Machine Learning Assisted Noise Classification and Mitigation in Quantum Key Distribution Protocol

A Thesis

submitted to
Indian Institute of Science Education and Research Pune
in partial fulfillment of the requirements for the
BS-MS Dual Degree Programme

by

Ashmi. A

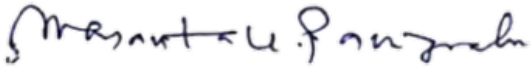
Indian Institute of Science Education and Research Pune
Dr. Homi Bhabha Road,
Pashan, Pune 411008, INDIA.

May, 2025

Supervisor: Prof. Prasanta K Panigrahi

Certificate

This is to certify that this dissertation entitled Machine Learning Assisted Noise Classification and Mitigation in Quantum Key Distribution Protocoltowards the partial fulfilment of the BS-MS dual degree programme at the Indian Institute of Science Education and Research, Pune, represents study/work carried out by Ashmi. A, at Indian Institute of Science Education and Research under the supervision of Prof. Prasanta K Panigrahi, Professor, Center for Quantum Science and Technology, SoA University, during the academic year 2024-2025.



Prof. Prasanta K Panigrahi

Ashmi. A

Committee:

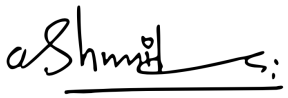
Prof. Prasanta K Panigrahi

Prof. T. S Mahesh

This thesis is dedicated to all students

Declaration

I hereby declare that the matter embodied in the report entitled Machine Learning Assisted Noise Classification and Mitigation in Quantum Key Distribution Protocol are the results of the work carried out by me at the Center for Quantum Science and Technology, Siksha 'O' Anusandhan University, Bhuvaneswar, Department of Physics, Indian Institute of Science Education and Research Pune under the supervision of Prof. Prasanta K Panigrahi and the same has not been submitted elsewhere for any other degree.



Ashmi. A
Reg No: 20191104

Acknowledgments

I would like to express my sincere appreciation to my supervisor, Prof. Prasanta K. Panigrahi, and my mentor, Asst.Prof. Shreya Banerjee, for their exceptional mentorship. I am deeply grateful for invaluable support, guidance, and the flexibility they provided throughout my thesis work. I also extend my sincere thanks to Prof. T. S. Mahesh for his valuable time and support as an expert.

I am especially thankful to my best friend, Sourabh Choudhary, for encouraging me to pursue this thesis work and for his constant support during this journey. I would like to thank Asst.Prof Shreya Banerjee for her incredible mentoring and guidance in Computational task. And Dr. Suchetana Pal for discussions on noise mitigation through singular value decomposition. My heartfelt thanks goes to my Parents, brother and friends - Haraprasad Nandi, Ziyad Thekkayil, Robin K.P for their support during this BSMS program.

Finally, I extend my thanks to my family and all my friends at IISER for their support throughout my BS-MS program and for the unforgettable experiences we shared.

Abstract

With recent advancements in quantum information and computation hardware, the field is at a pivotal stage where real-world applications rely on multiple rounds of error correction or mitigation. However, due to scalability challenges and the inherent noisiness of quantum hardware, error correction techniques are still in their infancy. As an alternative, error mitigation through classical post-processing provides a practical solution. Many mitigation techniques are optimized for specific noise types, making quantum noise characterization a key research area. Several approaches, including machine learning, have been explored for noise classification. Neural networks have been used to distinguish between quantum and classical parameters of dephasing noise and to classify Markovian and non-Markovian noise types.

Quantum Key Distribution (QKD) protocols, [1]such as BB84 and BBM92 [1], enable secure key exchange using single and two-qubit operations on gate-based quantum computers. However, noise in quantum gates can introduce quantum bit errors in the generated encryption key. In this work, I demonstrate that the distribution of Quantum Bit Error Rate (QBER) in QKD protocols can reveal underlying noise types. Using supervised machine learning algorithms,K-Nearest Neighbors[2], Gaussian Naïve Bayes [3], and Support Vector Machines[4] leads to high accuracy in binary classification of depolarizing, amplitude damping, and bit-flip noise models in Quantum key distribution channels and gate based quantum computers.

A key innovation of my approach is its simplicity, utilizing basic circuits, statistical analysis, and low-depth classifiers. Additionally, I have developed a noise mitigation method for QKD channels and gate-based quantum computers using Singular Value Decomposition (SVD) of noisy Matrices, achieving near-perfect efficiency. Looking ahead, I plan to develop a novel QKD protocol that leverages the transfer of the standard deviation of measurements between Alice and Bob using a three-qubit superposition state.

Contents

List of Figures

Chapter 1

Introduction

Quantum communication [1] is the quantum counterpart of classical communication, providing a fundamentally secure method for information transmission by utilizing the principles of quantum mechanics. In classical communication, information is transferred through various means, such as fiber-optic cables, where electrical or optical signals carry encoded data. In contrast, quantum communication relies on the transmission of polarized photons [5], which serve as carriers of quantum information. The encoding of information within these quantum states ensures the security of transmission, as the fundamental principles of quantum mechanics govern the process.

Quantum communication falls under the broader domain of quantum technologies, focusing on developing secure methods for data transmission. Over the years, numerous quantum communication and key distribution protocols have been proposed and developed to enhance the security of information exchange. The emergence of post-quantum cryptography further highlights the need for alternative security frameworks resilient against quantum threats.

One of the core principles of quantum communication is the encoding and decoding of information while adhering to the fundamental theorems of quantum mechanics, such as the no-cloning theorem [6] and the no-signaling theorem. The pioneering quantum key distribution (QKD) protocol, BB84[5], and the entanglement-based BBM92 [7] protocol laid the foundation for secure quantum communication. Additionally, other protocols, such as E91, SARG04, and multi-qubit entangled state protocols, have been developed to enhance security and efficiency. However, despite significant advancements, the scalability and practical utility of these protocols pose substantial challenges, primarily due to the complexities of physical implementation.

One of the major obstacles in the implementation of quantum communication is the interaction of quantum states with their surrounding environment. This interaction often leads to decoherence, causing the quantum properties of the transmitted information to change, thereby introducing errors in the expected outcome. Such errors can result in misinformation, ultimately compromising the reliability of quantum key distribution protocols. Addressing these challenges requires effective noise mitigation strategies to ensure accurate key generation with minimal errors.

In the Noisy Intermediate-Scale Quantum (NISQ) era [8] , considerable attention has been directed toward building and scaling practical quantum technologies. The focus has shifted towards bridging the gap between experimental and theoretical advancements, ensuring a parallel progression in the development of quantum communication technologies. As part of efforts to enhance the performance of quantum key distribution, our research has focused on identifying and classifying noise types in QKD channels and gate-based quantum computers using supervised machine learning methods. By analyzing the distribution of quantum bit error rates (QBER).

Our approach involves the use of machine learning algorithms to identify and classify noise sources affecting QKD channels. By training supervised learning models on QBER distributions, we can effectively pinpoint noise types and develop appropriate mitigation strategies. This approach allows for adaptive noise correction, enhancing the accuracy of quantum key distribution even in the presence of environmental disturbances. Furthermore, we have developed a noise-independent mitigation strategy for QKD channels and gate-based quantum computers using singular value decomposition (SVD) of noisy matrices. This method provides a robust framework for noise mitigation, ensuring improved performance of QKD protocols in real-world scenarios.

Beyond noise mitigation, our ongoing research focuses on developing a novel quantum key distribution protocol that leverages statistical transformations of measurement deviations between communicating parties. By utilizing the standard deviation of unsupervised measurements, we aim to design an new QKD protocol that enhances security and resilience against noise. Additionally, we intend to integrate machine learning-assisted techniques to create a noise-resistant QKD SDK framework, further strengthening the security and reliability of quantum communication.

The integration of machine learning into quantum communication represents a promising direction for improving the efficiency of quantum key distribution. By leveraging machine learning and statistical models, we can enhance error mitigation mechanisms and develop adaptive protocols capable of withstanding environmental perturbations. Our goal is to refine existing quantum key distribution techniques while introducing innovative methodologies that align with the evolving

landscape of quantum technologies.

As quantum computing continues to advance, the necessity for secure communication channels becomes increasingly critical. Quantum key distribution offers an unparalleled level of security, ensuring that information remains protected against eavesdropping and unauthorized access. However, achieving practical scalability and reliability requires addressing the inherent challenges associated with quantum noise, environmental interactions, and error correction. Through our research efforts, we aim to contribute to the development of more robust and scalable quantum communication technologies, paving the way for widespread implementation in secure data transmission.

In conclusion, quantum communication stands at the forefront of technological advancements in secure information exchange. The ongoing research in noise identification, mitigation strategies, and machine learning-assisted QKD protocols, By bridge theoretical insights with experimental implementations. Our work represents a step toward achieving this goal, ensuring that quantum key distribution remains a viable and effective solution for future secure communications.

Chapter 2

Preliminaries

2.1 Information and Quantum Information

Classically Information can be transformed from one party to another through a optical cable in the form of an electrical signal. In quantum information, data can be encoded in a polarized photon and transferred between parties, a process known as quantum communication.

2.1.1 Bits and Qubit

Bit is the smallest unit of digital information and is represented by 0 and 1 [9]. qubits are quantum analogous to the classical bit used in quantum computing. Physically we can represent this as a two-level system. with states $|0\rangle$and$|1\rangle$ where

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

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

These are called standard basis in quantum computing. Sometimes state can be in a superposition of the $|0\rangle$and$|1\rangle$states. which are $|+\rangle$and$|-\rangle$.

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

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

2.1.2 Quantum Gate

Quantum operator are used to represent the change of the physical state. Qubit states can be transformed by using unitary operators, often termed quantum gates[9] which are analogs to classical digital gates. Here are some widely used quantum gates.

1. **H gate**

   The Hadamard gate transforms the basis states as follow

$$H = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

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

2. **X** gate

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

$$\mathrm{X}|0\rangle = |1\rangle$$
$$\mathrm{X}|1\rangle = |0\rangle$$

3. **Z** gate(Phase gate)

$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

$$\mathrm{Z}|+\rangle = |-\rangle$$

4. **Y gate**

$$Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$$

X, Y, Z gates are known as Pauli gates. These gates rotate the qubit 180°around the respective axis of the bloch sphere.

