

*C02047: Doctor of Philosophy
33875 PhD Thesis: Computer Systems
August 2024*

Application of Quantum Machine Learning for Optimization and Data Imbalance Solutions

*A thesis submitted in fulfilment of the requirements
for the degree of*

Doctor of Philosophy
in
Quantum Machine Learning

by
Nishikanta Mohanty

to
Center For Quantum Software and Information
Faculty of Engineering and Information Technology
University of Technology Sydney
NSW - 2007, Australia

August 2024

CERTIFICATE OF ORIGINAL AUTHORSHIP

I, *Nishikanta Mohanty*, declare that this thesis is submitted in fulfilment of the requirements for the award of *Doctor of Philosophy*, in the *Faculty of Engineering and IT* at the University of Technology Sydney.

This thesis is wholly my own work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

This document has not been submitted for qualifications at any other academic institution.

This research is supported by the Australian Government Research Training Program.

Production Note:

SIGNATURE: Signature removed prior to publication.

[Nishikanta Mohanty]

DATE: 10th August, 2024

PLACE: Sydney, Australia

© 2024 by Nishikanta Mohanty
All Rights Reserved

ABSTRACT

This thesis investigates the inventive application of quantum computing methods to solve two critical challenges: optimization challenges with a focus on the Vehicle Routing Problem (VRP) and the issue of data imbalance, especially via the Synthetic Minority Oversampling Technique (SMOTE). The first section of research delves into the VRP, a critical optimization challenge, and emphasizes the use of Quantum algorithms to solve the vehicle routing problem using hybrid quantum algorithms, as well as the effects of quantum noise. The research then extends to solve the VRP using a Quantum Machine Learning algorithm, the Quantum Support Vector Machine. The second portion of the research uses quantum approaches to address the class imbalance by reinterpreting the synthetic minority oversampling strategy in quantum. Using these cutting-edge research efforts, the thesis highlights quantum computing's transformational potential for tackling complicated optimization issues and resolving data imbalance. It emphasizes the link between quantum physics and computer science, setting the path for future advances in quantum algorithms and their applications in various domains.

DEDICATION

I dedicate this thesis to:

- **My-Supervisor & Co-Supervisor:** [Dr. Christopher Ferrie, Dr. Simon Devitt] - for their guidance, mentorship, and invaluable insights. Your expertise has shaped this work and my growth as a researcher.
- **My Family:** For their unwavering support, encouragement, and sacrifices throughout my academic journey. Your love and belief in me have been my greatest motivation.
- **My Co-Author and Mentors:** I am very grateful to my co-author [Mr. Bikash Behera] and mentors [Dr.Moloy De, Pravat Dash] whose guidance, insight, and support have been instrumental in my research and publications.
- **Academic Institution :** I am very thankful to CQSI (*Center for Quantum Software and Information*) and SQA (*Sydney Quantum Academy*) For their inspiration, assistance institutional support in courses, tools, and networking opportunities.

This thesis is a testament to the collective support and encouragement I have received, and I am deeply grateful to each and every person who has been a part of this journey.

Nishikanta Mohanty
August 2024

ACKNOWLEDGMENTS

I Nishikanta Mohanty express my sincere gratitude to everyone who helped me navigate the complex paths of Quantum Machine Learning for the preparation of my thesis and during the tenure of my Study. I have been extremely fortunate to gather guidance, support, and encouragement from some extremely dynamic individuals and insightful sources. First and foremost, I would like to express sincere gratitude to my supervisor, Dr.Christopher Ferrie, whose knowledge and compassionate advice have been invaluable to my academic path at UTS. His unwavering support and wise counsel have greatly influenced the direction and focus of my research.

I am equally thankful to Dr. Simon Devitt, my co-supervisor, whose wisdom and suggestions have improved my comprehension of and methodology in this challenging area. His guidance and feedback to review of my papers and publication has been instrumental.

I am very grateful to Mr.Bikash Behera for his collaborative efforts and co-authorship. His creative ideas and constant engagement in navigating the research work have substantially enhanced the breadth and quality of our collaborative research endeavours.

I owe a significant debt of gratitude to the UTS academic and research community, whose stimulating discussions and constructive criticism have been instrumental in shaping my research. This environment has truly fostered an atmosphere conducive to exploring the vast potential of quantum machine learning.

Thank you, especially to Sydney Quantum Academy, for its inspiration and assistance. The academy's courses, tools, and networking opportunities have been invaluable in expanding my horizons and establishing connections with prominent leaders in the quantum area.

Finally, I would like to thank my friends, family, and classmates whose unwavering support and faith in my skills have given me bravery and strength. Their constant support has been my source of strength throughout this journey.

LIST OF PUBLICATIONS

Chapter: 3

Name: Analysis of The Vehicle Routing Problem Solved via Hybrid Quantum Algorithms in Presence of Noisy Channels

Authors: Nishikanta Mohanty, Bikash K. Behera, Christopher Ferrie

Contributions:

- NM Conceptualised Quantum Noise for VRP
- BKB Conceptualised of VRP formulation in Quantum.
- NM, BKB and CF Designed the custom Hamiltonian for VRP
- NM ran the simulation of VRP with quantum noise and recorded the results.
- CF reviewed the simulation data and suggested stats and figures.
- NM, BKB, and CF equally contributed to the preparation and review of the manuscript.

Status: Published in *IEEE Transactions On Quantum Engineering*

Address: <https://doi.org/10.1109/TQE.2023.3303989>

Signatures:

- Nishikanta Mohanty
- Bikash K. Behera
- Christopher Ferrie

Chapter: 4

Name: Solving The Vehicle Routing Problem via Quantum Support Vector Machines

Authors: Nishikanta Mohanty, Bikash K. Behera, Christopher Ferrie

Contributions:

- NM and BKB Conceptualised Quantum Support Vector Machine for solving VRP
- NM, BKB, and CF Designed the custom Hamiltonian for VRP
- NM ran the simulation of VRP with QSVM.
- NM, BKB, and CF contributed to the selection of quantum encoding schemes
- NM, BKB contributed to circuit building of encoding, VRP, and VQE.
- CF reviewed the simulation data and suggested stats and figures.
- NM, BKB, and CF equally contributed to the preparation and review of the manuscript.

Status: Published in *Quantum Machine Intelligence*

Address: <https://doi.org/10.1007/s42484-024-00161-4>

Signatures:

- Nishikanta Mohanty
- Bikash K. Behera
- Christopher Ferrie

Chapter: 5

Name: A Quantum Approach to Synthetic Minority Oversampling Technique (SMOTE)

Authors: Nishikanta Mohanty, Bikash K. Behera, Christopher Ferrie, Pravat Dash

Contributions:

- NM conceptualized the idea of Quantum-SMOTE
- NM, BKB conceptualized the idea of quantum synthetic data by vector rotation
- NM, BKB, and CF conceptualized the idea of replacing KNN with clustering
- BKB conceptualized the idea of a compact swap test to reduce qubit requirement.
- NM and BKB formulated the hyper-parameters of Quantum SMOTE
- NM and BKB ran the simulation of the Algorithm on a real-world dataset.
- NM and PD evaluated the testing process of and use of classical algorithms and statistics
- CF oversaw the entire process of formulation and testing of the algorithm.
- NM, BKB, and CF equally contributed to the preparation and review of the manuscript.

