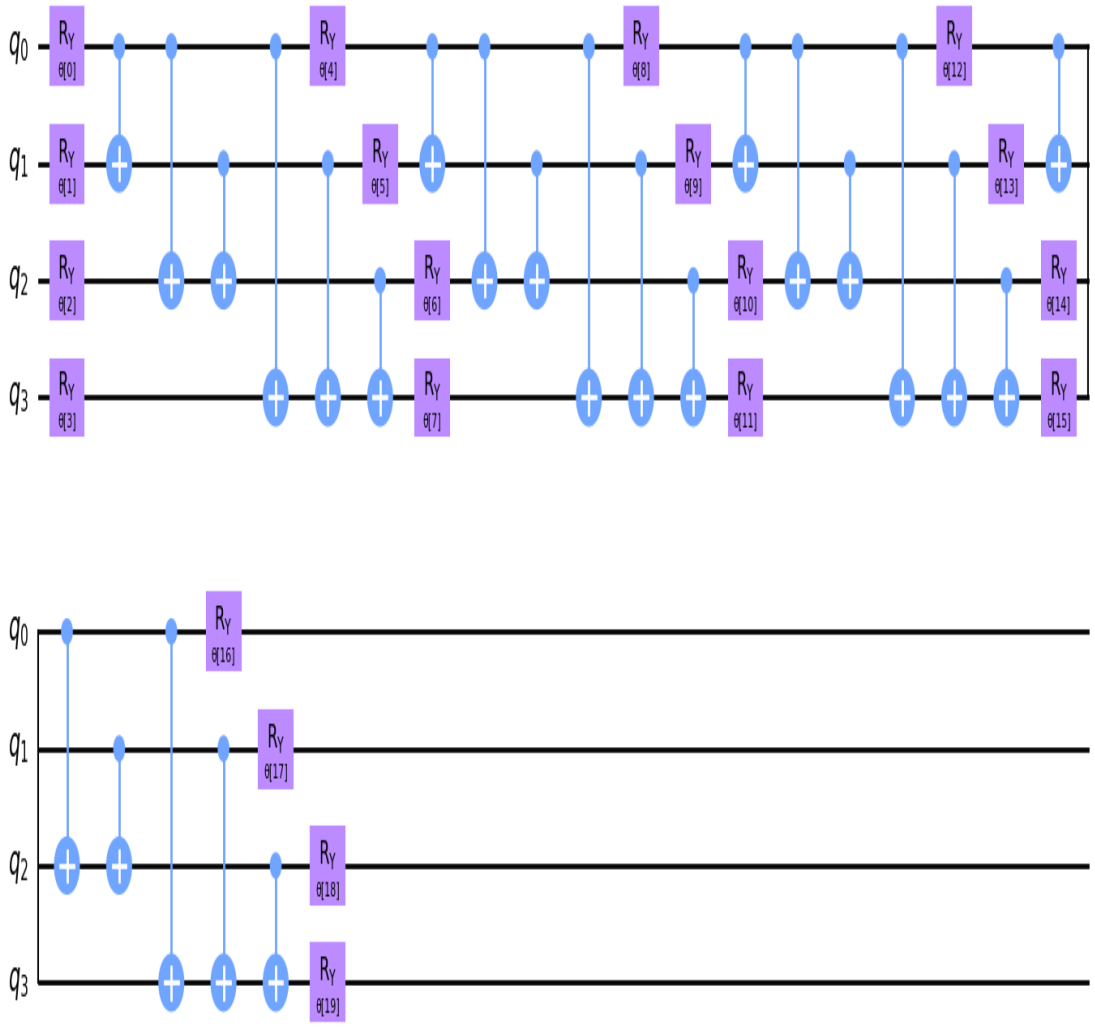

Figure 4.8: Ansatz for IoT Anomaly Dataset

VQC For KDD Cup 99 Dataset:

KDD Cup 99 is huge data, and so figure 4.9 shows the feature map for such a dataset using 7 of its best features. 7 qubits were needed, with Hadamard operations being applied to all of them. Instead of entanglement, the data was represented via repetitions of Hadamard and Pauli-X gates to reduce circuit depth.

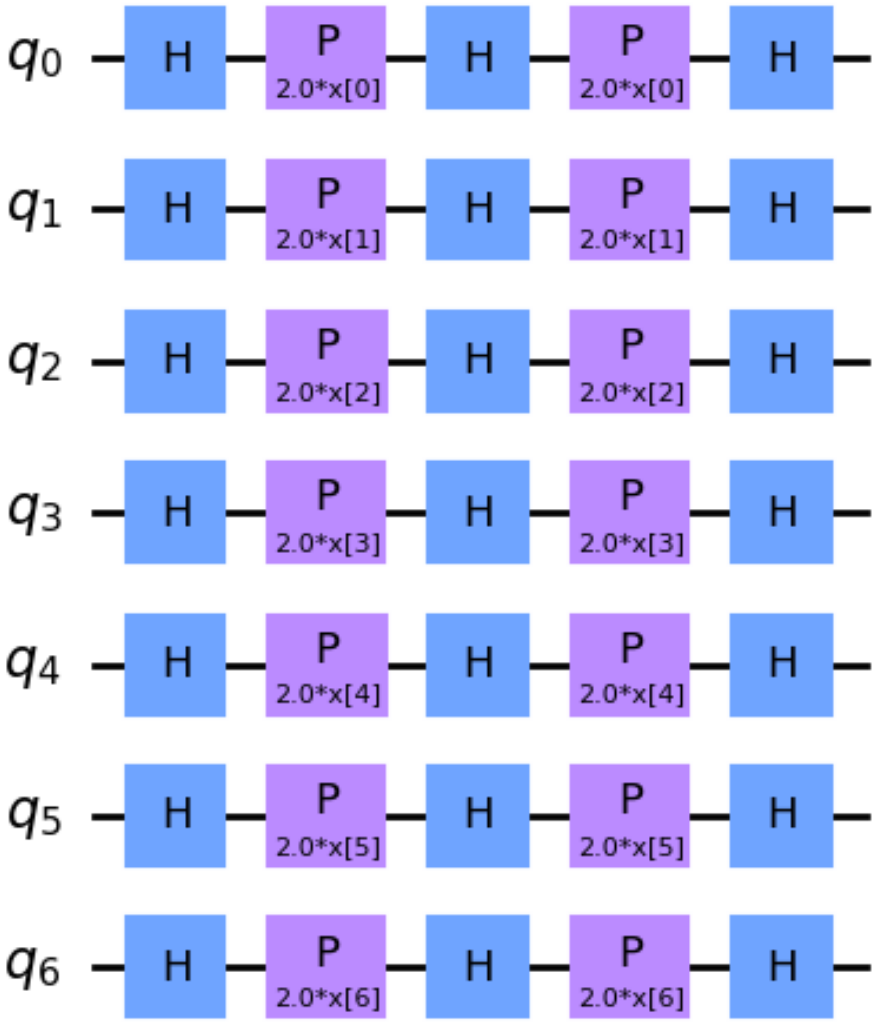

Figure 4.9: Feature Map For KDD Cup 99 Dataset

The ansatz for the experiment, shown by figure 4.10, utilises simple Pauli-Y gates with CNOT gates being applied to every qubit pair. Here, the number of repetitions was kept to one to reduce circuit depth and training time. A more generalised circuit was used in this experiment.