2.2 Quantum Key Distribution Protocol

Quantum Key Distribution (QKD) protocols are methods for generating a secret key between two or more parties using polarized photons, based on the principles of quantum mechanics. Over the years, various key distribution protocols have been developed. In my study, I focus primarily on the preliminary protocols BB84 [5] and BBM92 [7]

2.2.1 BB84

BB84 is a quantum key distribution scheme proposed by Charles Bennett and Gilles Brassard in 1984, which allows two parties, Alice and Bob, to securely share a secret key. The protocol involves the following steps:

1. **Preparation:** Alice randomly generates a string of bits and encodes each bit into a qubit using one of two bases: the computational basis $\{|0\rangle, |1\rangle\}$ or the diagonal basis $\{|+\rangle, |-\rangle\}$.
2. **Transmission:** Alice sends the qubit to Bob over a quantum channel.
3. **Measurement:** Bob randomly chooses a basis (computational or diagonal) to measure each qubit he receives.

4. **Basis Reconciliation:** Alice and Bob publicly announce the bases they used for each qubit. They discard bits where their bases do not match.

5. **Key Sifting:** The remaining bits form the raw key. Alice and Bob perform error correction and privacy amplification to obtain the final secret key.

Let b_i be the i-th bit in Alice's string, and let θ_i be the basis used for encoding b_i. The qubit q_i is prepared as:

$$q_i = \begin{cases} |0\rangle & \text{if } b_i = 0 \text{ and } \theta_i = 0 \\ |1\rangle & \text{if } b_i = 1 \text{ and } \theta_i = 0 \\ |+\rangle & \text{if } b_i = 0 \text{ and } \theta_i = 1 \\ |-\rangle & \text{if } b_i = 1 \text{ and } \theta_i = 1 \end{cases} \tag{2.1}$$

Bob measures each qubit q_i in a randomly chosen basis ϕ_i. If $\phi_i = \theta_i$, Bob's measurement result will match b_i with high probability.

2.2.2 BBM92

The BBM92 protocol is an entanglement-based quantum key distribution (QKD) scheme that allows two parties, Alice and Bob, to securely share a secret key. The protocol involves the following steps:

1. **Entanglement Generation:** A source generates pairs of entangled qubits and sends one qubit from each pair to Alice and the other to Bob.

2. **Measurement:** Alice and Bob randomly choose one of two measurement bases (e.g., computational basis $\{|0\rangle, |1\rangle\}$ or diagonal basis $\{|+\rangle, |-\rangle\}$) to measure their qubits.

3. **Basis Reconciliation:** Alice and Bob publicly announce the bases they used for each qubit. They keep the results where their bases match and discard the rest.

4. **Key Sifting:** The remaining bits form the raw key. Alice and Bob perform error correction and privacy amplification to obtain the final secret key.

The entangled state shared between Alice and Bob can be represented as:

$$\{|\Phi^+\rangle, |\Phi^-\rangle, |\Psi^+\rangle, |\Psi^-\rangle\}$$

Their expansions in the computational basis are:

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

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

$$|\Psi^+\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle)$$

$$|\Psi^-\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle)$$

These Bell states are maximally entangled two-qubit states and are widely used in quantum information protocols such as quantum teleportation, superdense coding, and entanglement-based quantum key distribution.

Alice and Bob measure their qubit in randomly chosen bases. If their measurement bases match, their results will be correlated according to the entangled state.

2.3 Noise Channels

In quantum computing, noise can significantly affect the performance and reliability of quantum systems. Noise in the quantum channel leads to a change in state Which results in an incorrect key in the QKD protocol. Here are some common types of noise errors:

2.3.1 Bit-Flip

A bit-flip error occurs when a qubit state flips from $|0\rangle$ to $|1\rangle$ or vice versa. This is analogous to a classical bit error where a 0 becomes a 1 or a 1 becomes a 0. The bit-flip error can be represented by the Pauli-X operator **X**.The Kraus operators describe how the quantum state evolves when interacting with a noisy environment. The Kraus operators defining this transformation are given as follows [9]:

$$E_0 = \sqrt{\lambda_{bp}}\mathbb{I};\ E_1 = \sqrt{1-\lambda_{bp}}\sigma_x \tag{2.2}$$

2.3.2 Depolarization

Depolarization noise which leads to the change in the qubit state as follows[9]:

$$\Lambda_{dp} = \lambda_{dp}\rho + (1-\lambda_{dp})\mathbb{I} \tag{2.3}$$

The depolarizing noise can be seen that takes one quantum state ρ to a linear combination of itself and the maximally mixed state $\mathbb{I}$, based on the strength parameter λ_{dp}.

$$E_0 = \sqrt{1-\frac{3p}{4}}\mathbb{I};\ E_1 = \sqrt{\frac{p}{4}}\sigma_x;\ E_2 = \sqrt{\frac{p}{4}}\sigma_y\ E_3 = \sqrt{\frac{p}{4}}\sigma_z, \tag{2.4}$$

2.3.3 Amplitude Damping

The amplitude damping noise channel models the energy dissipation in a quantum channel. The kraus operators for the transformation are represented as follows [9].

$$E_0 = \begin{pmatrix} 1 & 0 \\ 0 & \sqrt{1-p} \end{pmatrix};\ E_1 = \begin{pmatrix} 0 & \sqrt{p} \\ 0 & 0 \end{pmatrix} \tag{2.5}$$

2.3.4 Quantum bit error rate

Quantum channel used by Alice and Bob for QKD protocols are afflicted with noise, resulting key generated at Alice's end will differ from key generated at Bob's end. The metric defined to

measure this error is called Quantum bit error rate (QBER), and is given by,

$$\text{QBER} = \frac{\text{Number of erroneous key bits}}{\text{Number of same measurement bases for Alice and Bob}} \tag{2.6}$$

$$\text{QBER (for BBM92)} = \frac{\text{Number of erroneous key bits for either Alice or Bob}}{\text{Number of same measurement bases for Alice and Bob}} \tag{2.7}$$

2.4 Machine Learning Tools

Machine Learning algorithms learn from data and used to predict the outcome without explicitly programmed. Supervised machine learning is a class of problems where you can classify the unlabeled data. Here we take a data set with known feature values, label them corresponding classes and use these data to predict the class of the new dataset. Test set is the Subset of data which is used to identify the pattern and train the model to evaluate the result. Data can be represented in a two dimensional space. The co-ordinate of the data points are called feature. Feature is an individual measurable property or characteristic of a data set.

2.4.1 K-nearest Neighbor Classifier

The K-Nearest Neighbors (KNN) algorithm [2, 10, 11] is one of the most basic yet effective methods used in machine learning for both classification and regression tasks. This algorithm operates on the assumption that data points belonging to the same class are clustered together and here, the dataset is classified using two features. It classifies a new data point by examining the K nearest neighbors and assigning the majority class among them. The value of K depends on the dataset; if the data contains significant noise or outliers, choosing a higher K value is preferable. Additionally, it is recommended to use an odd value for K to minimize ties in classification.

2.4.2 Linear Support Vector Machine

The Linear Support Vector Machine (SVM) algorithm [4, 12, 13, 14] works by selecting a hyperplane or line that separates two classes of data points. The chosen boundary effectively distinguishes between the classes. It determine the optimal decision boundary by considering the

extreme data points, known as support vectors. Support vectors are data points that are close to the opposing class. Margin is the distance between support vectors of each class. SVM can choose the best hyperplane that maximizes the margin. It is also called Maximum Margin Classifier.

2.4.3 Gaussian Naive Bayes Classifier

The Gaussian Naïve Bayes model [3, 15, 16] is based on features that follow a Gaussian distribution. This method is called 'naïve' because it assumes that the variables or features are independent, which is rarely the case. To apply this model, calculate the mean, standard deviation, and variance of each feature in the dataset. Using the probability density function, the algorithm computes the likelihood (posterior probability) for each test sample across all features in the dataset. Finally, the likelihood values, along with an initial prior assumption, are used to make predictions.

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{2.8}$$

2.4.4 Principle Component Analysis

Principle component analysis [17] or PCA reduce the dimensionality of a large data set to principle component that retain most of the original information. The first principle component will be on the x axis and the second one on the y axis. Mostly used in data pre-processing for machine learning algorithm and application. PCA helps retain the most critical features from a large dataset, preserving essential information while reducing complexity, which can improve model performance. This principle components are linear combinations of original variables that have maximum variance compared to other linear combinations. The first principal component captures the highest variability in the dataset, preserving the most information. No other principal component can exceed its variability, while the second principal component (PC2) accounts for the next highest variance and remains uncorrelated with PC1.

2.5 Distribution Parameters

In probability and statistics, distribution parameters are values that define the characteristics of a probability distribution. These parameters determine the shape, spread, and central tendency of the distribution.

1. **Mean**
   Average of the distribution.

$$\bar{X} = \frac{\sum_{i=1}^{n} x_i}{n} \tag{2.9}$$

   where: $\bar{X}$ is the mean
   x_i is each value in the dataset
   n is the number of values in the dataset.

2. **Median**
   The most frequent value in a distribution. mode is the middle value in a distribution: The formula for the median depends on whether the dataset has an odd or even number of observations. For an even number of observations:

$$\text{Median} = \frac{x_{\left(\frac{n}{2}\right)} + x_{\left(\frac{n}{2}+1\right)}}{2} \tag{2.10}$$

   For odd number of observations:

$$\text{Median} = x_{\left(\frac{n+1}{2}\right)} \tag{2.11}$$

   where:
   x_i represents the i-th ordered value in the dataset,
   n is the number of observations in the dataset.

3. **Mode**
   The middle value of the distribution if arranged in increasing or decreasing order.

4. **Standared Deviation**

Measure that indicate how much data scatter around the mean.