Status: Under Review in *Quantum Machine Intelligence*

Address: <https://doi.org/10.48550/arXiv.2402.17398>

Signatures:

- Nishikanta Mohanty
- Bikash K. Behera
- Christopher Ferrie

TABLE OF CONTENTS

List of Publications	ix
List of Figures	xvii
List of Tables	xxiii
1 Introduction	1
1.1 Quantum Computing In NISQ Era	1
1.1.1 Content and Objectives :	4
1.1.2 Reserach Methods and Tools	6
I Part I	11
2 Foundational Methods and Algorithms	13
2.0.1 Quantum Gates	15
2.0.2 Quantum Approximate Optimization Algorithm (QAOA) . .	18
2.0.3 Variational Quantum Eigensolver (VQE)	28
2.0.4 Quantum Kernel Methods	31
2.0.5 Quantum Feature Maps	32
II Part II	41
3 Analysis of The Vehicle Routing Problem Solved via Hybrid Quantum Algorithms in the Presence of Noisy Channels	43
3.1 Introduction	44

TABLE OF CONTENTS

3.2	Mathematical Background	45
3.2.1	Combinatorial Optimization	46
3.2.2	Adiabatic Quantum Computation	46
3.2.3	QAOA	48
3.2.4	Ising Model	49
3.2.5	VQE	49
3.3	Modelling VRP in Quantum	50
3.4	Analysis And Circuit Building	54
3.5	VQE Simulation of VRP	57
3.6	Noise Model Simulation of VQE	61
3.6.1	Amplitude Damping	61
3.6.2	Bit-Flip Noise	62
3.6.3	Phase Flip Noise	62
3.6.4	Bit-Phase Flip Noise	62
3.6.5	Depolarizing Noise	63
3.7	Inferences from Simulation	70
3.7.1	Amplitude damping Noise	71
3.7.2	Bit-Flip Noise	71
3.7.3	Bit-Phase-Flip Noise	72
3.7.4	Phase-Flip and Depolarizing Noise Channel	72
3.7.5	Data gathering and Statistics Collection	72
3.7.6	Choice Of COBYLA	73
3.8	Discussion and Conclusion	74
3.9	Supplementary Tables and Statistics	75
3.9.1	No Noise VQE Simulation Stats	75
3.9.2	Amplitude Damping Tables	76
3.9.3	Bit-Flip Tables	77
3.9.4	Phase-Flip Tables	78
3.9.5	Bit-Phase-Flip Table	79
3.9.6	Depolarising Table	79
4	Solving The Vehicle Routing Problem via Quantum Support Vector Machines	81

4.1	Introduction	82
4.1.1	Quantum Computing	82
4.1.2	Vehicle Routing Problem	82
4.1.3	Quantum Support Vector Machine (QSVM)	83
4.1.4	Novelty and Contribution	84
4.1.5	Organization	84
4.2	Background	85
4.2.1	QAOA	85
4.2.2	Ising Model	86
4.2.3	Quantum Support Vector Machine	87
4.2.4	Amplitude Encoding(AE)	88
4.2.5	Angle Encoding (AgE)	89
4.2.6	Higher Order Encoding(HO)	89
4.2.7	IQP Encoding(IqpE)	90
4.2.8	VQE	90
4.3	Methodology	91
4.3.1	Modelling VRP in QSVM	91
4.3.2	Analysis And Circuit Building	95
4.4	Results	101
4.4.1	VQE Simulation of QSVM and VRP	101
4.4.2	Inferences from Simulation	110
4.4.3	Complexity and Cost Considerations	112
4.4.4	Experimental Setup, data gathering, and statistics	113
4.5	Conclusion	114
5	A Quantum Approach to Synthetic Minority Oversampling Technique (SMOTE)	117
5.1	Introduction	118
5.1.1	Unbalanced Classification	118
5.1.2	Overview of SMOTE	118
5.1.3	Existing works on SMOTE	118
5.1.4	Purpose and Scope	119
5.1.5	Organization	119

TABLE OF CONTENTS

5.2	Background	120
5.2.1	Basic Concept of SMOTE	120
5.2.2	How SMOTE Works	120
5.2.3	Variants of SMOTE	123
5.2.4	K-Means Clustering	124
5.2.5	ROC Curve	125
5.3	Emulating SMOTE Using Quantum	126
5.3.1	Swap Test	127
5.3.2	Applying Rotation to data point	133
5.3.3	Quantum SMOTE Algorithm	138
5.4	Case Study and Results	144
5.4.1	Improving Telecom Churn Prediction Using SMOTE	144
5.4.2	Inferences from Simulation	169
5.5	Conclusion	170
 III Part III		173
 6 Conclusion		175
6.1	Research Findings	175
6.1.1	Chapter 3	175
6.1.2	Chapter 4	176
6.1.3	Chapter 5	178
6.2	Concluding Remarks	179
 A Appendix		181
 Bibliography		183

LIST OF FIGURES

FIGURE	Page
2.1 Sample circuit for Rotation Encoding	33
2.2 (a) Linearly separable data. (b) Classification with SVM.	35
2.3 Support Vector Machine	36
2.4 Circular Data	38
2.5 Separating Circular data By SVM	39
3.1 (a) Sample circuit showing gate selections for H_{cost} . (b) Sample circuit showing gate selections with additional U gate after barrier for H_{mixer} . Note: The sample circuits displayed in the figures represent the building blocks of the actual circuit and do not represent actual angles that are obtained as a solution of VRP using VQE.	54
3.2 VRP Simulation circuit for 6 Qubits	58
3.3 Plot illustrating the circuit simulation of VRP with 5 layers using various optimizers (COBYLA, L_BFGS_B, SLSQP, SPSA). The plot consists of two separate graphs depicting the simulation output of 6 qubit and 12 qubit circuits each executed with 1024 shots, respectively. Each plot, in turn, consists of four lines indicating energy values for different optimizers. The average value at each Layer is represented in pairs with (layer, average energy) format.	59

3.4	Plot illustrating the circuit simulation of VRP with 5 layers using various optimizers (COBYLA, L_BFGS_B, SLSQP, SPSA). The plot consists of two graphs depicting the simulation output of 6 qubit and 12 qubit circuits each executed with 1024 shots, respectively. Each plot consists of four lines indicating energy values for different optimizers. The minimum value at each Layer is represented in pairs with (layer, minimum energy) format.	60
3.5	Plot illustrating the average Deviation of the energy cost of VRP with 4 layers using various Amplitude damping and Bit-flip noise models. The plot consists of two charts depicting the simulation output of 6 qubit and 12 qubit circuits, respectively. The Average Deviation at each Layer is represented in pairs with (Noise parameter, Average Deviation of Energy cost) format.	64
3.6	Plot illustrating the average Deviation of the energy cost of VRP with 4 layers using Bit-phase-flip, and Depolarising noise. The plot consists of two charts depicting the simulation output of 6 qubit and 12 qubit circuits, respectively. The Average Deviation at each Layer is represented in pairs with (Noise parameter, Average Deviation of Energy cost) format.	65
3.7	Plot illustrating the average energy cost of VRP with 4 layers using the Phase-flip noise model. The plot consists of two charts depicting the simulation output of 6 qubit and 12 qubit circuits, respectively. The average deviation at each Layer is represented in pairs with (Noise parameter, Average Deviation of Energy cost) format.	66
4.1	Schematic diagram depicting quantum circuit for Kernel estimation.	88
4.2	(a) Circuit example illustrating gate operations for H_c . (b) Circuit example displaying gate selections with an additional u gate for H_{mxr}	97
4.3	Plot illustrating different encoding methods for two qubits. (a) Amplitude encoding, (b) angle encoding, (c) Higher order encoding, (d) IQP encoding.	100