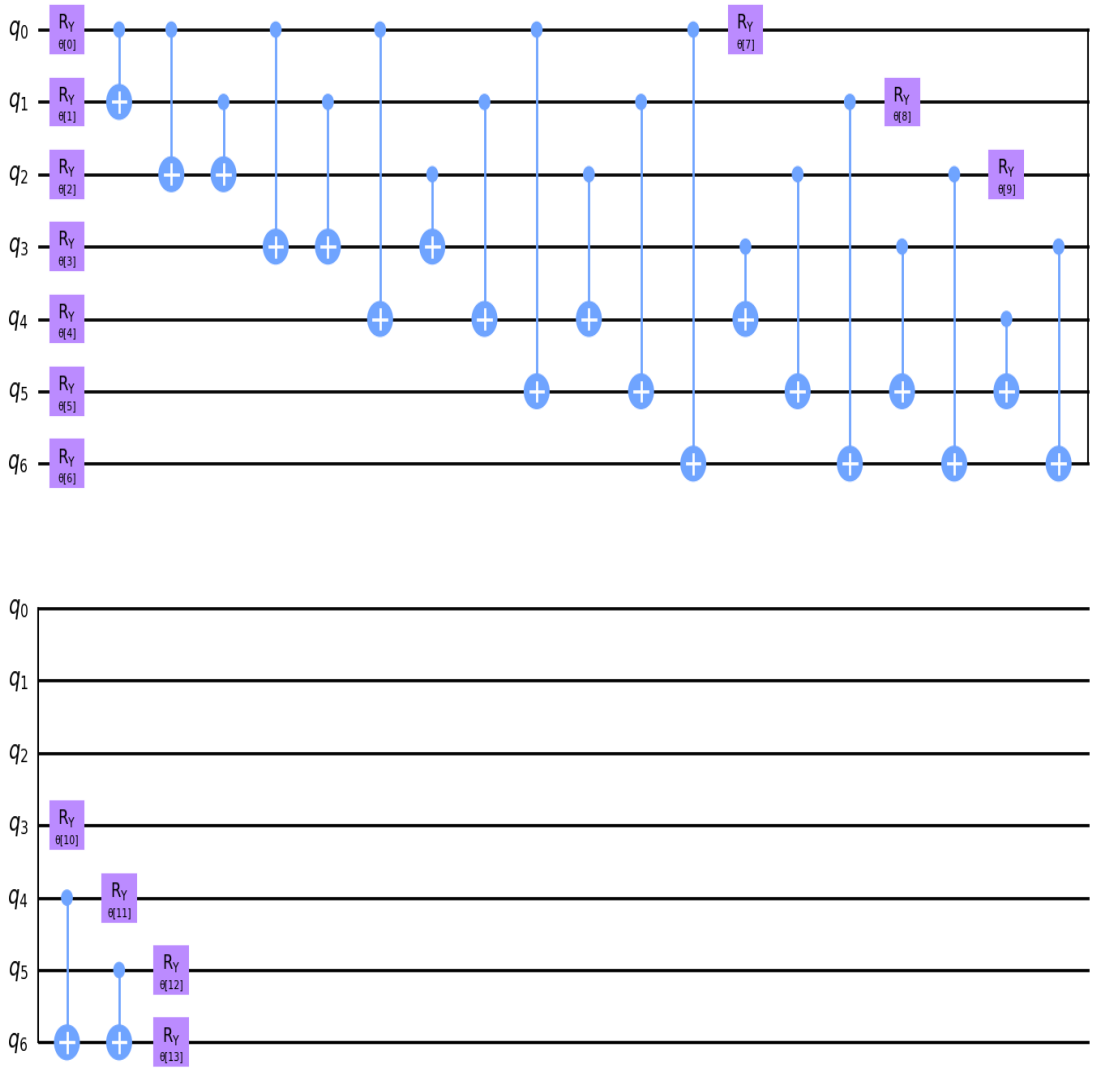

Figure 4.10: Ansatz For KDD Cup 99 Dataset

Figure 4.11 showcases a custom feature map for the KDD Cup dataset, created using the genetic algorithm discussed in [2]. The main goal here was to increase simplicity by reducing entanglement, whilst keeping expressibility at a good level.

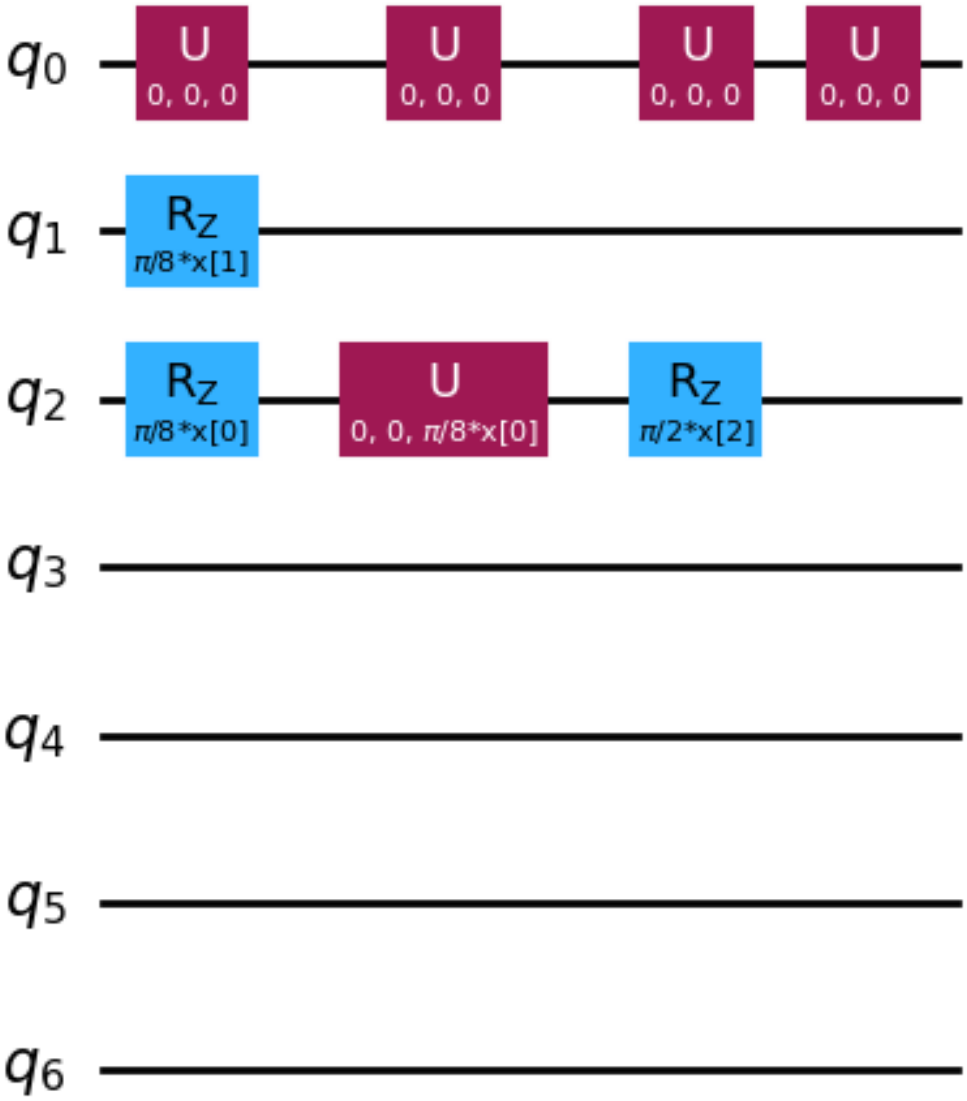

Figure 4.11: Custom Feature Map For KDD Cup 99 Dataset

VQC For TWT Dataset:

The feature map for the TWT dataset, shown by figure 4.12, was created using the **PauliFeatureMap** class from qiskit and consists of an alternating series of Hadamard and Pauli-X gates. Two CNOT gates are present to offer entanglement.