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2} \tag{2.12}$$

Where:
σ is the standard deviation
N is the number of data points
x_i is each individual data point
μ is the mean of the data.

5. **Area under the curve**

$$A = \int_a^b f(x)\,dx \tag{2.13}$$

where:
A is the area under the curve
$f(x)$ is the function representing the curve
a and b are the limits of integration.

6. **Kurtosis** It is a measure the peakness of a frequency distribution.Kurtosis helps to understand the tailedness or sharpness of the peak of a distribution.

$$K = \frac{n\sum_{i=1}^{n}(x_i - \bar{x})^4}{\left(\sum_{i=1}^{n}(x_i - \bar{x})^2\right)^2} \tag{2.14}$$

n is the number of data points
x_i are the data points
$\bar{x}$ is the mean of the data points.

7. **Skewness**
Skewness is the measure of asymmetry or bias of the distribution. It helps in understanding the direction and extent of the deviation from normal distribution.

$$S = \frac{n\sum_{i=1}^{n}(x_i - \bar{x})^3}{\left(\sum_{i=1}^{n}(x_i - \bar{x})^2\right)^{3/2}} \tag{2.15}$$

n is the number of data points
x_i are the data points

$\bar{x}$ is the mean of the data points.

2.6 IBM Qiskit

Qiskit [18] is designed for working with quantum computers at the level of circuits, algorithms, and application modules. It is a high-performance software stack built to help developers and researchers harness the full power of quantum computers at the utility scale and beyond. The SDK includes high-performance tools and services such as the Qiskit Runtime Service, which enables optimized computations on IBM Quantum computers through the cloud using primitives. The Qiskit Transpiler Service provides state-of-the-art heuristics and AI-powered methods that enhance performance for common quantum circuit optimization tasks. Qiskit functions as a catalog of IBM and third-party services

2.7 Singular Value Decomposition of Random Matrix

Every $m \times n$ matrix factors in to

$$A = U\Sigma V^T$$

[19]

$$U = \begin{bmatrix} \uparrow & & \uparrow \\ \mathbf{u_1} & \dots & \mathbf{u_m} \\ \downarrow & & \downarrow \end{bmatrix}, \quad \Sigma = \begin{bmatrix} \sigma_1 & 0 & \dots & 0 \\ 0 & \sigma_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sigma_r \end{bmatrix}, \quad V = \begin{bmatrix} \leftarrow \mathbf{v_1} \rightarrow \\ \vdots \\ \leftarrow \mathbf{v_n} \rightarrow \end{bmatrix}$$

where:

- U is an $m \times m$ orthogonal (or unitary) matrix.
- Σ is an $m \times n$ diagonal matrix of singular values.
- V is an $n \times n$ orthogonal (or unitary) matrix.

where the singular values satisfy:

$$\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_r \geq 0.$$

- Unlike eigenvalues, singular values are always **real and non-negative**, making them useful in applications like image compression and noise reduction.

2.8 Bayesian Optimization

Bayesian Optimization is a method of optimizing a black box function by creating a surrogate function from randomly chosen data points [20]. An additional data point can be added using an acquisition function, and the surrogate function is then re-evaluated. This loop continues until the maximum of the surrogate function does not change, the variance falls below a threshold, or the function is exhausted. This method helps find the minimum or maximum of a black-box function. The acquisition function finds the optimal point by balancing the trade-off between exploration and exploitation.

Chapter 3

Detection of Noise in QKD Protocols from QBER

Noise in the quantum key distribution (QKD) protocol introduces errors in the generated key. The proposed protocol considers two different QKD schemes under three distinct noise channels in various scenarios. I use the Quantum Bit Error Rate (QBER) as a metric to assess noise in a QKD system, focusing primarily on the BB84 and BBM92 protocols.

3.1 Noise Classification in Quantum Channel

In this work, I consider three different noise channels that affect the key distribution scheme: amplitude damping noise, bit-flip noise, and depolarizing noise. For this analysis I use operator-sum representation of quantum noise channels that models the impact of noise on a quantum state ρ as follows,

$$\mathscr{E}(\rho) = \sum_k E_k \rho E_k^\dagger, \tag{3.1}$$

[9] where, operators E_k are the operational elements satisfying $\sum_k E_k^\dagger E_k = \mathbb{I}$. The complete set of operators $\{E_k\}$ provides a framework for analyzing different quantum noise effect in the Hilbert space of the quantum state itself. This work uses the operator sum representation to compute the effect of different noise channels on the QKD schemes in different scenarios by employing the kraus operators for different noise. consider the impact of noise in quantum channel during the

transfer of qubits from Alice to Bob. Assuming no noise is present in the preparation and measurement stage of qubits. I simulated 16 qubit BB84 protocol under the impact of bit-flip and amplitude damping noise under operator-sum representation of both the noise channel in python.

The transformation of each qubit state in BB84 protocol under different noise channel and measurement basis is shown in the tables 3.1 and 3.2

Bit	Alice's basis	ρ	$\mathcal{E}(\rho)$	Bob's basis	RP('0')	RP('1')
'0'	Computational	$\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 1-p & 0 \\ 0 & p \end{pmatrix}$	Computational	$1-p$	p
'1'	Computational	$\begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$	$\begin{pmatrix} p & 0 \\ 0 & 1-p \end{pmatrix}$	Computational	p	$1-p$
'0'	Diagonal	$\frac{1}{2}\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$	$\frac{1}{2}\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$	Diagonal	1	0
'1'	Diagonal	$\frac{1}{2}\begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}$	$\frac{1}{2}\begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}$	Diagonal	0	1

Figure 3.1: BB84 QKD protocol: State transformation when basis matches under *Bit-flip* channel, RP('0')-retrieval probability of Bob's key as '0', RP('1') retrieval probability of Bob's bit being'1', p represents the noise strength parameter

Bit	MB	ρ	$\mathcal{E}(\rho)$	RP('0')	RP('1')
'0'	C	$\|0\rangle\langle 0\|$	$\begin{pmatrix} 0.5 & 0 \\ 0 & 0.5 \end{pmatrix}$	0.5	0.5
'1'	C	$\|1\rangle\langle 1\|$	$\begin{pmatrix} p & 0 \\ 0 & 1-p \end{pmatrix}$	p	$1-p$
'0'	D	$\|+\rangle\langle +\|$	$\frac{1}{3}\begin{pmatrix} 1+p & \sqrt{p}+\sqrt{1-p} \\ \sqrt{p}+\sqrt{1-p} & 2-p \end{pmatrix}$	$\frac{1}{2}+\frac{\sqrt{p}+\sqrt{1-p}}{3}$	$\frac{1}{2}-\frac{\sqrt{p}+\sqrt{1-p}}{3}$
'1'	D	$\|-\rangle\langle -\|$	$\frac{1}{3}\begin{pmatrix} 1+p & -(\sqrt{p}+\sqrt{1-p}) \\ -(\sqrt{p}+\sqrt{1-p}) & 2-p \end{pmatrix}$	$\frac{1}{2}-\frac{\sqrt{p}+\sqrt{1-p}}{3}$	$\frac{1}{2}+\frac{\sqrt{p}+\sqrt{1-p}}{3}$

Figure 3.2: BB84 QKD protocol: State transformation when basis matches under *Amplitude damping channel*, RP('0')-retrieval probability of Bob's key as '0', RP('1') retrieval probability of Bob's bit being'1', p represents the noise strength parameter

From the table it is evident that, due to the noise in the quantum channel, Bob has a finite probability of measuring the key bit as '1', despite his measurement basis being same as Alice also One can also infer from fig. 3.1 and fig. 3.2, that the errors generated in Bob's key can differ if the underlying noise in the quantum channel is different.

The method of classification of quantum noise channel from QBER is as follows.

1. Simulated secret key through BB84 protocol under different noise effects using operator-sum representation on density matrix of the quantum state. To generate a key, the process is repeated 16 times and computed the QBER as given.

2. Repeated the key generation process 200000 times for each noise type with random noise strength of the channels for each transfer, accumulating a list of 200000 QBER values for both types of noises.

3. Created individual histograms with 10 bins for QBERs generated due to amplitude damping noise and bit-flip noise, by dividing our data into multiple blocks (say m) for each noise, such that, each histogram contains $n = \frac{200{,}000}{m}$ QBER distributions.

4. Plotted the average of all the histogram for bit-flip and amplitude damping noise. The distributions, on their own, are not distinguishable properly fig 3.3.

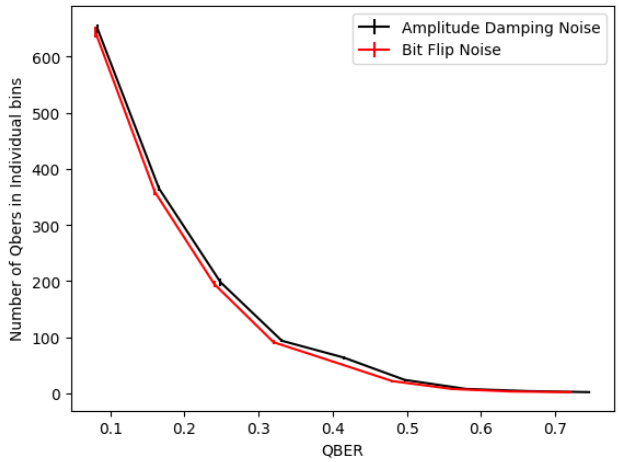


Figure 3.3: Average Histogram distribution of QBERs (only tip is shown) with bit-flip and amplitude damping noise are applied to a quantum channel

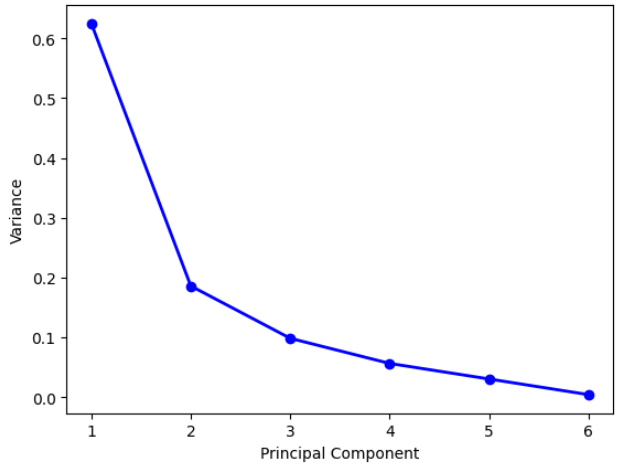


Figure 3.4: PCA of the data set is presented. In this dataset, the first three elements can explain 91% of the data-variance.