4.4	Plot illustrating Amplitude encoding results for QSVM solution of VRP. (a) Amplitude encoding 6 qubits Fix Hamiltonian, (b) Amplitude encoding 6 qubits Variable hamiltonian	102
4.5	Plot illustrating angle encoding results for QSVM solution of VRP. (a) Angle encoding 6 qubits Fix hamiltonian, (b) Angle encoding 12 qubits Fix hamiltonian, (c) Angle encoding 6 qubits Variable hamiltonian, (d) Angle encoding 12 qubits Variable hamiltonian.	104
4.6	Plot illustrating Higherorder encoding results for QSVM solution of VRP. (a) Higherorder encoding 6 qubits Fix hamiltonian, (b) Higherorder encoding 12 qubits Fix hamiltonian, (c) Higherorder encoding 6 qubits Variable hamiltonian, (d) Higherorder encoding 12 qubits Variable hamiltonian.	106
4.7	Plot illustrating IQP encoding results for QSVM solution of VRP. (a) IQP encoding 6 qubits Fix hamiltonian, (b) IQP encoding 12 qubits Fix hamiltonian, (c) IQP encoding 6 qubits Variable hamiltonian, (d) IQP encoding 12 qubits Variable hamiltonian.	108
5.1	Plot illustrating different SMOTE mechanisms. (a) Classical SMOTE, (b) Quantum SMOTE.	127
5.2	Swap test circuit.	128
5.3	Compact Swap test circuit.	133
5.4	Plot representing Sample data points of Minority class from population without any rotation	135
5.5	Plot illustrating impact of X Rotation on Sample data points of Minority class. (a) X Rotation with <i>split_factor</i> = 2, (b) X Rotation with <i>split_factor</i> = 5, (c) X Rotation with <i>split_factor</i> = 10.	136
5.6	Plot illustrating impact of Y Rotation on Sample data points of Minority class. (a) Y Rotation with <i>split_factor</i> = 5, (b) Y Rotation with <i>split_factor</i> = 100	137
5.7	Plot illustrating impact of Z Rotation on Sample data points of Minority class. (a) Z Rotation with <i>split_factor</i> = 5, (b) Z Rotation with <i>split_factor</i> = 10	138
5.8	Data point rotation circuit.	147

LIST OF FIGURES

5.9 Plot illustrating impact of synthetic data generation on Sample data points of Minority class. (a) data points with no synthetic, (b) 30% synthetic, (c) 40% synthetic, (d) 50% synthetic. 149

5.10 Plot illustrating distribution of 3 columns: (a) Tenure, (b) MonthlyCharges, and (c) TotalCharges. 150

5.11 Plot illustrating distribution of 3 columns with induction of synthetic datapoints with overall 30% minority : (a) Tenure, (b) MonthlyCharges, and (c) TotalCharges. 151

5.12 Plot illustrating distribution of 3 columns with induction of synthetic data points with overall 40% minority: (a) Tenure, (b) MonthlyCharges, and (c) TotalCharges. 152

5.13 Plot illustrating distribution of 3 columns with induction of synthetic data points with overall 50% minority : (a) Tenure, (b) MonthlyCharges, and (c) TotalCharges. 153

5.14 Plot illustrating Model Charts for random forest model with out SMOTE. (a) Confusion Matrix Random Forest Model, (b) Normalized Confusion Matrix Random Forest Model, (c) AUC-ROC Random Foerest Model, (d) Precision Recall Curve Random Forest Model. 155

5.15 Plot illustrating Model Charts for logistic regression model without SMOTE. (a) Confusion Matrix Logistic Regression Model, (b) Normalized Confusion Matrix Logistic Regression Model, (c) AUC-ROC Logistic Regression Model, (d) Precision Recall Curve Logistic Regression Model. 156

5.16 Plot illustrating Confusion Matrix for random forest model with and without smote for comparison. (a) Confusion Matrix Random Forest Model without smote, (b) Confusion Matrix Random Forest Model with smote and 30% Minority, (c) Confusion Matrix Random Forest Model with smote and 40% Minority, (d) Confusion Matrix Random Forest Model with smote and 50% Minority. 157

5.17 Plot illustrating Normalized Confusion Matrix for random forest model with and without smote for comparison. (a) Normalized Confusion Matrix Random Forest Model without smote, (b) Normalized Confusion Matrix Random Forest Model with smote and 30% Minority, (c) Normalized Confusion Matrix Random Forest Model with smote and 40% Minority, (d) Normalized Confusion Matrix Random Forest Model with smote and 50% Minority. 158

5.18 Plot illustrating Area Under Receiver Operating Characteristic Curve (AUC-ROC) for random forest model with and without smote for comparison. (a) AUC-ROC Random Forest Model without smote, (b) AUC-ROC Random Forest Model with smote and 30% Minority, (c) AUC-ROC Random Forest Model with smote and 40% Minority, (d) AUC-ROC Random Forest Model with smote and 50% Minority. 159

5.19 Plot illustrating Precision-Recall Curve (AUC) for random forest model with and without smote for comparison. (a) Precision-Recall Curve (AUC) Random Forest Model without smote, (b) Precision-Recall Curve (AUC) Random Forest Model with smote and 30% Minority, (c) Precision-Recall Curve (AUC) Random Forest Model with smote and 40% Minority, (d) Precision-Recall Curve (AUC) Random Forest Model with smote and 50% Minority. 160

5.20 Plot illustrating Confusion Matrix for Logistic Regression model with and without smote for comparison. (a) Confusion Matrix Logistic Regression Model without smote, (b) Confusion Matrix Logistic Regression Model with smote and 30% Minority, (c) Confusion Matrix Logistic Regression Model with smote and 40% Minority, (d) Confusion Matrix Logistic Regression Model with smote and 50% Minority. 161

5.21 Plot illustrating Normalized Confusion Matrix for Logistic Regression model with and without smote for comparison. (a) Normalized Confusion Matrix Logistic Regression Model without smote, (b) Normalized Confusion Matrix Logistic Regression Model with smote and 30% Minority, (c) Normalized Confusion Matrix Logistic Regression Model with smote and 40% Minority, (d) Normalized Confusion Matrix Logistic Regression Model with smote and 50% Minority. 162

5.22 Plot illustrating Area Under Receiver Operating Characteristic Curve (AUC-ROC) for Logistic Regression model with and without smote for comparison. (a) AUC-ROC Logistic Regression Model without smote, (b) AUC-ROC Logistic Regression Model with smote and 30% Minority, (c) AUC-ROC Logistic Regression Model with smote and 40% Minority, (d) AUC-ROC Logistic Regression Model with smote and 50% Minority. . . 163

5.23 Plot illustrating Precision-Recall Curve (AUC) for Logistic Regression model with and without smote for comparison. (a) Precision-Recall Curve (AUC) Logistic Regression Model without smote, (b) Precision-Recall Curve (AUC) Logistic Regression Model with smote and 30% Minority, (c) Precision-Recall Curve (AUC) Logistic Regression Model with smote and 40% Minority, (d) Precision-Recall Curve (AUC) Logistic Regression Model with smote and 50% Minority. 164

LIST OF TABLES

TABLE	Page
3.1 VQE simulation of amplitude damping, bit-flip, phase-flip, bit-phase-flip, and depolarizing channel for 6 qubits with 4 layers involving optimizer COBYLA, where energy costs are averaged over 10 simulations.	67
3.2 VQE simulation of amplitude damping, bit-flip, phase-flip, bit-phase-flip, and depolarizing channel for 12 qubits with 4 layers involving optimizer COBYLA, where energy costs are averaged over 10 simulations.	68
3.3 For 6 qubits with 4 layers and using COBYLA as an optimizer, the table above shows the average deviation from the classical minimum (over 10 simulations) for VQE simulations utilizing amplitude damping, bit-flip, phase-flip, bit-phase-flip, and depolarising channels.	69
3.4 For 12 qubits with 4 layers and using COBYLA as an optimizer, the table above shows the average deviation from the classical minimum (over 10 simulations) for VQE simulations utilizing amplitude damping, bit-flip, phase-flip, bit-phase-flip, and depolarising channels.	70
3.5 Table summarizing the inferences on deviation from the classical minimum for VQE simulation VRP using amplitude damping, bit-flip, phase-flip, bit-phase-flip, and depolarizing channels for 6 and 12 qubits.	73
3.6 Table summarizing the Average Energy Cost (out of 15 runs) of VQE simulation using 5 Optimizers.	76
3.7 Table containing the Minimum Energy value (out of 15 runs) across each layer of VQE simulation using 5 Optimizers.	76
3.8 Table containing the Average Energy values (out of 10 runs) for Amplitude Damping Noise Channel across each layer of VQE simulation Using COBYLA.	77