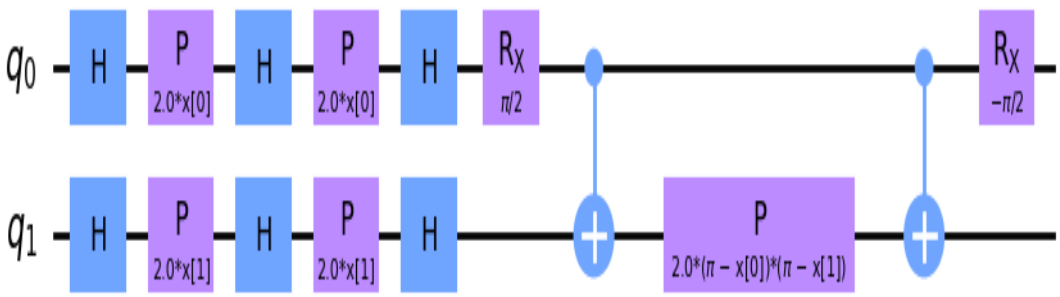

Figure 4.12: Feature Map For TWT Dataset

The ansatz for the TWT dataset, shown by figure 4.13, consists of only four Pauli-Y rotation gates and one CNOT gate. This was done to make training time short and allow us to check for the generalization ability of such a circuit.

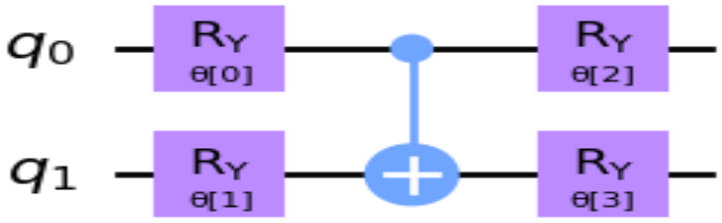

Figure 4.13: Ansatz For TWT Dataset

VQC For Fridge Dataset: The feature map for the Fridge dataset (figure 4.14) is identical to the one used for the TWT dataset. This is because the Fridge dataset contains similar features to the TWT dataset but in a much larger volume. This resulted in us using a similar ansatz for both datasets.

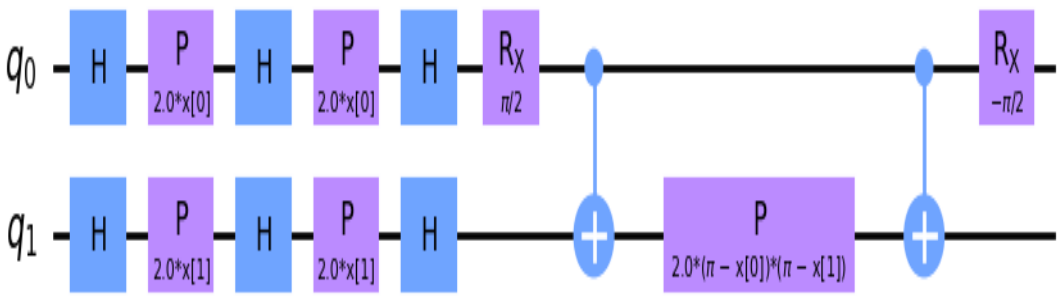

Figure 4.14: Feature Map For Fridge Dataset

In the case of the ansatz (figure 4.15), it is also similar to the TWT dataset, but with there being more repetitions of the ansatz. There are 4 repetitions present in the circuit. This is because the dataset is much larger and so would require a deeper circuit.

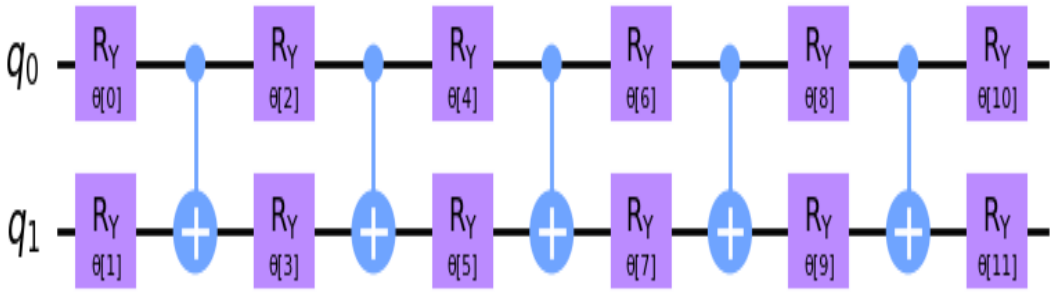

Figure 4.15: Ansatz For Fridge Dataset

VQC For IoT Telemetry Dataset:

Figure 4.16 showcases the feature map used for the IoT Telemetry dataset. It can be seen that the IoT Telemetry dataset, like KDD Cup 99, is huge and consists of 7 feature values. As such, 7 qubits were required. The **PauliFeatureMap** was used in this scenario as this is a good feature map for general classification.

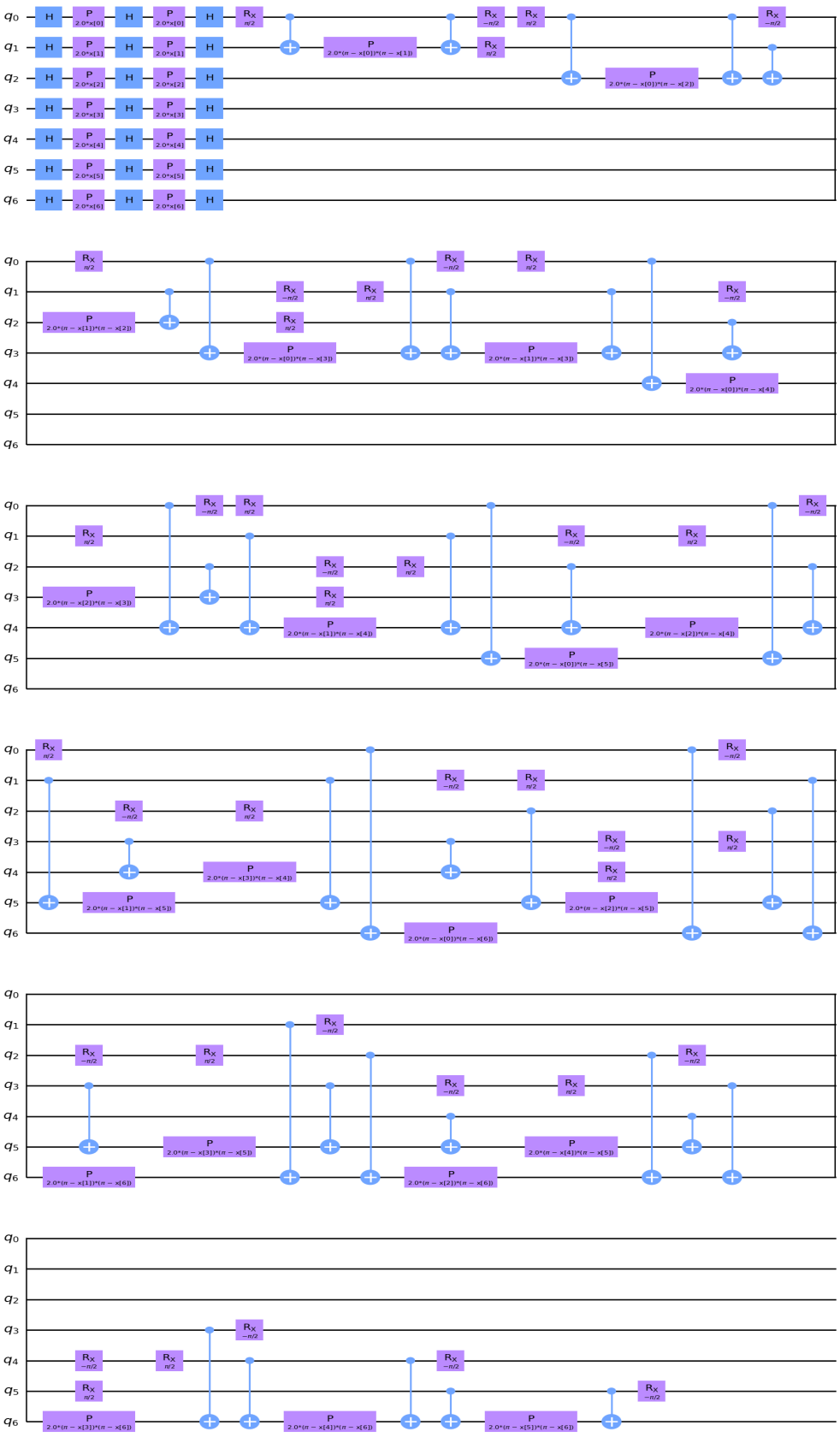

Figure 4.16: Feature Map For IoT Telemetry Dataset

The ansatz for the IoT Telemetry dataset is shown in figure 4.17. It consists of two layers of repetitions of Pauli-Y gates followed by CNOT gates between each qubit pair. This was chosen to make training faster over a reduced dataset.