5. Chosen seven different features of the histogram distributions as an input to our classification model- mean, median, mode, standard deviation, skewness, kurtosis, and area under the curve and generated a labeled dataset of 2 × m data.

6. To increase the number of data points in our dataset, with no loss of generality, we shuffled the original dataset randomly 100 times, and repeated the histogram analyses, to finally create a larger dataset of 2m × 100 entries.

7. Calculated the PCA of the data set and used the first three principle component to train the supervised machine learning models, fig 3.4.

8. Dataset is divided into training set and test set with a 7:3 ratio. Applied three different supervised classification models; K-Nearest Neighbor (KNN), Gaussian Naive Bayesian (GNB), and Support Vector Machine (SVM) on the training set.

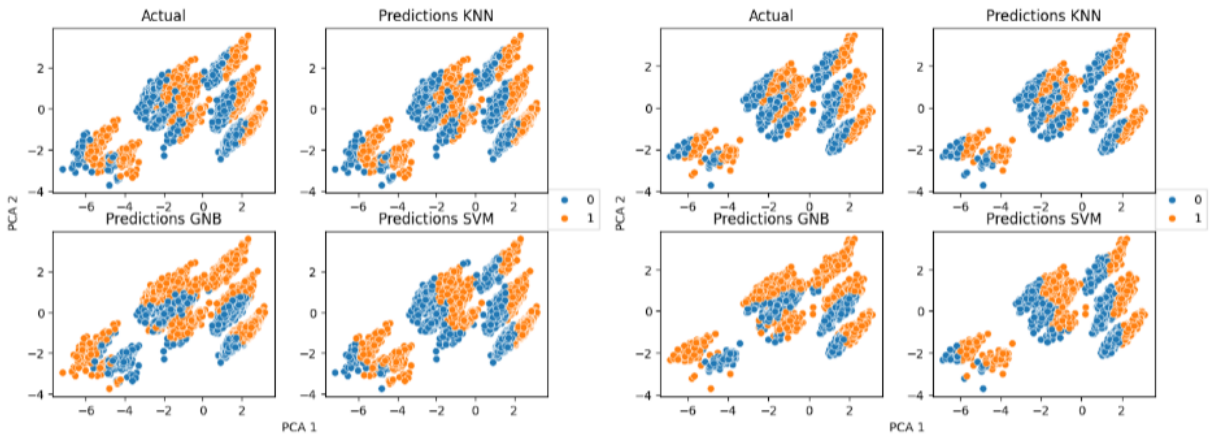
9. Applied the models on the test dataset to compare the accuracy of classification. Represented the accuracy of our classification method for training and test datasets.

3.1.1 Result and Discussion

The Accuracy of training and Test data set for different values of *m* and *n* are shown in the fig 3.5

# Data-instances	# QBERs per histogram	ML model	Train Acc	Test Acc
40000	1000	KNN	87.9%	77%
		GNB	71.9%	72%
		SVM	77.1%	77.2%
20000	2000	KNN	92%	87%
		GNB	58.5%	56.9%
		SVM	82.9%	82.5%
10000	4000	KNN	97%	96%
		GNB	67%	64%
		SVM	92.5%	91.5%

Figure 3.5: Accuracy comparison for quantum noise channel classification with classical machine learning model, k = 2, and SVM with rbf kernel, degree 4 an GNB for all the analysis



(a) Training data (b) Test Data

Figure 3.6: Classification of quantum noise channels in QKD (BB84) protocol for training(a) and test data(b), trained by K-Nearest Neighbors (KNN), Gaussian Naive Bayes (GNB), and Support Vector Machine (SVM) classifiers. 0' and '1' respectively represent amplitude damping noise and bit-flip noise. Here m and n used is 50 and 4000 respectively. PCA1 and PCA2 refers to the Principal Components of the data set

Here, Alice and Bob are remotely located, and they communicate through a quantum channel.

From the fig 3.5 it is evident that, the accuracy of the training and test data depends on the *m* and *n* value. However, KNN with k = 2 provides the best results for both training and test cases with 97% accuracy for training set and 96% accuracy for the test set, when the dataset contains a total of 5000 instances per noise, but each QBER histogram distribution contains 4,000 entries. For the other two cases as well, we see KNN provides better accuracy than SVM (rbf kernel, degree=4) and GNB. From fig 3.6, we could observe that the training and test data set are overlapping. For overlapping data set KNN provide better accuracy.

histogram distribution of quantum bit error rates (QBERs) originating due to amplitude damping and bit-flip noises have overlapping characteristics, leading to classification with high accuracy possible only with K-nearest neighbor classifier. Further, we have shown that the classification accuracy increases if we assign more data to analyze the histogram features, instead of having a larger dataset.

3.2 Noise Classification in Gate- based Quantum Computers

Gate based quantum computing uses quantum gates to solve problems by levaraging space and time advantages of quantum algorithms over classical algorithms. Here I present the protocol designed to classify the *Bit-flip* and *depolarisation* noise in Preliminary QKD protocols, BB84 and BBM92.

Here Alice and Bob posses two different qubits of a noisy gate-based quantum computer

The fig 3.7 shows the protocol for classification.

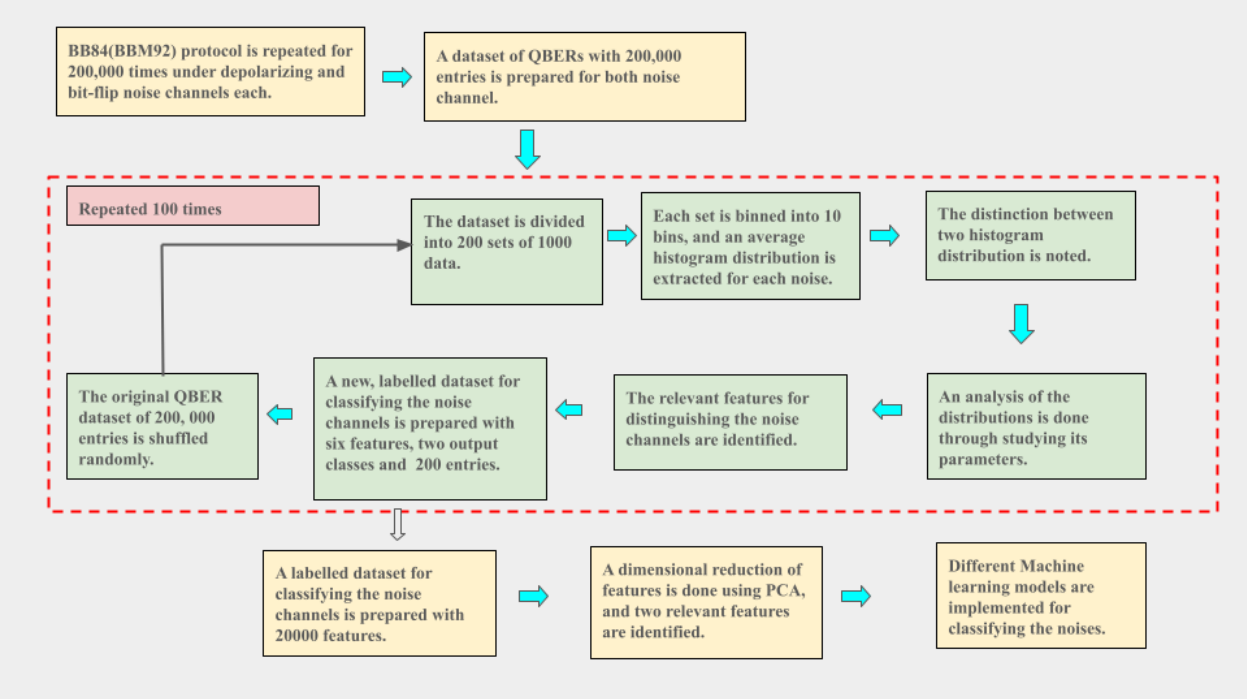


Figure 3.7: Novel protocol for classifying the noise channels in QKD in gate based quantum computer

3.2.1 Methodology

1. Implemented the 16 qubit BB84 and BBM92 quantum key distribution protocols in Qiskit SDK offered by IBM and generated the final secret key.

2. Added the depolarizing and bit-flip noise model in the key distribution protocol and generated the erroneous key.

3. Computed the QBER value using the actual and incorrect keys for the BB84 and BBM92 . Plotted the QBER distribution for 1,000 QBER values using 10 bins for both noise models.

4. Plotted the average histogram of 200 QBER distributions for both QKD protocols and analyzed the distribution differences for each noise model.

5. Calculated the seven distribution parameters(mean, median, mode, standard deviation, skewness, kurtosis, area under the curve) for both distribution.