LIST OF TABLES

3.9 Table containing the Minimum Energy values (out of 10 runs) for Amplitude Damping Noise Channel across each layer of VQE simulation Using COBYLA. 77

3.10 Table containing the Average Energy values (out of 10 runs) for BitFlip Noise Channel across each layer of VQE simulation Using COBYLA. . 77

3.11 Table containing the Minimum Energy values (out of 10 runs) for BitFlip Noise Channel across each layer of VQE simulation Using COBYLA. . 78

3.12 Table containing the Average Energy values (out of 10 runs) for Phase-Flip Noise Channel across each layer of VQE simulation Using COBYLA. 78

3.13 Table containing the Minimum Energy values (out of 10 runs) for Phase-Flip Noise Channel across each layer of VQE simulation Using COBYLA. 78

3.14 Table containing the Average Energy values (out of 10 runs) for Bit-Phase-Flip Noise Channel across each layer of VQE simulation Using COBYLA. 79

3.15 Table containing the Minimum Energy values (out of 10 runs) for Bit-PhaseFlip Noise Channel across each layer of VQE simulation Using COBYLA. 79

3.16 Table containing the Average Energy values (out of 10 runs) for Depolarising Noise Channel across each layer of VQE simulation Using COBYLA. 80

3.17 Table containing the Minimum Energy values (out of 10 runs) for Depolarising Noise Channel across each layer of VQE simulation Using COBYLA. 80

4.1 For 6 and 12 qubit VRP circuits using SVM with 2 layers, the table above shows the Accuracy and Error with reference to classical minimum (over 50 iterations) for VQE simulations over a fixed Hamiltonian; utilizing Amplitude, Angle, Higher-Order, and IQP encoding schemes, Over the use of COBYLA, SLSQP and L_BGFS_B optimizers. 109

4.2	For 6 and 12 qubit VRP circuits using SVM on with 2 layers, the table above shows the Accuracy and Error with reference to classical minimum (over 50 iterations) for VQE simulations on variable hamiltonians utilizing Amplitude, Angle, Higher-Order, and IQP encoding schemes, Over the use of COBYLA, SLSQP and L_BGFS_B optimizers.	110
4.3	Table consisting of Quantum Depth and Quantum Cost of Various Encoding Schemes compared with Standard VQE implementation of VRP.	114
5.1	Table comparing Accuracy, F1, and AUC score of Random Forest Model for telecom churn dataset without SMOTE, and post SMOTE with minority % as 30%, 40%, and 50%.	168
5.2	Table comparing Confusion matrix of Random Forest and Logistic regression without SMOTE, and post SMOTE with minority% as 30%, 40%, and 50%.	169

INTRODUCTION

1.1 Quantum Computing In NISQ Era

The emergence of quantum computing has introduced a new computational era, that has the potential to significantly transform several computational domains such as machine learning, optimization, chemistry, cryptography, materials science, and other related subjects. Quantum computing, in contrast to classical computing, utilizes qubits to process information in a multi-dimensional computational space, allowing for the parallel processing of information on an unprecedented scale. This approach leverages the principles of quantum mechanics, as opposed to relying on binary bits used by current computers. [1].

Preskill (2018) has presented the idea of Noisy Intermediate-Scale Quantum (NISQ) computing technology, which has generated significant interest and advancement in recent years. At present, NISQ devices, which have 50,Äì100 qubits, can perform computing tasks that go beyond classical capabilities. However, they are limited by factors related to noise and operational integrity. These devices, though lacking significant error correction capabilities, represent a crucial advancement towards the realization of the quantum computing era [2].

Navigating the NISQ environment requires meeting a unique combination of

challenges and opportunities despite its computational advantages over classical computing. The presence of intrinsic noise presents substantial challenges to the reliability and accuracy of algorithms in NISQ devices. However, this period also presents opportunities for innovative algorithms that are specially tailored to make use of the unique features of NISQ machines, while simultaneously evaluating the balance between sensitivity to noise and quantum advantage [3].

Throughout recent years, multiple algorithms have been proposed to tackle a variety of problems with use of quantum computing that can take advantage of the existing NISQ devices. Some of these algorithms are worth mentioning which are regarded as the foundation or fundamental principle on which most of quantum research and applications are created. Following points touch upon the overview of these algorithms

- **Variational Quantum Eigensolver (VQE):** The Variational Quantum Eigensolver (VQE) is a very promising NISQ algorithm that showcases the potential for quantum advantage in the field of quantum chemistry. The eigenvalue issue for molecular Hamiltonians is addressed by VQE via the use of hybrid quantum-classical approaches. This approach provides valuable insights into molecular characteristics, which have significant implications for drug development and materials research [4].
- **Quantum Approximate Optimization Algorithm (QAOA):** The Quantum Approximate Optimization Algorithm (QAOA), focuses on addressing optimization problems. The approach known as QAOA was presented by [5] as a means to tackle combinatorial optimization. The authors showcased the algorithm's capability to approximate solutions for issues that are considered difficult for classical computers.
- **Grover's Algorithm :** Grover's algorithm, renowned for its quadratic acceleration in search problems, encounters difficulties in the NISQ era owing to noise and coherence time limitations. However, its principles still serve as a source of inspiration for algorithmic advances that are compatible with NISQ. The purpose of these changes is to use some aspects of Grover's acceleration in the presence of noise [6].

- **Shor’s algorithm** : Shor’s algorithm is a quantum computing technique created by Peter Shor in 1994 [7]. It is meant to effectively factor big numbers into their prime factors, which is computationally difficult for traditional computers. Its relevance arises from its ability to break commonly used cryptographic methods like as RSA, which depend on the difficulty of factorization. The programme uses quantum features like superposition and entanglement to execute operations tenfold quicker than the most well-known conventional algorithms. The Quantum Fourier Transform is fundamental to Shor’s method, since it assists in determining the periodicity of functions, an essential step in factorization. While promising, the actual implementation of Shor’s algorithm requires the creation of sufficiently big and error-corrected quantum computers.
- **Quantum Support Vector Machine (QSVM)** : In the era of NISQ, quantum machine learning is emerging as a promising field. Algorithms like as Quantum Support Vector Machines (QSVMs) are showing potential benefits in data processing and analytical tasks. In their study, Havlicek et al. (2019) showed the potential of quantum kernel techniques to revolutionise machine learning by surpassing conventional approaches in certain situations.

The pursuit of achieving quantum advantage, which refers to the stage at which quantum computers surpass their classical counterparts in significant tasks, continues to be the overarching objective. The realization of complete quantum advantage is dependent upon the development of fault-tolerant quantum computing. However, NISQ algorithms provide a window of opportunity to attain this significant milestone within certain domains [8]. This intriguing aspect of quantum algorithms on NISQ devices motivates this research. The primary objective of this thesis is to establish an association between two distinct groups: machine learning engineers who are interested in making contributions to the advancement of quantum technology and quantum researchers who are committed to developing applications using quantum computers. In this study, our goal is to bring together these separate groups exploring the key aspects of machine learning in quantum systems. Furthermore, the thesis presents a number of novel methodologies for

harnessing machine learning via the use of quantum concepts in order to tackle complex industrial scenarios.