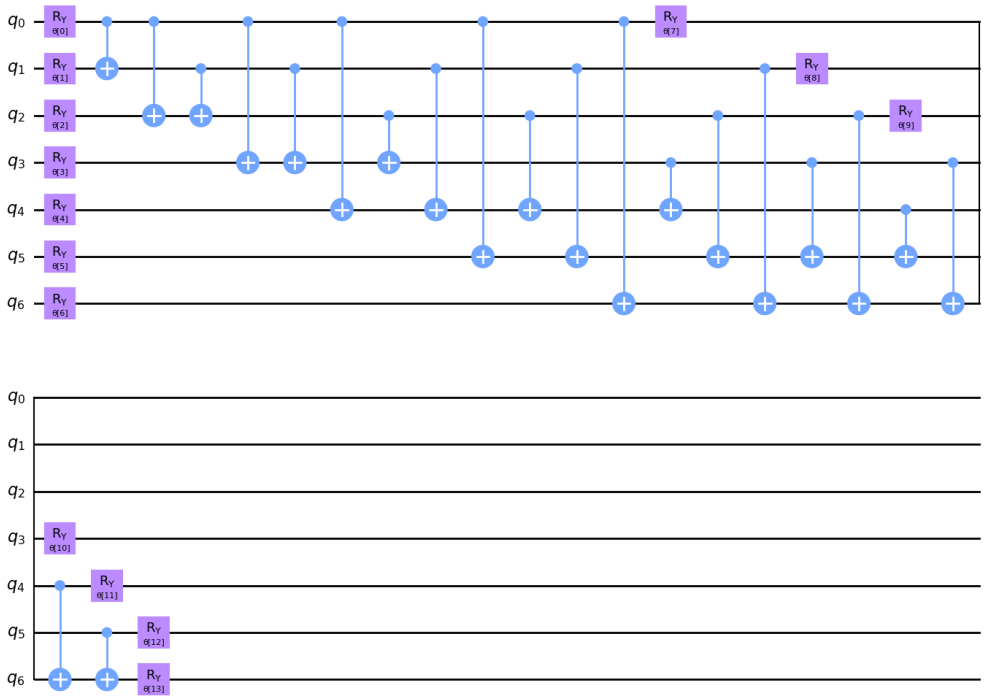

Figure 4.17: Ansatz For IoT Telemetry Dataset

4.2.3 Results And Analysis

After propagating through the ansatz for different datasets calculations were made and the value was measured.

Learning Curve

The loss function over iteration was plotted to understand how long the model took to converge. It can be observed that some of the models are yet to converge. This is due to the lack of proper hardware. Due to the sheer size of the dataset, it was difficult to run over a long period. In such cases, down-sampling of data was taken as an added measure. In other cases, the models converged very easily. Proper parameters were taken so that the model could converge easily.

It can be seen that the learning curves for the KDD Cup 99 dataset (figure 4.18) and the IoT Telemetry dataset (figure 4.19) did not converge. This is due to our hardware limitations, resulting in us not being able to carry out more iterations, even on a scaled-down dataset, to get a convergence point for the datasets.

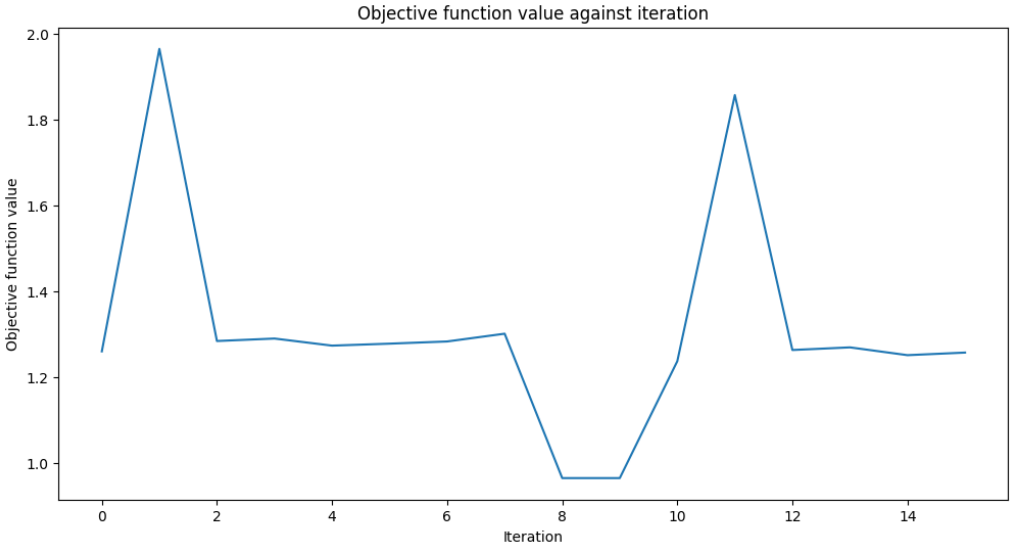

Figure 4.18: Learning Curve For KDD Cup 99 Dataset

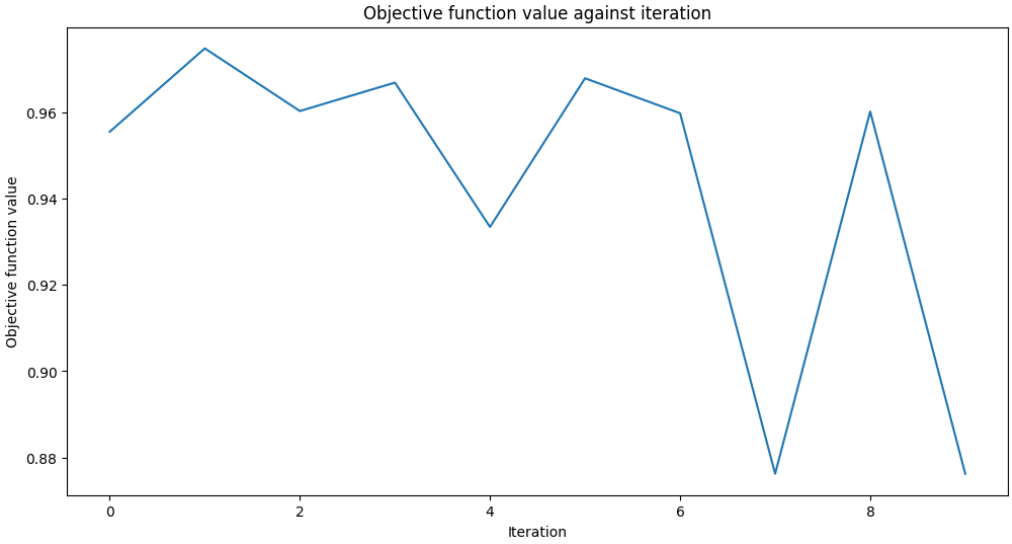

Figure 4.19: Learning Curve For IoT Telemetry Dataset

On the other hand, the TWT, IoT Anomaly and Fridge datasets were small enough that we were able to find the convergence points for all of them. TWT had the least amount of data, so it converged the earliest (figure 4.21). Although the fridge dataset had more than the IoT Anomaly dataset, it converged faster (figure 4.22) because the VQC for the IoT Anomaly dataset had to work with 4 qubits in comparison to the 2 qubits used for the Fridge dataset (figure 4.20).