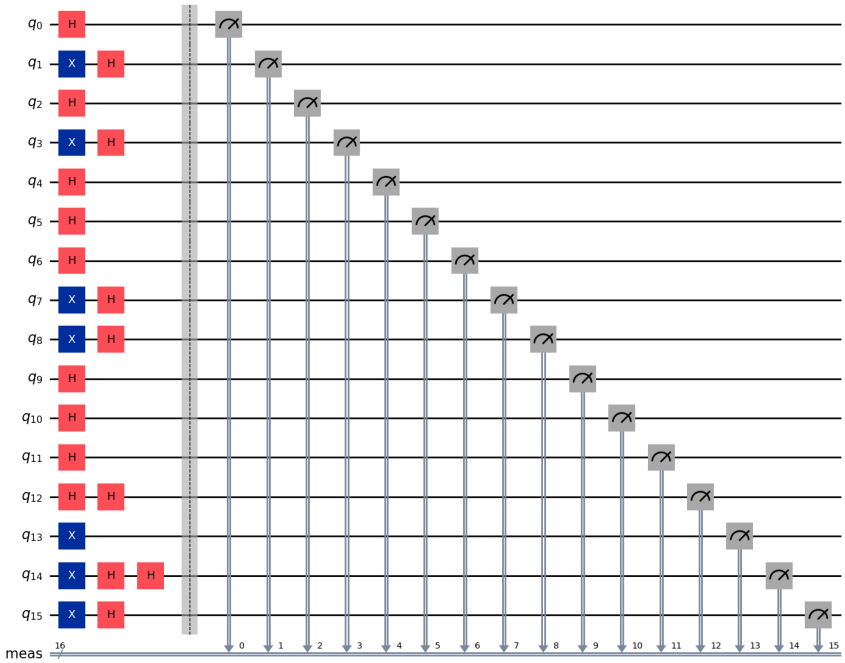


Figure 3.8: 16 qubit BB84 protocol simulated in IBM Qiskit for a random key generation

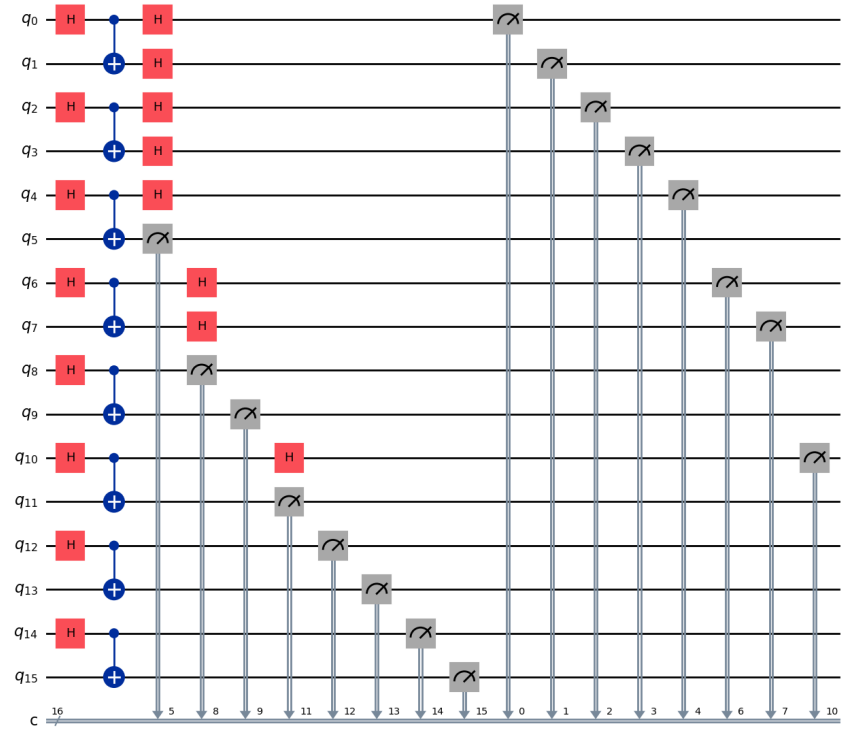


Figure 3.9: 8 bell pair BBM92 protocol simulated in IBM Qiskit for a random key generation

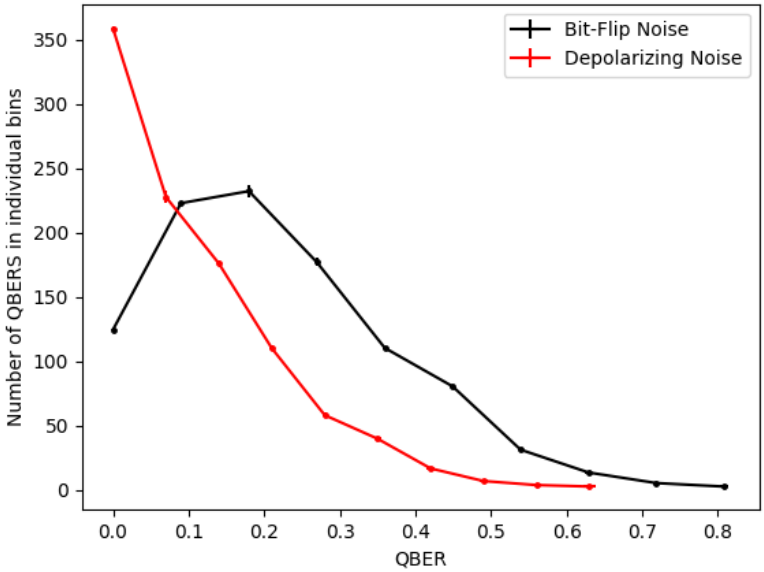


Figure 3.10: BB84 QKD: Average of 200 QBER distribution for bit-flip and depolarising model with 10 bins, each distribution conatins 1000 QBER values.

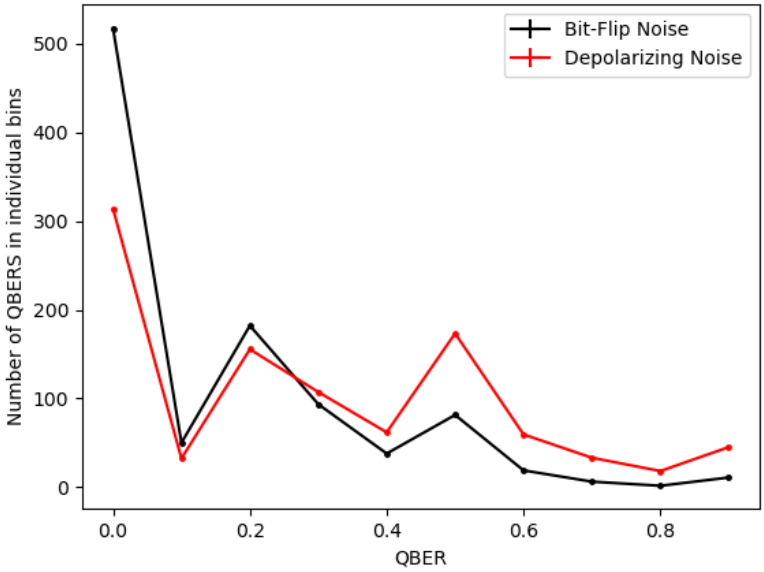


Figure 3.11: BBM92 QKD: Average of 200 QBER distribution for bit-flip and depolarising model with 10 bins, each distribution contains 2000 QBER values.

6. Eliminated the mean for future studies, as it is similar for both distribution.

7. Performed the principal component analysis on the 6 featured to produce the two principal components. which was used to train the machine learning model.

8. 70% of the data is kept as a training set and 30% of the data set is kept test set.

9. Trained the Gaussian Naive Bayes classifier, K- Nearest neighbors, Support vector machine models using the data sets.

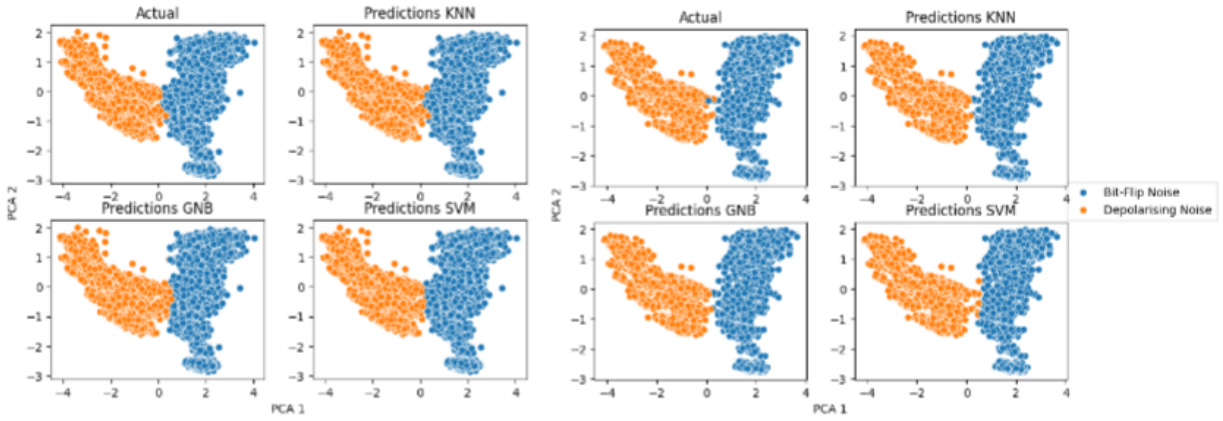
10. Classified the depolarizing and bit-flip noise for the BB84 and BBM92 quantum key distribution protocol.

3.2.2 Result and Discussion

As amplitude damping was not readily available in Qiskit at the time of this work, it was not included. Readout error was also not considered when simulating the QKD. Here, the difference in distribution for both noise effects is distinguishable for both noise models, unlike in a quantum channel. This difference arises due to the distinct sources of noise, primarily originating from errors introduced during gate operations.

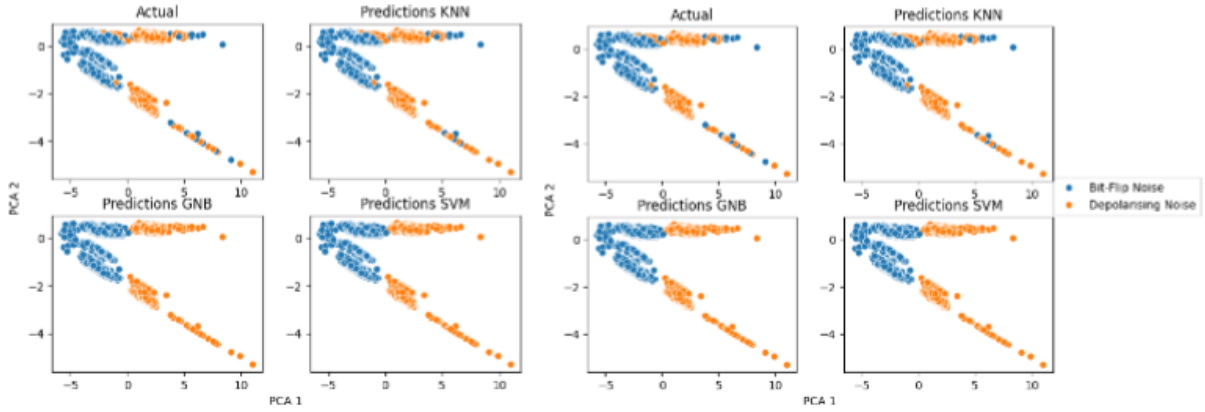
The first principal component covers 93% of the variation. We trained KNN (k=2), Linear SVM, and GNB classifiers using a training set of 14,000 data points and tested them on a dataset of 6,000 data points. For BBM92, a dataset of 40,000 data points was used.