1.1.1 Content and Objectives :

The primary objective of this thesis is to investigate the emerging convergence of quantum computing and machine learning. The aim is to provide insight into the benefits of integrating concepts derived from quantum mechanics into computational models, with a specific focus on optimization and data preprocessing for machine learning. This study aims to shed light on the potential impact of quantum computing on conventional machine-learning approaches by conducting a thorough analysis of three seminal articles. By doing so, it seeks to enhance understanding and accelerate the progress of both fields. The selected examples not only represent the advanced methods in developing quantum algorithms but also demonstrate the wide range of applications and issues that these innovative techniques may address.

The first publication, titled "An Examination of the Vehicle Routing Problem Resolved through Hybrid Quantum Algorithms in the Presence of Noisy Channels," plays an essential part in this thesis by showcasing the pragmatic use of hybrid quantum algorithms for addressing intricate optimization challenges along with quantifying the impact of various types of noise on these solutions. This work has significant relevance for professionals in the fields of machine learning engineering and quantum algorithm development. It effectively demonstrates the potential of quantum-enhanced computations in addressing the constraints of classical computing, especially in conditions of noisy intermediate-scale quantum (NISQ) settings. Quantum computing is well-suited to address real-world difficulties such as the vehicle routing problem (VRP), which is a crucial challenge in logistics and supply chain management. The thesis seeks to analyze this research in order to emphasize the methodology and results of using quantum algorithms in real-world situations, with a particular focus on the significance of hybrid approaches in the present quantum computing environment.

The second study, titled "Solving The Vehicle Routing Problem via Quantum Support Vector Machines," explores the use of quantum computing for machine

learning problems, moving beyond quantum-enhanced optimization to the field of quantum machine learning. This study signifies a notable advancement in the use of quantum algorithms to machine learning models, with a particular focus on Quantum Support Vector Machines (QSVMs). The aim of this study is to examine and comprehend the underlying mechanism by which QSVMs are capable of processing and analyzing data of a level of complexity and size that surpasses the capabilities of classical machines. The paper also proposes to use various quantum encoding schemes along with a cost function to construct a QSVM that can be used to solve challenging Optimization problems. This study not only expands the range of applications in quantum computing but also acquaints machine learning professionals with the innovative opportunities and difficulties associated with quantum-enhanced algorithms. This study aims to provide a scholarly contribution to the ongoing conversation around the interdependent connection between quantum computing and machine learning. It emphasizes the significant impact that quantum algorithms may have on data-driven applications, highlighting their revolutionary capabilities.

The third work, titled "A Quantum Approach to Synthetic Minority Oversampling Technique (SMOTE)," tackles a significant obstacle in the domain of machine learning, namely, the issue of data imbalance. The thesis showcases a fascinating case study on the use of quantum computing to enhance the SMOTE approach. This application demonstrates how quantum algorithms may enhance the preparation stage of machine learning processes. This work investigates quantum-enhanced data augmentation strategies to elucidate the wider ramifications of quantum computing in enhancing the equity and precision of machine learning models. The thesis aims to showcase the many contributions of quantum computing to machine learning, including improvements in algorithmic efficiency and data pre-treatment.

The overall aim of this thesis is to find use cases where quantum machine learning and the current scope of computing can be integrated into the existing computational landscape. The thesis also aims to accomplish several objectives by conducting a thorough analysis of these three papers. These objectives include providing a clear understanding of the present condition and future potential of quantum-enhanced machine learning, demonstrating the practical uses and difficulties of implementing quantum algorithms in real-life situations and pro-

moting collaboration between machine learning engineers and quantum scientists. By accomplishing each of these objectives, the thesis strives to make significant contributions to the ongoing advancement of quantum computing and its integration with machine learning, thereby facilitating upcoming breakthroughs and applications that harness the capabilities of quantum technologies.

Finally, this thesis aims to consolidate existing research and advancements in the intersection of quantum computing and machine learning. This study seeks to expand our comprehension and utilization of quantum-enhanced computing methods by conducting a thorough analysis of relevant literature and generating fresh perspectives. By doing so, it attempts to pave the way for future technological advancements in these domains.

1.1.2 Reserach Methods and Tools

1.1.2.1 Statistical Analysis and Visualization

The utilization of statistical analysis, advanced visualizations, and a comprehensive comprehension of data distributions is crucial in the examination of hybrid quantum algorithm performance in real-world scenarios, as demonstrated in the paper titled "Analysis of The Vehicle Routeing Problem Solved via Hybrid Quantum Algorithms in Presence of Noisy Channels." The research methodology used in this study entails a methodical analysis of algorithmic results in comparison to different noise models. Statistical techniques are utilized to evaluate the robustness and effectiveness of quantum solutions in addressing the Vehicle Routeing Problem (VRP). Visualization methods, such as line graphs and box plots, play a crucial role in effectively displaying the comparative performance metrics of hybrid quantum algorithms under varying noise intensities. These approaches enable the identification of patterns and variations that may not be readily observable just via numerical analysis. An illustration of the algorithm's accuracy or solution quality degradation with increasing noise level may be successfully shown using line graphs. Additionally, box plots can provide valuable insights into the variability of the algorithm's performance across numerous runs or configurations.

The 2nd Paper, "Solving The Vehicle Routing Problem via Quantum Support Vector Machines," used statistical analysis and visualizations as a vital study

approach to thoroughly assess the results of this quantum support vector machine across different encoding schemes. This methodology enables a comprehensive analysis of the performance metrics of Q-SVMs, hence permitting a comparative evaluation of various quantum data encoding schemes and their influence on effectively solving the VRP. The assessment of the Q-SVM model's accuracy, computing time, and scalability under various encoding methods relies heavily on the use of descriptive and inferential statistical techniques.

1.1.2.2 Model Evaluation Methods

The use of a comprehensive set of model assessment techniques becomes crucial in order to thoroughly evaluate the effectiveness of the quantum-enhanced methodology presented in the third article titled "A Quantum Approach to Synthetic Minority Oversampling Technique (SMOTE)." The methodologies used in a quantum machine learning thesis, including as the Receiver Operating Characteristic (ROC) curve, Area Under the Curve (AUC), Confusion Matrix, F1 Score, and accuracy, are fundamental components of a thorough study process.

The receiver operating characteristic (ROC) curve is a visual representation that enhances the diagnostic capability of a binary classifier system. It serves as a crucial instrument for assessing the effectiveness of the quantum-enhanced SMOTE algorithm. The ROC curve offers insights into the trade-offs between sensitivity and specificity by graphing the true positive rate versus the false positive rate at different threshold values. In addition, the AUC, which measures the model's ability to differentiate across classes, provides a single numerical result that summarises the model's overall performance overall potential classification thresholds. A greater area under the curve (AUC) signifies superior model performance, particularly when dealing with unbalanced datasets.

The Confusion Matrix is an important method for understanding the model's performance in classifying minority and majority classes. It offers a comprehensive description of the model's predictions, including true positives, false positives, true negatives, and false negatives. The detailed analysis provided enables the calculation of accurate performance measures, including precision, recall, and the F1 Score (the harmonic mean of accuracy and recall). The F1 Score is a

valuable metric in the context of unbalanced datasets due to its ability to strike a balance between precision and recall, hence providing a broader evaluation of model performance that goes beyond traditional accuracy measures.

Accuracy, while a clear and obvious indicator of model performance, may not always provide an accurate picture, particularly when there is an imbalance in class distribution. Hence, it is used in combination with the previously mentioned metrics in order to provide an in-depth review of the efficacy of the quantum-enhanced SMOTE in improving machine learning models.