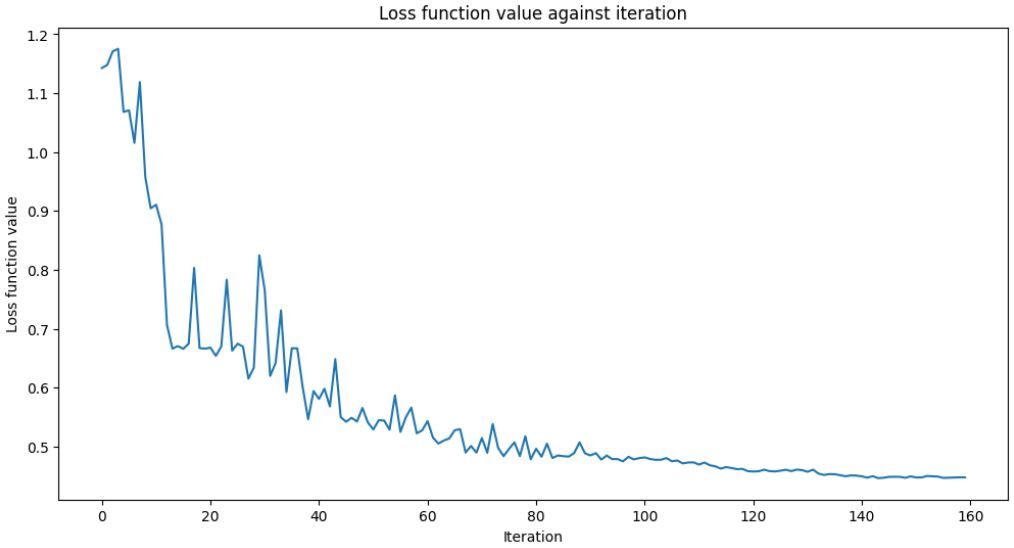

Figure 4.20: Learning Curve For IoT Anomaly Dataset

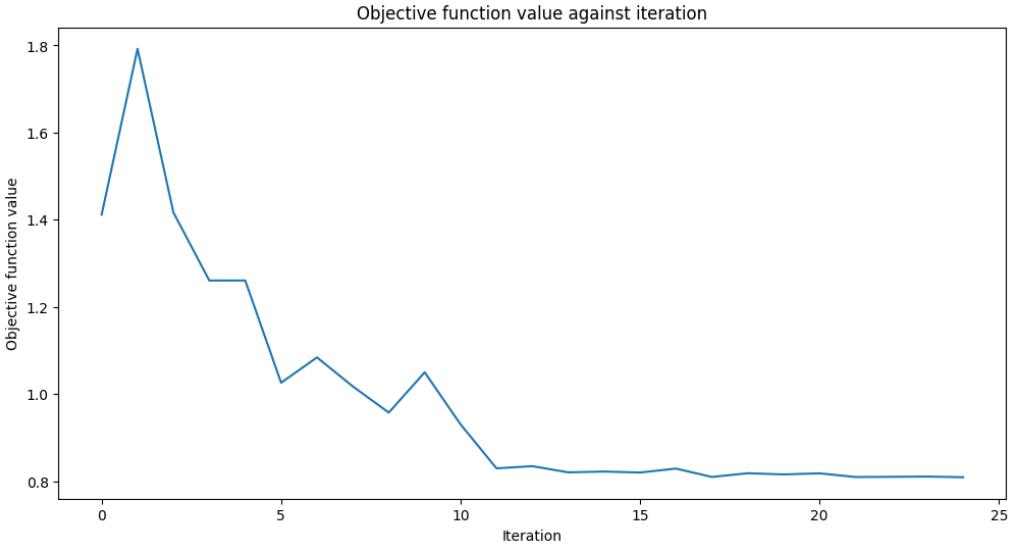

Figure 4.21: Learning Curve For TWT Dataset

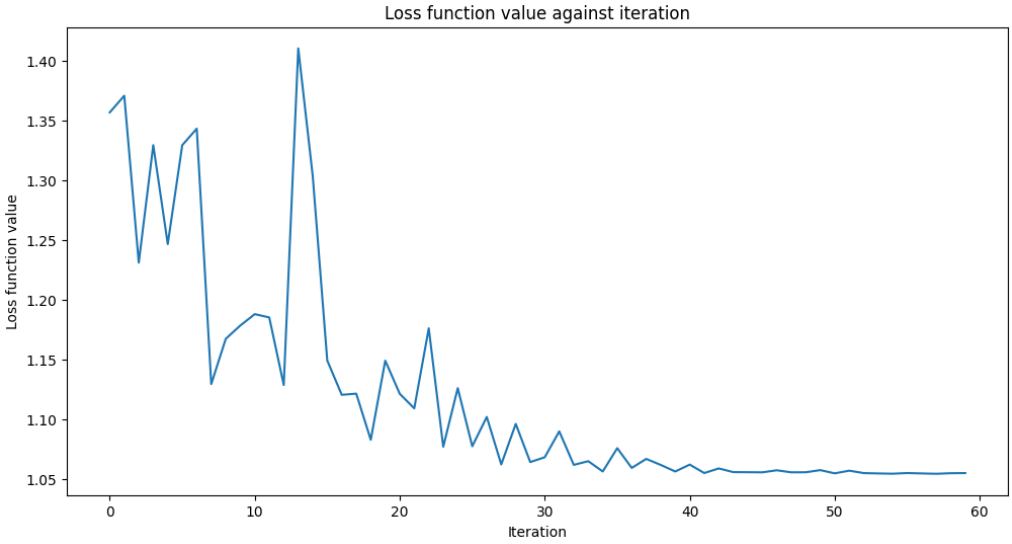

Figure 4.22: Learning Curve For Fridge Dataset

After the model was done learning we made our predictions and the following were the results.

Table 4.10: Results For The Variational Quantum Circuit

Dataset	Number of Iterations	Accuracy (%)
KDDCUP99	10	80.33
anoML-IoT	60	92
IoT Telemetry	10	65.65
TWTDUS	25	73
IoT_Fridge	35	58

It can be seen from the results that VQC worked best for the anoML-IoT dataset. This may be because the VQC algorithm had a feature set and ansatz that was well suited for the dataset. On the other hand, it managed to perform relatively well for the KDDCUP99 dataset even on scaled-down data. Data from TWTDUS may suffer from overfitting from our model, whilst our preprocessing and feature mapping may have been wrong for the IoT_Fridge dataset, where we may have chosen a circuit that traded in trainability for expressibility.

Besides the variational quantum circuit, we also ran UU-† for the anoML-IoT dataset and the texas turbine dataset. The results were as follows:

Table 4.11: Results for UU-† Circuit

Dataset	Number of Shots	Accuracy (%)
KDDCUP99	100	83.36
anoML-IoT	100	82.03
TWTDUS	100	99.62
IoT Telemetry	100	96.08
IoT Fridge	100	55.33

The UU-† circuit showed better levels of accuracy. It showcases the most basic equation of quantum computing, and it is not surprising that it has such a high level of accuracy. As long as enough time is given, the circuit will be able to give good accuracy for an appropriate number of shots.