Fig 3.14 represents the accuracy of the training set and test set for both QKD protocols, showing 99.9% accuracy on both the test dataset and training dataset for classifying the noise. The high accuracy on the test set is due to the distinguishable differences between the statistical parameters for each class in both protocols



(a) Training data (b) Test Data

Figure 3.12: BB84: Trained K-Nearest Neighbors (KNN), Gaussian Naive Bayes (GNB), and Support Vector Machine (SVM) classifiers. PCA1 and PCA2 refers Principal Component Components. '0' and '1' represent amplitude damping noise and bit-flip noise respectively.



(a) Training data (b) Test Data

Figure 3.13: BBM92 Classification of test-set data with trained K-Nearest Neighbors (KNN), Gaussian Naive Bayes (GNB), and linear Support Vector Machine (SVM) classifiers. PCA1 and PCA2 refers to the Principal Components.

QKD	# Data-instances	# QBERs per histogram	ML model	Train Acc	Test Acc
BB84	20000	1000	KNN	99.9%	99.9%
			GNB	99.9%	99.8%
			SVM	99.8%	100%
BBM92	40000	2000	KNN	99.8%	99.7%
			GNB	99.7%	99.6%
			SVM	99.7%	99.6%

Figure 3.14: Accuracy comparison for quantum noise model classification with classical machine learning models. We have used the KNN model with k = 2, and SVM with linear kernel for all the analysis.

In case of gate-based quantum computer, where the noises attacks only the gates used, the distributions have non-overlapping features, leading to even a 100% accuracy with liner SVM. We have also shown that this observation is independent of the choice of QKD protocol. In future, we plan to extend our study to mixed noise channels, as well as study the performance of our protocol using real data.

Chapter 4

Noise Mitigation Strategies According to the Detected Noises

Earlier methods shows how to classify the noise in QKD. Here I explains our new method to mitigate the noise based on the identified noise. This helps to increase the efficiency of the protocol for generating key.

4.1 Mitigation of Noise in Quantum Channel

The following method describe the way in which the we can mitigate the noise in QKD by generating the large number of key.[1]

1. Generate the ideal key matrix of m rows and l column. Each row represents a key of l elements.
2. Assign noise strength parameter value to a variable p. Implement the noisy key generation using operator-sum of depolarization noise over the density matrix formalism of quantum state leading to a noisy key matrices of m rows and l column.
3. Create n noisy matrices by using this method.
4. Take the zeroth row of n matrix and create another matrix by putting it in each row. which

creates a $n \times l$ matrix. Did this for all the row of $m \times l$ matrix leading to form m matrix of $n \times l$ dimension.

5. Perform singular value decomposition of each matrix and reconstruct the matrix by keeping only the first singular value of the diagonal matrix.

6. Taken the average of each column of all the reconstructed matrix which gives m rows. Create the $m \times l$ denoised matrix.

7. Fix the cut off value. Transform all the element in the final matrix to 0 if it is less than cutoff and to 1 if it is equal or above than the cut off.

8. Measure the positional ambiguity which is the equivalent of qber in term of matrix of keys. Which is Number of mismatched values between ideal matrix and the final matrix divided by the total number of elements in the key.

4.1.1 Result and Discussion

I have plotted the positional ambiguity under depolarizing noise for different values of the noise strength parameter, the number of noisy matrices, the key length, and the number of keys in a noisy matrix (p, n, l, m).

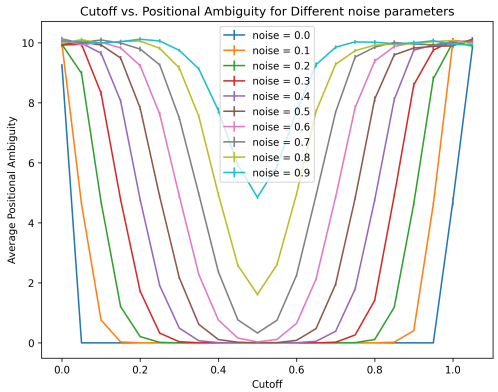


Figure 4.1: Depolarization noise: change of average positional ambiguity when the cut off varies, at noise strength parameter changing from 0.0 to 1 with an interval of 0.1.

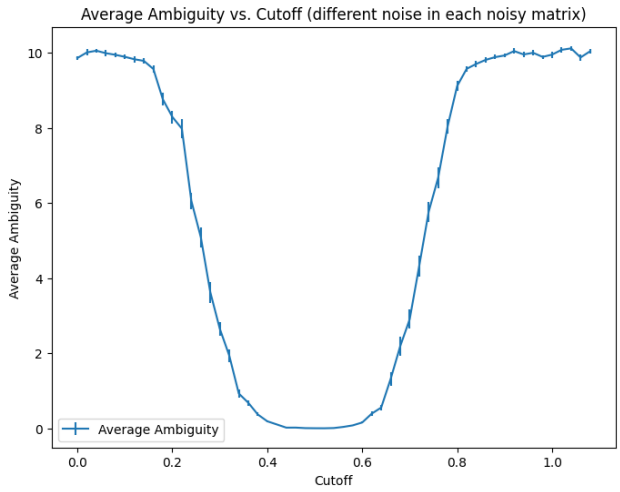


Figure 4.2: Variation in positional ambiguity vs cutoff when different noise strength parameters are randomly assigned to 50 noisy matrices

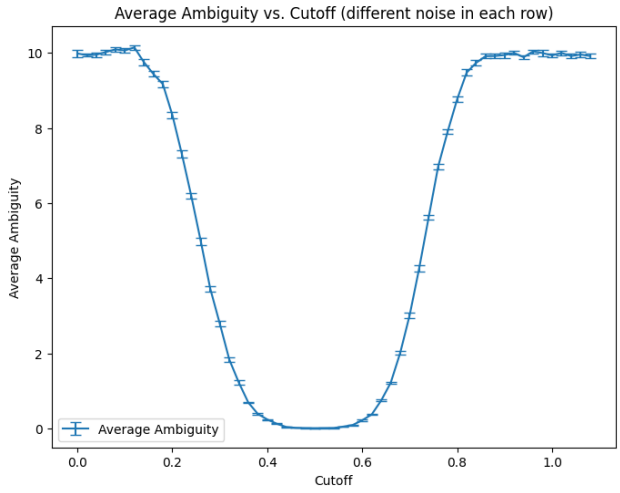


Figure 4.3: Variation in positional ambiguity vs cutoff when different noise strength parameters are randomly assigned to each key in all noisy matrices.

Figures 4.1, 4.2, 4.3, and 4.4 shows that a cutoff of 0.5 results in the minimum positional ambiguity, regardless of the noise strength parameter and its distribution. Additionally, the positional ambiguity increases with the noise strength parameter, as expected.

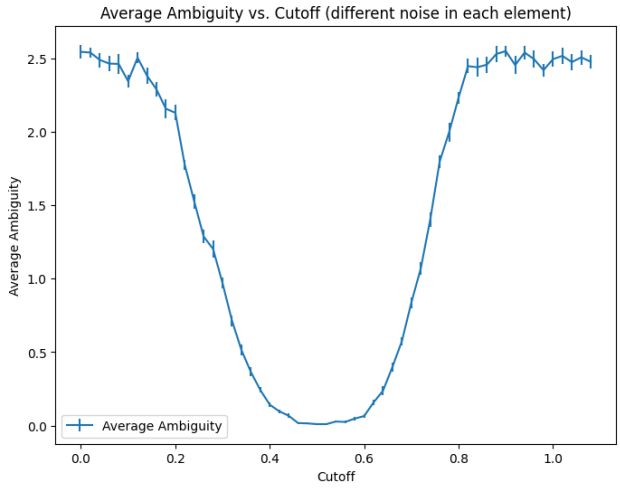


Figure 4.4: Variation in positional ambiguity vs cutoff when different noise strength parameters are randomly assigned to each element of each key in every noisy matrix.

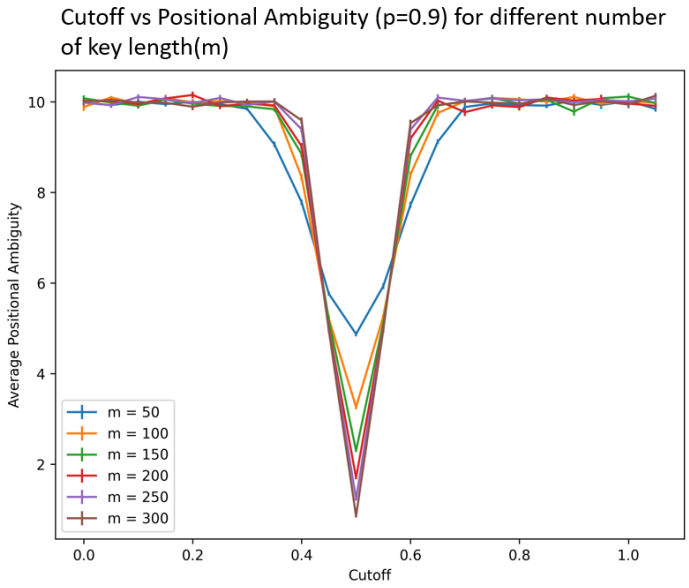


Figure 4.5: Depolarization noise (p = 0.9): Average positional ambiguity vs cutoff as the number of keys (m) varies from 50 to 300 in increments of 50.