The combination of these model assessment methodologies provides an extensive foundation for evaluating the effectiveness of the quantum-enhanced SMOTE algorithm in addressing data imbalance, hence offering a multidimensional perspective on the performance of the model. The thesis seeks to carefully apply these evaluation techniques to confirm the efficacy of integrating quantum computing into SMOTE. Additionally, it aims to facilitate further investigation into the potential collaboration between quantum computing and machine learning, especially when addressing inherent challenges like data imbalance.

1.1.2.3 Regression Testing

Regression testing is a critical research methodology used to validate research outcomes and draw statistical inferences from output data in the context of quantum computing applications. This is particularly important in the exploration of the Vehicle Routing Problem (VRP) and the innovative application of Synthetic Minority Oversampling Technique (SMOTE) within quantum paradigms. The research articles titled "Analysis of The Vehicle Routing Problem Solved via Hybrid Quantum Algorithms in Presence of Noisy Channels," "Solving The Vehicle Routing Problem via Quantum Support Vector Machines," and "A Quantum Approach to Synthetic Minority Oversampling Technique (SMOTE)" employ regression testing as a methodology to thoroughly investigate the correlation between different input parameters and the performance metrics of their respective quantum-optimized models.

In the present study, regression testing was utilized to evaluate the resilience and dependability of quantum algorithms across various scenarios. These scenarios

included differences in noise levels for hybrid quantum algorithms, as well as the influence of multiple encoding schemes on the effectiveness of Quantum Support Vector Machines. We were able to validate the effectiveness and stability of their models across a range of scenarios by systematically adjusting these parameters and observing the resulting changes in outcomes, such as solution quality for VRP and accuracy for models used by quantum-enhanced SMOTE.

In addition, the use of regression testing enabled the derivation of critical statistical conclusions from the empirical data, so facilitating the discovery of key factors that influence algorithm performance and the measurement of their effects. The statistical methodology used in this study not only confirmed the robustness of the suggested quantum computing applications but also offered valuable insights into the algorithm parameters to improve overall performance. For example, the examination of the impact of quantum noise on the solution quality of hybrid quantum algorithms for VRP has facilitated the determination of noise tolerance thresholds for quantum circuits and the variability of real outcomes in the absence of noise. In the same way, the assessment of various quantum data encoding schemes by Q-SVMs has provided useful insights into the selection of suitable encoding techniques that may optimize the efficiency of quantum machine learning models.

Regression testing played a crucial role in the research methods of these significant investigations, connecting theoretical developments in quantum computing with their actual implementations. By conducting thorough regression tests, the studies not only confirmed the effectiveness of the proposed quantum algorithms but also enhanced their understanding of the complex relationship between quantum computing parameters and model performance, thus facilitating potential improvements in the field.

Part I

Part I

FOUNDATIONAL METHODS AND ALGORITHMS

The development and application of quantum technologies to complex computational problems have made progress through the use of foundational algorithms developed in quantum computing. Some of these algorithms include the Quantum Approximate Optimization Algorithm (QAOA), the Variational Quantum Eigensolver (VQE), Quantum Kernel Methods (such as Quantum Support Vector Machines (Q-SVMs)), and the use of Rotational Gates. The above-mentioned algorithms not only demonstrate the ability of quantum computing to surpass conventional computational limitations but also establish a base for developing solutions for complex problems. This is quite evident in research that include the Vehicle Routing Problem (VRP) and Synthetic Minority Oversampling Technique (SMOTE) within a quantum framework.

The QAOA algorithm [5] is developed to address combinatorial optimization problems. Its basic approach is to integrate the solution to a particular problem into the ground state of a Hamiltonian. The algorithm uses a sequence of quantum gates to systematically modify parameters and minimize the cost function linked to the problem, with the target of finding an approximate solution that improves with each iteration. The significance of QAOA is clearly demonstrated in the first research paper "Analysis of The Vehicle Routing Problem Solved via Hybrid Quantum Algorithms in Presence of Noisy Channels", especially when confronted

with noisy quantum channels.

VQE [4] uses a combination of quantum and classical methods to solve eigenvalue problems and determine the ground state energy of a system. The quantum state is fine-tuned by using classical optimizers via the adjustment of parameters in a quantum circuit and subsequent measurement of the output. The use of this approach is critical in the formulation and solving of quantum iterations of the VRP, where the objective may be defined as obtaining the energy state with the lowest energy that corresponds to the most optimum path.

Quantum Kernel methods and Q-SVMs [9] expand the scope of classical kernel techniques into the quantum domain, giving the ability to leverage the high-dimensional Hilbert spaces that are accessible to quantum systems for the purpose of machine learning. For example, Q-SVMs demonstrate superior performance in handling complicated datasets with convoluted decision boundaries compared to standard SVMs. Q-SVMs can classify and evaluate routing alternatives in order to solve the VRP, using quantum-enhanced pattern recognition to determine optimal routes.

In quantum circuits, the manipulation of qubits and encoding of information heavily rely on rotational gates, including the Pauli-X, Y, and Z gates, as well as other unitary operations. The exact use of these techniques enables the manipulation and regulation of qubit states, which is crucial for the implementation of quantum circuits that serve as the foundation for algorithms such as QAOA and VQE. Within the realm of quantum-enhanced SMOTE, the use of rotational gates serves to promote the generation of synthetic samples in a superposition manner. This configuration allows for the equitable representation of minority classes within datasets that exhibit imbalances.

Collectively, these fundamental algorithms and quantum operations are the central focus of research endeavors focused on using quantum computing to address real-world challenges. The research conducted on the VRP and quantum-enhanced SMOTE demonstrates the practical implementation of these quantum algorithms in addressing real-world problems, highlighting the wide range of possible effects that quantum computing may have. Incorporating quantum computing concepts into several domains, such as optimizing logistical operations and boosting machine learning models, signifies a prospective scenario in which quantum algorithms

play a substantial role in resolving intricate challenges with heightened efficiency and effectiveness.

2.0.1 Quantum Gates

The core components of quantum circuits are quantum gates, which bear resemblance to the logic gates used in conventional computers. In contrast to conventional gates, quantum gates function on qubits, using the principles of quantum mechanics to execute complex computations capable of simultaneously incorporating numerous states due to the phenomena of superposition and entanglement. Quantum gates are technically described as unitary matrices, and their functionalities may be represented graphically on the Bloch sphere for individual qubits or more elaborate geometric depictions for multi-qubit systems. The integration of these gates into quantum circuits enables the execution of quantum algorithms, such as Shor's algorithm for integer factorization and Grover's algorithm for database searching. This exemplifies the capacity of quantum computing to address issues that are inherently difficult for conventional computers to handle.

Throughout the development of the quantum algorithms and related research for this article, we have extensively used quantum gates to create quantum circuits. Of all the gates used, we are describing some fundamental gates below.

Pauli X Gate: The Pauli-X gate, often known as the NOT gate, is a basic logic gate with a single qubit that executes a bit-flip operation. The matrix representation is as follows:

$$(2.1) \quad X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

The Pauli-X gate is used to convert a qubit in state $|0\rangle$ to $|1\rangle$, and conversely.

Hadamard Gate: One example of a single-qubit gate that generates superposition is the Hadamard gate. The matrix representation is as follows:

$$(2.2) \quad H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

When the Hadamard gate is applied to a qubit in state $|0\rangle$, it undergoes a transformation to $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$.