Chapter 5

Conclusion

The main objective of this research was to build a foundation for quantum machine learning to tackle the size and growth of big data. Lives are getting more and more connected through various IoT devices such as sensors, fridges, thermostats, coffee machines, air conditioners etc. With all this connectivity data is being cultivated. The immense growth of big data urges more efficient and faster computation to become future-proof. This research explores one such avenue i.e. Quantum Computing and Quantum Machine Learning. Particularly experiments were done to detect anomaly detection in IoT networks using Quantum Machine Learning models.

The key finding of this research is the promising capabilities of Quantum Machine Learning. Mainly two models (UU-dagger and variational quantum circuit) were experimented on five datasets. In each of the datasets, promising results were found to have the highest accuracy of 92 percent. The training time did not improve significantly as the experiments were done in a quantum simulator and due to the huge volume of data convergence was difficult in the limited hardware capabilities. However, in quantum hardware, this would significantly improve and show how the models are giving us good accuracy the potential is very high.

The research works as a building block for applying quantum machine learning algorithms and models. It clearly shows the potential of quantum machine learning and how it can be used to perform certain classifications much faster using superposition provided the hardware catches up. The contribution is providing a new way of computation apart from classical machine learning with the potential of exponential speedup.

The main limitation of this research at the moment is the lack of hardware capabilities. Quantum computers are still at an early stage of its glory. While it promises

wonderful things such as exponential speedup due to the superposition of qubits, the availability of quantum computers is the major limitation. However, many simulating platforms such as IBM Quantum Lab, blue qubit etc. can be used to build your circuit and find results. There is also the possibility of queuing into quantum computers virtually but the feasibility is questionable due to long waiting times bandwidth limitation and security concerns. To get the full advantage of quantum computing the hardware needs to catch up with the implementations and applications of quantum machine learning. The research shows the potential of quantum machine learning in classification. This will hopefully inspire a lot of people to try more advanced things with quantum machine learning including simulation of reality. With the advent of more available qubits the possibilities of quantum machine learning are endless.

This research was done to explore a new domain of computation. One which could keep up with the ever-increasing size of big data. While quantum machine learning may not be the solution to most generic problems. It is a powerhouse when it comes to computing large amounts of data through superposition provided the right hardware is available. With this immense potential in mind, this research serves as a foundation and proof of work of quantum machine learning and its implementation.

References

[1] A. Alsaedi, N. Moustafa, Z. Tari, A. Mahmood, and A. Anwar, "Ton_iot telemetry dataset: A new generation dataset of iot and iiot for data-driven intrusion detection systems," *Ieee Access*, vol. 8, pp. 165 130–165 150, 2020.

[2] S. Altares-López, A. Ribeiro, and J. J. García-Ripoll, "Automatic design of quantum feature maps," *Quantum Science and Technology*, vol. 6, no. 4, p. 045 015, Aug. 2021, ISSN: 2058-9565. DOI: 10.1088/2058-9565/ac1ab1. [Online]. Available: http://dx.doi.org/10.1088/2058-9565/ac1ab1.

[3] V. Bolon-Canedo, N. Sanchez-Marono, and A. Alonso-Betanzos, "Feature selection and classification in multiple class datasets: An application to kdd cup 99 dataset," *Expert Systems with Applications*, vol. 38, no. 5, pp. 5947–5957, 2011.

[4] K. DeMedeiros, A. Hendawi, and M. Alvarez, "A survey of ai-based anomaly detection in iot and sensor networks," *Sensors*, vol. 23, no. 3, p. 1352, 2023.

[5] Y. Du, T. Huang, S. You, M.-H. Hsieh, and D. Tao, "Quantum circuit architecture search for variational quantum algorithms," *npj Quantum Information*, vol. 8, no. 1, p. 62, 2022.

[6] M. A. Al-Garadi, A. Mohamed, A. Al-Ali, X. Du, I. Ali, and M. Guizani, "A survey of machine and deep learning methods for internet of things (iot) security," *IEEE Communications Surveys Amp; Tutorials*, vol. 22, pp. 1646–1685, 3 2020. DOI: 10.1109/comst.2020.2988293.

[7] J. Goh, S. Adepu, K. N. Junejo, and A. Mathur, "A dataset to support research in the design of secure water treatment systems," in *Critical Information Infrastructures Security: 11th International Conference, CRITIS 2016, Paris, France, October 10–12, 2016, Revised Selected Papers 11*, Springer, 2017, pp. 88–99.

[8] X. Guo, T. Muta, and J. Zhao, "Quantum circuit ansatz: Abstraction and reuse of quantum algorithm design," *arXiv preprint arXiv:2405.05021*, 2024.

[9] R. Gurunath, M. Agarwal, A. Nandi, and D. Samanta, "An overview: Security issue in iot network," in *2018 2nd international conference on I-SMAC (IoT in social, Mobile, analytics and cloud)(I-SMAC) I-SMAC (IoT in social, Mobile, an-*

alytics and cloud)(I-SMAC), 2018 2nd international conference on, IEEE, 2018, pp. 104–107.

[10] H. Kayan, Y. Majib, W. Alsafery, M. Barhamgi, and C. Perera, "Anoml-iot: An end to end re-configurable multi-protocol anomaly detection pipeline for internet of things," *Internet of Things*, vol. 16, p. 100 437, 2021.

[11] J. S. Kumar and D. R. Patel, "A survey on internet of things: Security and privacy issues," *International Journal of Computer Applications*, vol. 90, no. 11, 2014.

[12] P. Kushwaha, H. Buckchash, and B. Raman, "Anomaly based intrusion detection using filter based feature selection on kdd-cup 99," in *TENCON 2017-2017 IEEE Region 10 Conference*, IEEE, 2017, pp. 839–844.

[13] A. Nakayama, K. Mitarai, L. Placidi, T. Sugimoto, and K. Fujii, *Vqe-generated quantum circuit dataset for machine learning*, 2023. arXiv: 2302.09751 [quant-ph].

[14] A. Nazir, S. Sholla, and A. Bashir, "Internet of things security: Issues, challenges and counter-measures," *International Journal of Network and Technology*, vol. 7, no. 3, 2019.

[15] A. Pellow-Jarman, I. Sinayskiy, A. Pillay, and F. Petruccione, "A comparison of various classical optimizers for a variational quantum linear solver," *Quantum Information Processing*, vol. 20, no. 6, Jun. 2021, ISSN: 1573-1332. DOI: 10.1007/s11128-021-03140-x. [Online]. Available: http://dx.doi.org/10.1007/s11128-021-03140-x.

[16] J. Qin, "Review of ansatz designing techniques for variational quantum algorithms," in *Journal of Physics: Conference Series*, IOP Publishing, vol. 2634, 2023, p. 012 043.

[17] S. K. Satpathy, V. Vibhu, B. K. Behera, S. Al-Kuwari, S. Mumtaz, and A. Farouk, "Analysis of quantum machine learning algorithms in noisy channels for classification tasks in the iot extreme environment," *IEEE Internet of Things Journal*, 2023.

[18] S. K. Satpathy, V. Vibhu, B. K. Behera, S. Al-Kuwari, S. Mumtaz, and A. Farouk, "Analysis of quantum machine learning algorithms in noisy channels for classification tasks in the iot extreme environment," *IEEE Internet of Things Journal*, 2023.

[19] T. Sharma and R. A. H. Khan, "Optimizing network security using lstm algorithm for traffic classification on unswnb15 and kddcup99 dataset," *International Journal of Intelligent Systems and Applications in Engineering*, vol. 12, no. 8s, pp. 671–682, 2024.

[20] S. Sim, P. D. Johnson, and A. Aspuru-Guzik, "Expressibility and entangling capability of parameterized quantum circuits for hybrid quantum-classical algorithms," *Advanced Quantum Technologies*, vol. 2, no. 12, Oct. 2019, ISSN: 2511-9044. DOI: 10.1002/qute.201900070. [Online]. Available: http://dx.doi.org/10.1002/qute.201900070.