Figure 4.5 shows that even with different numbers of keys in a noisy matrix, a cutoff of 0.5 results in the minimum positional ambiguity. As the m values increase the average positional ambiguity decreases.

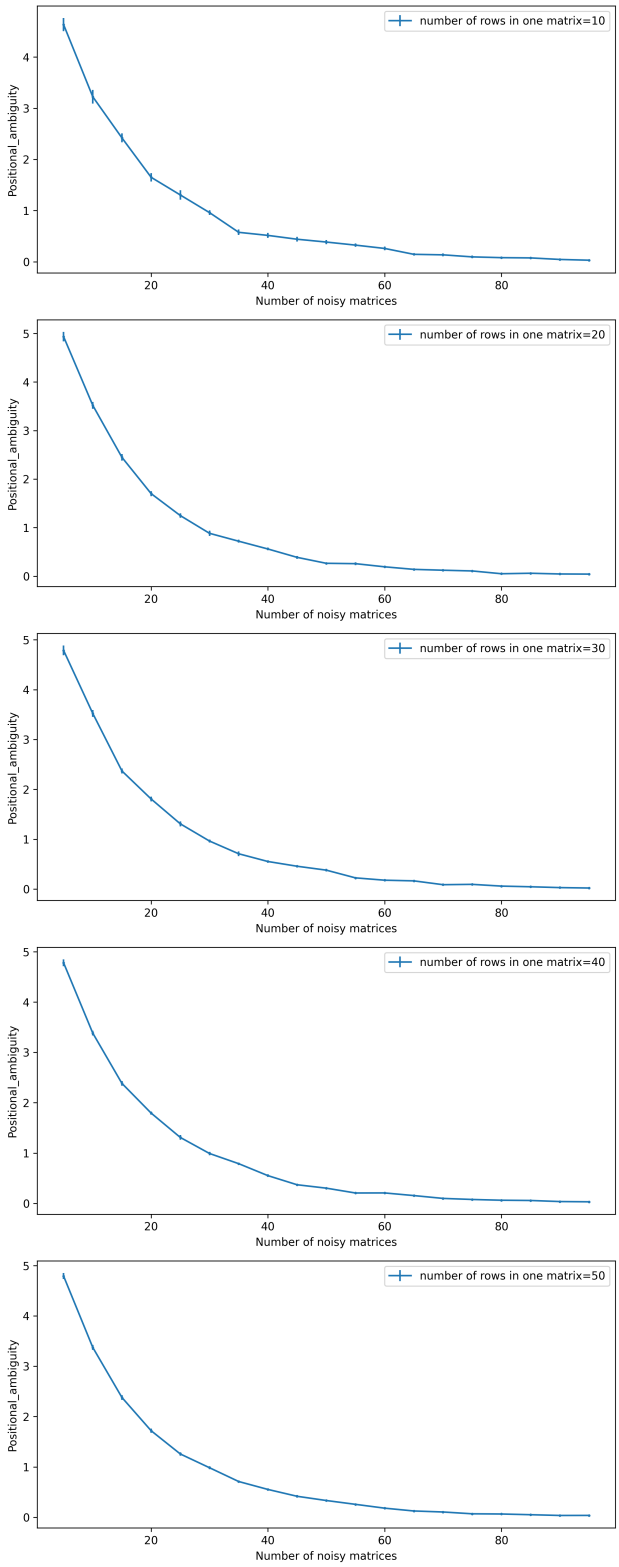


Figure 4.6: Depolarization noise:p=0.7, Positional ambiguity when the number of matrix n varies from 10 to 50.

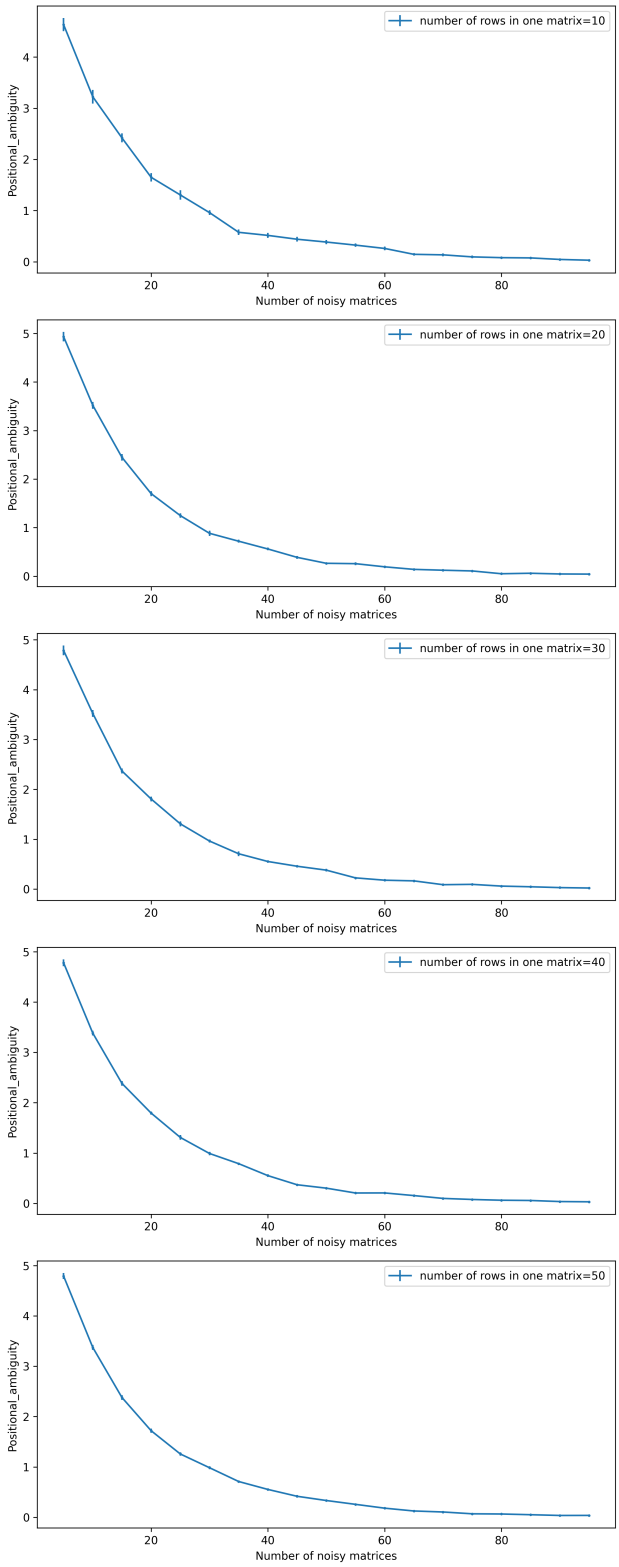


Figure 4.7: Depolarization noise:p=0.8, Positional ambiguity when the number of matrix n varies from 10 to 50.

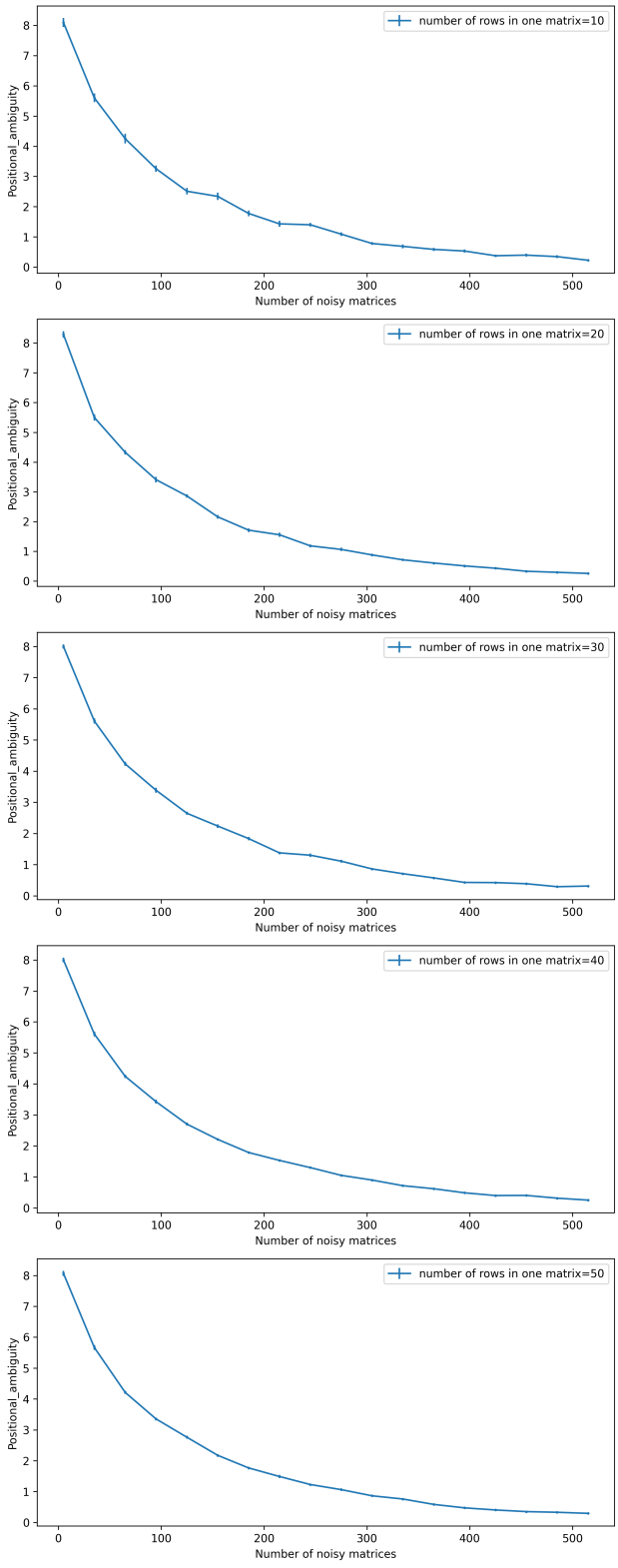


Figure 4.8: Depolarization noise:p=0.9, Positional ambiguity when the number of matrix n varies from 10 to 50

To achieve minimum positional ambiguity, the number of noisy matrices should be at least 60 or more for a noise strength parameter p= 0.7. As p increases the number of noisy matrices required to minimize error also increases. For p= 0.8, n = 80 and p=0.9, n=400.