CNOT Gate: A Controlled NOT (CNOT) gate is a kind of gate that operates on two qubits. It carries out a NOT operation on the target qubit (second qubit) only when the control qubit (first qubit) is in state $|1\rangle$. The matrix representation is as follows:

$$(2.3) \quad \text{CNOT} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

The operational mechanism of quantum gates involves the multiplication of their matrix representations with the state vector of the qubits involved in their operation. The application of a single-qubit gate U to a qubit state $|\psi\rangle$ may be expressed as follows:

$$(2.4) \quad |\psi'\rangle = U|\psi\rangle$$

The operation of a two-qubit gate is described by a tensor product, as shown by:

$$(2.5) \quad |\phi'\rangle = (U \otimes V)|\phi\rangle$$

Qubits are subjected to a series of quantum gates in quantum circuits. In a circuit schematic, each horizontal line corresponds to a qubit, and the gates are applied in a sequential manner from left to right. The design of quantum circuits plays a critical role in ensuring effective quantum processing, since they serve as the fundamental building blocks for quantum algorithms.

Quantum gates are fundamental to various quantum algorithms. The quantum parallelism and entanglement enabled by these gates form the basis for algorithms like Shor's factoring algorithm [7] and Grover's search algorithm [6].

2.0.1.1 Rotation Gates

Rotational gates play a crucial role in the field of quantum computing as they allow for the manipulation of the quantum state of a qubit via the rotation of said qubit along a designated axis inside the Bloch sphere. These gates are of utmost importance in a wide range of quantum algorithms and jobs related to quantum information processing.

The exponential of the Pauli matrices provides the generic single-qubit rotation operator around the unit vector $\hat{n} = (\sin\theta \cos\phi, \sin\theta \sin\phi, \cos\theta)$, The equation $R(\theta, \phi)$ may be expressed as follows:

$$(2.6) \quad R(\theta, \phi) = R_z(\phi)R_y(\theta)R_z(-\phi)$$

The Pauli matrices are denoted as σ_x , σ_y , and σ_z . The operator in question executes a rotation on the Z-axis (Phase Gate) by an angle θ with respect to the axis specified by \hat{n} .

The phase gate is a frequently used rotational gate that executes a rotating movement along the Z-axis. The matrix serves as a representation.

$$(2.7) \quad R_z(\theta) = \begin{bmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{bmatrix}$$

The gate incorporates a phase component $e^{i\theta}$ into the state, while maintaining the probability amplitudes at their original values.

The rotation of the X-axis and Y-axis is a subtopic of interest.

$$(2.8) \quad R_x(\theta) = \begin{bmatrix} \cos(\theta/2) & -i \sin(\theta/2) \\ -i \sin(\theta/2) & \cos(\theta/2) \end{bmatrix}$$

$$(2.9) \quad R_y(\theta) = \begin{bmatrix} \cos(\theta/2) & -\sin(\theta/2) \\ \sin(\theta/2) & \cos(\theta/2) \end{bmatrix}$$

The phase rotation is implemented by this gate only when the control qubit is in the $|1\rangle$ state.

The following section pertains to compounds.

Multiple rotating gates may be used to generate composite rotations. For example, the implementation of a rotation around an arbitrary axis $\hat{n} = (\sin\theta \cos\phi, \sin\theta \sin\phi, \cos\theta)$ may be expressed as follows:

The equation $R(\theta, \phi)$ may be expressed as

$$(2.10) \quad R(\theta, \phi) = R_z(\phi)R_y(\theta)R_z(-\phi)$$

The given sequence executes a rotational motion around the axis \hat{n} .

Parameterized gates are often used in practical applications, whereby the rotation angles are dynamically modified in accordance with specified optimization or computational criteria. The representation of these gates is as follows:

The equation $R_z(\theta)$ may be expressed as follows:

$$(2.11) \quad R_z(\theta) = \begin{bmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{bmatrix}$$

In the context of quantum computing, the parameter denoted as θ is subject to adjustment.

Quantum rotational gates serve as the foundation for several quantum algorithms and quantum machine learning applications, enabling precise and controlled manipulation and control of quantum states.

2.0.2 Quantum Approximate Optimization Algorithm (QAOA)

The Quantum Approximate Optimization Algorithm (QAOA) is a quantum algorithm developed by Farhi et al.[10] aimed at solving combinatorial optimization problems. It has emerged as a promising approach for tackling a wide range of optimization tasks due to its potential advantage over classical algorithms, particularly for complex optimization problems. It leverages gate-based quantum computing to find candidate solutions to combinatorial optimization problems using a variational circuit with parameters tuned in a classical outer loop.

Since QAOA is based on adiabatic computation and works well for solving combinatorial optimization problems, we will briefly touch on the concepts in the subsection below before diving into QAOA.

2.0.2.1 Combinatorial Optimization

Selecting the best item from a limited collection is an example of a traditional combinatorial optimization (CO) problem. Due to the possibly enormous number of items, an exhaustive search becomes challenging while trying to locate the ideal one. If s is a string in a set S with m clauses, where $s \geq m$, then we have a CO problem, which may be defined mathematically as an optimization or minimization problem. According to Guerrero [11], every clause takes a string argument and returns a value that corresponds to it. The total cost function of the string is the sum of all m clauses. With z as the input string and C_α as the clauses, we can express the total cost function as

$$(2.12) \quad C(z) = \sum_{\alpha=1}^m C_\alpha(z)$$

To achieve this, we need to locate an element \bar{z} in S such that, for every z in S , we have $C(\bar{z}) \geq C(z)$ (or, in the case of minimization, $C(\bar{z}) \leq C(z)$). Here, \bar{z} is not required to be unique.

For simplification, restrictions are put on clauses and input strings where clauses are restricted to integers 0 and 1, whereas input strings are restricted to binary representation of integers 0 through $2^n - 1$. Hence z can be written as $z = z_0z_1z_2\dots\dots z_{n-1}$ for $z_i \in \{0, 1\}$. Also, considering only maximization problems the minimization problems can be studied as $C'_\alpha(z) = 1 - C_\alpha(z)$

$$(2.13) \quad C'(\bar{z}) = \sum_{k=0}^{m-1} C'_\alpha(\bar{z}) = \sum_{k=0}^{m-1} (1 - C_\alpha(\bar{z})) = m - \sum_{k=0}^{m-1} C_\alpha(\bar{z}) = m - C(\bar{z})$$

Classical combinatorial optimization problems include the search for the most optimum item within a limited collection of items. The problem is formulated in terms of constraints that need to be fulfilled, with the goal of finding the "optimal object" that fulfils the maximum number of requirements. Typically, the collection of items is too extensive to thoroughly search (i.e., the task is NP-hard), thus we rely

on approximating the best answer. (The inclusion of "Approximate Optimization" in the name "Quantum Approximate Optimization Algorithm" is due to this.)

2.0.2.2 Adiabatic Computation

The theoretical foundation of Adiabatic Quantum Computation was established by Edward Farhi, a professor at MIT who developed QAOA [5]. The underlying premise of Adiabatic Quantum Computation (AQC) is derived from the adiabatic theorem, which claims that a system in the ground state of a given Hamiltonian will remain in its ground state provided the Hamiltonian undergoes gradual but continuous changes.

The energy state of a system is represented by a Hamiltonian. In the context of AQC, we are dealing with two Hamiltonians: the driver Hamiltonian (H_d) and the problem Hamiltonian (H_p). The driver Hamiltonian, denoted as (H_d), represents the ground state of a system that can be readily prepared. On the other hand, the problem Hamiltonian, denoted as (H_p), represents the desired ground state that we want to achieve via evolution. The amount of time required is dependent upon the difference in energy between the two minima of the interpolating Hamiltonian. The estimation of this gap in a general is not feasible. In simple terms, we begin with a ground state that is straightforward to create (i.e., the ground state of (H_d)) and aim to get the desired quantum state (i.e., the ground state of (H_p)), which is often much more challenging to create.