[21] D. Srinivasan, K. Chakrabarti, N. Chopra, and A. Dutt, *Quantum circuit optimization through iteratively pre-conditioned gradient descent*, 2023. arXiv: 2309.09957 [quant-ph].

[22] O. C. Stoica, *Born rule: Quantum probability as classical probability*, 2023. arXiv: 2209.08621 [quant-ph].

[23] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the kdd cup 99 data set," in *2009 IEEE symposium on computational intelligence for security and defense applications*, Ieee, 2009, pp. 1–6.

[24] Y. Tjandra and H. S. Sugiarto, "An evolutionary algorithm design for pauli-based quantum kernel classification," 2022.

[25] I. P. Turnipseed, *A new scada dataset for intrusion detection research*. Mississippi State University, 2015.

[26] N. Veyrat-Charvillon and F.-X. Standaert, "Mutual information analysis: How, when and why?" In *International Workshop on Cryptographic Hardware and Embedded Systems*, Springer, 2009, pp. 429–443.

[27] Y.-Y. Yang, C. Rashtchian, H. Zhang, R. R. Salakhutdinov, and K. Chaudhuri, "A closer look at accuracy vs. robustness," *Advances in neural information processing systems*, vol. 33, pp. 8588–8601, 2020.

[28] A. Yazdinejad, M. Kazemi, R. M. Parizi, A. Dehghantanha, and H. Karimipour, "An ensemble deep learning model for cyber threat hunting in industrial internet of things," *Digital Communications and Networks*, vol. 9, no. 1, pp. 101–110, 2023.

[29] A. Zeguendry, Z. Jarir, and M. Quafafou, "Quantum machine learning: A review and case studies," *Entropy*, vol. 25, no. 2, p. 287, 2023.

Appendices

Appendix A

Quantum Information

A.1 Qubit

The main difference between classical and quantum computing is the unit of information. For classical computing, we have bits; whereas for quantum computing we have quantum bits or qubits. Bits can only represent one state (0 or 1) at a time. But qubits can store multiple states (both 0 and 1) in superposition. Qubits are mathematically represented using Dirac notations: $|\rangle$ ket and $\langle|$ bra.

- $|0\rangle$: Ket 0 represents a qubit in state '0'. It has the vector representation of:

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

- $|1\rangle$: Ket 1 represents a qubit in state '1'. It has the vector representation of:

$$|1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

- $|\psi\rangle$: Ket ψ represents the general state of a qubit. Here 'ψ' is a quantum state. It has the vector representation of:

$$|\psi\rangle = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}$$

  Here, α and β are complex numbers representing probability amplitudes.

- $\langle 0|$: Bra 0 represents the row vector:

$$\langle 0| = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

- $\langle 1|$: Bra 1 represents the row vector:

$$\langle 1| = \begin{bmatrix} 0 & 1 \end{bmatrix}$$

- $\langle \psi|$: Bra ψ represents the row vector:

$$\langle \psi| = \begin{bmatrix} \alpha^* & \beta^* \end{bmatrix}$$

  Here, α^* and $\beta *$ are the complex conjugates of α and β.

$|0\rangle$ and $|1\rangle$ are considered basis states in a single qubit system. In case of two qubits, the basis states are: $|00\rangle$, $|01\rangle$, $|10\rangle$ and $|11\rangle$, where each digit corresponds to the state of a single qubit.

A qubit in superposition is represented as:

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

Here, α and β are complex numbers representing probability amplitudes and $|\alpha^2| + |\beta^2| = 1$.

Two qubits in superposition are represented as:

$$|\psi\rangle = \alpha\,|00\rangle + \beta\,|01\rangle + \gamma\,|10\rangle + \delta\,|11\rangle$$

Here, α, β, γ and δ are complex numbers representing probability amplitudes and $|\alpha^2| + |\beta^2| + |\gamma^2| + |\delta^2| = 1$.

Besides superposition, qubits utilize fundamental quantum properties such as entanglement and teleportation to further diversify the capabilities of quantum computing.

A.2 Hilbert Space

A Hilbert space is a complete vector space equipped with an inner product. This inner product allows us to measure angles and distances between vectors, which are vital for an understanding of quantum states and their evolution. Hilbert space constitutes the foundational concept of quantum mechanics and quantum computing. It lays the mathematical foundation for describing the state space of quantum systems.

A.2.1 Key Properties

- **Vectors (States):** The elements of a Hilbert space are vectors representing the possible states of a quantum system. Other standard terms for these state vectors in quantum computing are state vectors and quantum states.
- **Inner Product:** An inner product is just a complex-valued function which, given two vectors, provides a scalar. If we have vectors $|\psi\rangle$ and $|\phi\rangle$, we denoted their inner product as $\langle\psi|\phi\rangle$.
- **Norm:** The norm of a vector $|\psi\rangle$ is $|\psi| = \sqrt{\langle\psi|\psi\rangle}$. Quantum states are usually normalized so that $||\psi|| = 1$.
- **Orthogonal**: Two vectors $|\psi\rangle$ and $|\phi\rangle$ are orthogonal if $\langle\psi|\phi\rangle = 0$. Orthogonal states thus represent states that have mutually exclusive outcomes.

A.2.2 Basis and Dimension

- **Basis Vectors:** In Hilbert space, essentially a set of vectors composes a basis, which builds up the whole space. The most common basis in quantum computing is the computational basis, whereas the single qubit base vectors are usually written as $|0\rangle$ and $|1\rangle$.
- **Superposition:** Any vector of the Hilbert space can be written as a linear superposition of basis vectors. For a single qubit, a general state is $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, where α and β are complex numbers.
- **Dimension:** The dimension of Hilbert space is the number of basic vectors required to span it. For a system of n qubits, the Hilbert space is of dimension 2^n

A.2.3 Operators

- **Operators:** In the Hilbert space, operators are functions that map vectors to other vectors within the same space. In quantum mechanics, operators are generally physical observables, so, in fact, they are linear.
- **Hermitian Operators:** This class of operators is of particular importance, since their eigenvalues correspond directly to measurable quantities. A Hermitian operator is one in which $A = A\dagger$, where $A\dagger$ is the conjugate transpose of the operator A.
- **Unitary operators:** These are operators that preserve the norm of vectors and are used to describe the time evolution of quantum state vectors. An operator U is said to be unitary when, $U\dagger U = UU\dagger = I$, where I is the identity operator.

A.3 Entanglement

Entanglement in the quantum state arises when the quantum states of two or more particles are so connected that a particle's state can no longer be described independently of the states of the others. For example,

$$|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$

This represents two qubits that are perfectly correlated: if one of them is measured to be in the $|0\rangle$ state, the other will also be in the $|0\rangle$ state, and likewise for $|1\rangle$. Here, $|\Phi^+\rangle$ is a well known Bell State. The Bell states are the simplest and most relevant entangled states of two qubits.

A.3.1 Properties

- **Quantum Correlation:** In a quantum-entangled state, particles' properties are very strongly correlated, independent of their separation distance. This means an instantaneous change at one will affect the other, a relation that Einstein famously called spooky action at a distance.
- **EPR Paradox:** Named after Einstein, Podolsky, and Rosen, this paradox highlights the bizarre nature of entanglement. It questions whether quantum mechanics gives a complete description of reality and has driven numerous interpretations and debates in the foundations of quantum theory.

- **Bell's Theorem:** John Bell formulated inequalities to test predictions of quantum mechanics against classical physics (local realism). Experiments that violate Bell's inequalities are consistent with non-classical predictions made by quantum mechanics, reinforcing the reality of entanglement.

A.3.2 Applications

- **Quantum Teleportation:** Entanglement enables a quantum state to be teleported from one place to another without moving a particle.
- **Superdense Coding:** Two classical bits of information can be encoded in one qubit through entanglement.
- **Quantum Cryptography:** Entangled particles are used in quantum key distribution protocols to create theoretically secure communication channels.