The BB84 and BBM92 protocol in quantum channel has gaussian noise distribution. During the singular value decomposition of the noisy matrix each singular value signifies the Certain degree of information about the matrix including certain amount for noise. During reconstruction we have to identify that which all eigen values we have to choose in order to minimize the noise. i.e We have to choose the singular values which has gaussian noise which give results mean 0, which is zero noise. In this case the first element it self give the major information component with least noise, resulting least positional ambiguity, creating a efficient method of mitigation can be performed independendly regarding the type of noise.

4.2 Mitigation of Noise in Gate-based Quantum Computer

As mentioned earlier, noise mitigation in QKD protocols on gate-based quantum computers can be achieved using singular value decomposition and noisy matrices, followed by Bayesian optimization.

4.2.1 Methodology

1. Implementing BB84 in IBM qiskit model and Generated the ideal key matrix of m rows and l column by padding with 0's when a key was shorter than the maximum key length.

2. Generate n noisy key matrices by applying the noise model to the protocol. Create n noisy matrices by using this method.

3. Take the zeroth row of n matrix and create another matrix by putting it in each row. which creates a $n \times l$ matrix. Do this for all the row of $m \times l$ matrix leading to form m matrix of $n \times l$ dimension. Perform singular value decomposition of each matrix and reconstruct the matrix by keeping only the first singular value of the diagonal matrix.

4. Take the average of each column of all the reconstructed matrix which gives m rows. Create the $m \times l$ denoised matrix.

5. Fix the cut off value. Transform all the element in the final matrix to 0 if it is less than cutoff and to 1 if it is equal or above than the cut off.

6. Measure the positional ambiguity which is the equivalent of QBER in term of matrix of keys. Which is Number of mismatched values between ideal matrix and the final matrix divided by the total number of elements in the key.

7. Defined the cost function as positional ambiguity, dependent on cutoff and sigma values. Optimized it using Bayesian Optimization over 100 iterations.

4.2.2 Result and Discussion

With this method, the positional ambiguity reduced from 0.346 to 0.336 after 20 iterations and then leads to saturation. Due to the unsatisfactory results, I did not pursue this method further.

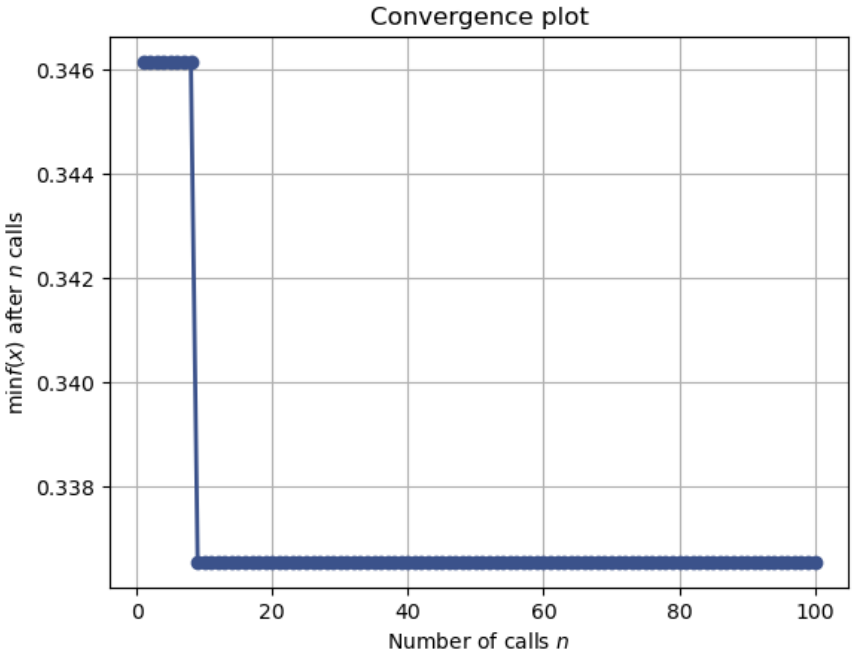


Figure 4.9: Variation in Positional Ambiguity with Increasing Iterations in Bayesian Optimization

This method of Singular value decomposition of the noisy matrix is not giving a good result in gate based quantum computer. So I performed Bayesian optimization to reduce positional ambiguity. But does not yield significant good results.

Chapter 5

Devising a Novel Quantum Key Distribution Protocol

5.1 Protocol

I am planning to propose a novel quantum key distribution protocol where the idea is to create a 3 qubit superposition state between alice and bob as shown here.

$$\frac{1}{\sqrt{8}}(|000\rangle + |001\rangle + |010\rangle + |011\rangle + |100\rangle + |101\rangle + |110\rangle + |111\rangle) \tag{5.1}$$

where Alice perform unsharp measurement with gaussian effect operator. and the standered deviation of the measurement outcome is shared between alice and bob which as the secret key. we hope that once formalized this protocol will provide better security than the customary protocol.

5.2 Methodology

Alice uses a combination of two gaussian effect operators to create an entangled state from given state it the Eq.5.1. Further she shares two of her three qubit to Bob and perform a sharp measurement. By choice Alice two gaussian have different variance which depending on her sharp

measurement, transmit to Bob. We are planning to use DQC1(Deterministic quantum computing with one qubit) for bobs measurement and retrieval of the variance. However it has not been fully developed yet.

Chapter 6

Conclusion and Future work

1. Developed a novel method for accurately classifying noise in quantum key distribution channels and gate-based quantum computers.
2. Designed an efficient technique to mitigate the identified noise in quantum key distribution channels and low efficient method in gate-based quantum computers.
3. Currently working on developing a new quantum key distribution protocol using unsharp measurements on a three-qubit entangled state.

Further study on each topic will be conducted.

1. Classification of mixed noise and Performance of protocol on the real data.
2. Identify why the mitigation protocol is not giving a good result in gate based quantum computer.
3. Development of novel QKD
4. Planning to create a Machine Learning Assisted noise resistant Quantum key distribution SDK, that integrate classification and mitigation protocol.

Two manuscripts explaining classification and mitigation of noise in Quantum Key distribution protocol are under construction.

Bibliography

[1] Kaushik Ray, Q. M. Jonathan Wu, Gaurab Basu, and Prasant K. Panigrahi. Random matrix route to image denoising. In *2012 International Conference on Systems and Informatics (ICSAI2012)*, pages 1975–1980, 2012.

[2] T. Cover and P. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27, 1967.

[3] M. Vijay Anand, B. KiranBala, S. R. Srividhya, Kavitha C., Mohammed Younus, and Md Habibur Rahman. Gaussian naïve bayes algorithm: A reliable technique involved in the assortment of the segregation in cancer. *Mobile Information Systems*, 2022(1):2436946, 2022.

[4] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, Sep 1995.

[5] Charles H. Bennett and Gilles Brassard. Quantum cryptography: Public key distribution and coin tossing. *Theoretical Computer Science*, 560:7–11, December 2014.

[6] W. K. Wootters and W. H. Zurek. A single quantum cannot be cloned. *Nature*, 299:802–803, 1982.

[7] Artur K. Ekert. *Quantum Cryptography and Bell's Theorem*, pages 413–418. Springer US, Boston, MA, 1992.

[8] John Preskill. Quantum Computing in the NISQ era and beyond. *Quantum*, 2:79, August 2018.

[9] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 10th anniversary edition, 2010.

[10] Panos K. Syriopoulos, Nektarios G. Kalampalikis, Sotiris B. Kotsiantis, and Michael N. Vrahatis. knn classification: a review. *Annals of Mathematics and Artificial Intelligence*, Sep 2023.

[11] Rajib Kumar Halder, Mohammed Nasir Uddin, Md. Ashraf Uddin, Sunil Aryal, and Ansam Khraisat. Enhancing k-nearest neighbor algorithm: a comprehensive review and performance analysis of modifications. *Journal of Big Data*, 11(1):113, Aug 2024.

[12] Thomas Hofmann, Bernhard Schölkopf, and Alexander J. Smola. Kernel methods in machine learning. *The Annals of Statistics*, 36(3), June 2008.

[13] Xiaojian Ding, Jian Liu, Fan Yang, and Jie Cao. Random radial basis function kernel-based support vector machine. *Journal of the Franklin Institute*, 358(18):10121–10140, 2021.

[14] Mariette Awad and Rahul Khanna. *Support Vector Machines for Classification*, pages 39–66. Apress, Berkeley, CA, 2015.

[15] P.J.Beslin Pajila, B.Gracelin Sheena, A. Gayathri, J. Aswini, M. Nalini, and Siva Subramanian R. A comprehensive survey on naive bayes algorithm: Advantages, limitations and applications. In *2023 4th International Conference on Smart Electronics and Communication (ICOSEC)*, pages 1228–1234, 2023.

[16] Indika Wickramasinghe and Harsha Kalutarage. Naive bayes: applications, variations and vulnerabilities: a review of literature with code snippets for implementation. *Soft Computing*, 25(3):2277–2293, Feb 2021.

[17] Karl Pearson. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901.

[18] Ali Javadi-Abhari, Matthew Treinish, Kevin Krsulich, Christopher J. Wood, Jake Lishman, Julien Gacon, Simon Martiel, Paul D. Nation, Lev S. Bishop, Andrew W. Cross, Blake R. Johnson, and Jay M. Gambetta. Quantum computing with qiskit, 2024.

[19] V. A. Marčenko and L. A. Pastur. Distribution of eigenvalues for some sets of random matrices. *Mathematics of the USSR-Sbornik*, 1(4):457, apr 1967.

[20] Peter I. Frazier. A tutorial on bayesian optimization, 2018.