A.4 Measurement

Quantum measurement is the process in which classical information is extracted from a quantum system; generally, this leads to the collapse of the quantum state into one of the eigenstates of the observable being measured.

A.4.1 Key Points

- **Observable and Eigenstates:** Every measurable quantity in quantum mechanics corresponds to an observable—a Hermitian operator. The eigenstates of the corresponding operator identify the possible results of a measurement.
- **State Collapse:** During measurement, a quantum system 'collapses' from a superposition state into one of the eigenstates of the observable. The probability of collapsing to a particular eigenstate is the square of the amplitude of that eigenstate in the original superposition.
- **Born Rule**: This rule yields the probability for obtaining a certain measurement outcome. If a quantum state $|\psi\rangle$ is provided as a superposition of the eigenstates $|\phi_i\rangle$ of some observable A,

$$|\psi\rangle = \sum_i c_i |\phi_i\rangle ,$$

then the probability of measuring the eigenvalue a_i corresponding to $|\phi_i\rangle$ is $|c_i|^2$.

- **Projective Measurement:** This is the standard model of measurement in quantum mechanics. If an observable A is measured on a quantum state $|\psi\rangle$ it collapses into the eigenstate corresponding to the measured eigenvalue.

A.4.2 Measurement Example

Single Qubit Measurement: Consider a qubit in the following state:

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

- The probability of measuring $|0\rangle$ is $|\alpha|^2$.
- The probability of measuring $|1\rangle$ is $|\beta|^2$.

After measuring:

- If $|0\rangle$ is measured, the state collapses to $|0\rangle$.
- If $|1\rangle$ is measured, the state collapses to $|1\rangle$.

Two Qubit Measurement: Consider a system of two qubits in the following state:

$$|\psi\rangle = \alpha\,|00\rangle + \beta\,|01\rangle + \gamma\,|10\rangle + \delta\,|11\rangle\,.$$

When measuring in the computational basis $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$:

- The probability of measuring $|00\rangle$ is $|\alpha|^2$.
- The probability of measuring $|01\rangle$ is $|\beta|^2$.
- The probability of measuring $|10\rangle$ is $|\gamma|^2$.
- The probability of measuring $|11\rangle$ is $|\delta|^2$.

After measuring:

- If $|00\rangle$ is measured, the state collapses to $|00\rangle$.
- If $|01\rangle$ is measured, the state collapses to $|01\rangle$.
- If $|10\rangle$ is measured, the state collapses to $|10\rangle$.
- If $|11\rangle$ is measured, the state collapses to $|11\rangle$.

Appendix B

Quantum Gates

Quantum Gates are the basic components in a quantum circuit. They manipulate qubits to perform operations. Essentially, each quantum gate is a special well defined matrix. These matrices are multiplied with qubit vectors to carry out quantum operations.

B.1 Single Qubit Gates

B.1.1 Hadamard Gate

Hadamard gate is a fundamental quantum gate that is used to create a superposition of $|0\rangle$ and $|1\rangle$ with equal probabilities. This is a crucial operation to achieve parallelism. This gate is denoted as "H" and its matrix representation is:

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

Applying the Hadamard gate on a single qubit transforms the basis states as following:

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

For n qubits in a quantum system, applying the Hadamard gate yields a superposition of all 2^n possible basis states.

B.1.2 Pauli Gates

Pauli Gates in quantum computing are a set of 3 gates modeled after the Pauli Matrices. They are known as the X, Y and Z gates.

1. Pauli-X Gate (X Gate):
    - Matrix representation:
    $$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$
    - Operation: Similar to classical NOT gate, it flips the state of a qubit.
2. Pauli-Y Gate (Y Gate):
    - Matrix representation:
    $$Y = \begin{bmatrix} 0 & -\mathrm{i} \\ \mathrm{i} & 0 \end{bmatrix}$$
    - Operation: It flips the state of a qubit and performs a phase shift.
3. Pauli-Z Gate (Z Gate):
    - Matrix representation:
    $$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$
    - Operation: Performs a phase shift on the qubit.

B.1.3 Phase Shift Gate (S Gate)

The Phase Shift Gate, often denoted as the S gate, applies a phase shift of $\frac{\pi}{2}$ to the $|1\rangle$ state, leaving the $|0\rangle$ state unchanged. Its matrix representation is:

$$S = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$$

Applying the Phase Shift Gate on a single qubit transforms the basis states as follows:

- $S|0\rangle = |0\rangle$
- $S|1\rangle = i|1\rangle$

B.1.4 T Gate ($\pi/8$ Gate)

The T Gate, also known as the $\pi/8$ Gate, applies a phase shift of $\frac{\pi}{4}$ to the $|1\rangle$ state, leaving the $|0\rangle$ state unchanged. Its matrix representation is:

$$T = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}$$

Applying the T Gate on a single qubit transforms the basis states as follows:

- $T|0\rangle = |0\rangle$
- $T|1\rangle = e^{i\pi/4}|1\rangle$

B.2 Multi Qubit Gates

B.2.1 CNOT Gate (Controlled-NOT Gate)

The CNOT Gate is a two-qubit gate that flips the target qubit if the control qubit is in the $|1\rangle$ state. Its matrix representation is:

$$\text{CNOT} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Applying the CNOT Gate on two qubits transforms the basis states as follows:

- $\text{CNOT}|00\rangle = |00\rangle$
- $\text{CNOT}|01\rangle = |01\rangle$
- $\text{CNOT}|10\rangle = |11\rangle$
- $\text{CNOT}|11\rangle = |10\rangle$

B.2.2 SWAP Gate

The SWAP Gate is a two-qubit gate that swaps the states of two qubits. Its matrix representation is:

$$\text{SWAP} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Applying the SWAP Gate on two qubits transforms the basis states as follows:

- $\text{SWAP}\,|00\rangle = |00\rangle$
- $\text{SWAP}\,|01\rangle = |10\rangle$
- $\text{SWAP}\,|10\rangle = |01\rangle$
- $\text{SWAP}\,|11\rangle = |11\rangle$

B.2.3 Toffoli Gate (CCNOT Gate)

The Toffoli Gate, also known as the Controlled-Controlled-NOT Gate, is a three-qubit gate that flips the target qubit if both control qubits are in the $|1\rangle$ state. Its matrix representation is:

$$\text{CCNOT} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Applying the Toffoli Gate on three qubits transforms the basis states as follows:

- $\text{CCNOT}\,|abc\rangle = |ab(a \oplus (b \cdot c))\rangle$

B.2.4 Controlled Phase Gate (CZ Gate)

The Controlled Phase Gate, often denoted as the CZ Gate, applies a phase flip to the target qubit if the control qubit is in the $|1\rangle$ state. Its matrix representation is:

$$\mathrm{CZ} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$$

Applying the Controlled Phase Gate on two qubits transforms the basis states as follows:

- $\mathrm{CZ}\,|00\rangle = |00\rangle$
- $\mathrm{CZ}\,|01\rangle = |01\rangle$
- $\mathrm{CZ}\,|10\rangle = |10\rangle$
- $\mathrm{CZ}\,|11\rangle = -\,|11\rangle$

B.2.5 Unitary Gate (U-gate)

The Unitary gate is provided by qiskit to act as a universal Pauli gate, where the manipulation of the input parameters can result in different phase operations on a qubit.

$$U(\theta, \phi, \lambda) = \begin{pmatrix} \cos(\theta/2) & -e^{i\lambda}\sin(\theta/2) \\ e^{i\phi}\sin(\theta/2) & e^{i(\phi+\lambda)}\cos(\theta/2) \end{pmatrix}$$

An example of the U-gate is as follows:

$$U(\theta, -\pi/2, -\pi/2) = RX(\theta)$$