A mathematical function $s(t)$ is defined on the interval $[0, T]$, where $s(0) = 0$ and $s(T) = 1$. The value of time, denoted as T , is sufficiently high to satisfy the adiabatic theorem. Thus the Hamiltonian is defined as

$$(2.14) \quad H(t) = (1 - s(t))H_D + s(t)H_P$$

According to the Adiabatic Theorem, if we have the right $s(t)$, we will remain in the ground state of this $H(t)$ for the entire interval $[0, T]$. Thus, it is evident that at time $t = 0$ we are in the ground state of H_d and by time $t = T$ we have transitioned to the desired ground state of H_p .

The temporal evolution under this time-dependent Hamiltonian includes an integral that is difficult to calculate.

$$(2.15) \quad U(t) = \tau \exp \left\{ \frac{-i}{\hbar} \int_0^t H(T) dT \right\}$$

This Hamiltonian may be evaluated using the Trotterization approach. The process involves dividing the matrix $U(T)$ into intervals of Δt that are sufficiently short to ensure that the Hamiltonian remains essentially constant throughout each interval. This enables the use of a more streamlined equation for the time-independent Hamiltonian.

Let $U(b, a)$ denote the temporal progression from time a to time b ,

$$(2.16) \quad U(T, 0) = U(T, T - \Delta t)U(T - \Delta t, T - 2\Delta t) \cdots U(\Delta t, 0)$$

$$(2.17) \quad = \prod_{j=1}^p U(j\Delta t, (j-1)\Delta t)$$

$$(2.18) \quad \approx \prod_{j=1}^p e^{-iH(j\Delta t)\Delta t}$$

The approximation improves as the value of p increases (or, alternatively, as Δt decreases), and we have selected Δt to be measured in units of \hbar . The integral $U(t)$ may be obtained by using the approximation $e^{i(A+B)x} = e^{iAx}e^{iBx} + O(x^2)$ and including the Hamiltonian $H(j\Delta t) = (1 - s(j\Delta t))H_D + s(j\Delta t)H_p$.

$$(2.19) \quad U(T, 0) \approx \prod_{j=1}^p \exp\{-i(1 - s(j\Delta t))H_D\Delta t\} \exp\{-is(j\Delta t)H_p\Delta t\}$$

The approximation of this AQC involves allowing the system to develop under H_p for a small value of $s(j\Delta t)\Delta t$, and then H_D for a small value of $(1 - s(j\Delta t))\Delta t$. Unitaries may be created for these operations. The equation $U = e^{-i\alpha H\Delta t}$; represents the scaling due to $s(j\Delta t)$, where α is a number in the range $[0, 1]$ [11, 12].

2.0.2.3 QAOA Algorithm

Let us get into a comprehensive analysis of the QAOA algorithm. It is considered a hybrid algorithm since it incorporates both classical and quantum approaches. These algorithms have been the favoured option throughout the NISQ period and may be used to tackle various optimization challenges.

In the preceding part, we carefully discussed the concept of quantum adiabatic computing, specifically focusing on the transition from the eigenstate of the driver Hamiltonian to the eigenstate of the problem Hamiltonian. The Problem Hamiltonian may be expressed as

$$(2.20) \quad C|z\rangle = \sum_{\alpha=1}^m C_{\alpha}(z)|z\rangle$$

The solution to the combinatorial optimization problem is the minimal energy eigenstate of C . Similarly, we choose the driver Hamiltonian as

$$(2.21) \quad B = \sum_{j=1}^n \sigma_j^x$$

Let σ_j^x be the Pauli operator on bit z_j . B is often referred to as the mixing operator. We will define $U_c(\gamma) = e^{-i\gamma C}$ and $U_B(\beta) = e^{-i\beta B}$ to represent the evolution of the system under C for a certain γ time and under B for a specific β time, respectively.

QAOA is an algorithm that accepts a collection of m clauses, a set of p angles represented by $\{\gamma_0, \gamma_1, \gamma_2, \dots, \gamma_{p-1}\}$, and a set of p angles represented by $\{\beta_0, \beta_1, \beta_2, \dots, \beta_{p-1}\}$. P . Therefore, the variable P represents the number of iterations performed by the algorithm [12].

Assuming $P = 1$, we may look at the variables $\gamma_0 = \gamma$ and $\beta_0 = \beta$ accordingly. The unitary function $U_c(\gamma)$ has been defined.

$$(2.22) \quad U_c(\gamma) = e^{-i\gamma C} = \prod_{\alpha=1}^m e^{-i\gamma C_{\alpha}} = \prod_{\alpha=1}^m \exp(-i\gamma C_{\alpha})$$

The diagonal matrix C_{α} has dimensions $n \times n$ and is defined by the entries

$$(2.23) \quad C_{\alpha}(i, j) = \begin{cases} 1 & \text{if } i = j \text{ and } C_{\alpha}(i - 1) = 1 \\ 0 & \text{otherwise} \end{cases}$$

The index of z in the preceding formulation is denoted by $i - 1$, and it varies from 0 to $2^n - 1$. Taking example of $n = 2$ and the clause

$$(2.24) \quad C_1(z) = \begin{cases} 1 & \text{if } z \in \{1, 2\} \\ 0 & \text{if } z \in \{0, 3\} \end{cases}$$

We arrive at the matrix

$$(2.25) \quad C_1 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

Using the 2nd clause

$$(2.26) \quad C_2(z) = \begin{cases} 1 & \text{if } z \in \{1\} \\ 0 & \text{if } z \in \{0, 2, 3\} \end{cases}$$

from both equations above we can form $U_c(\gamma)$

$$(2.27) \quad U_C(\gamma) = \prod_{\alpha=1}^2 \exp(-i\gamma C_\alpha) = \exp \left[-i\gamma \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \right] \exp \left[-i\gamma \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \right]$$

$$= \exp \left\{ -i\gamma \left[\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \right] \right\}$$

$$(2.28) \quad = \exp \left[-i\gamma \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \right]$$

By combining the two cost functions, $C_1(z)$ and $C_2(z)$, we may get the original cost function $C(z)$ as the total of the two. The values of C at different inputs are as follows: $C(0) = 0$, $C(1) = 2$, $C(2) = 1$, $C(3) = 0$. Therefore, the expression $U_C(\gamma)$ may be restated as

$$(2.29) \quad U_C(\gamma) = \exp(-i\gamma C)$$

Here, C is a matrix with components that are only present on its diagonal.

$$(2.30) \quad C(i, j) = \begin{cases} C(i-1) & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$$

The integer values of $C(z)$ allow us to consider the variable γ within the range of 0 to 2π . This is further supported by the observation that for every $\gamma_1 \notin (0, 2\pi)$, there exists a $\gamma_0 \in (0, 2\pi)$ such that $\gamma_1 = \gamma_0 + 2\pi k$, where k in \mathbb{Z} .

Ultimately, we have

$$(2.31) \quad \begin{aligned} U_C(\gamma_1) &= \exp(-i\gamma_1 C) \\ &= \exp[-i(\gamma_0 + 2\pi k)C] \\ &= \exp(-i\gamma_0 C) \exp(-i2\pi k C) \end{aligned}$$

As we know C is a matrix with integer values the above changes to

$$(2.32) \quad \begin{aligned} &= \exp(-i\gamma_0 C) \begin{pmatrix} e^{-i2\pi k C(0)} & 0 & \dots & 0 \\ 0 & e^{-i2\pi k C(1)} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & e^{-i2\pi k C(2^n-1)} \end{pmatrix} \\ &= \exp(-i\gamma_0 C) \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{pmatrix} = \exp(-i\gamma_0 C) I = \exp(-i\gamma_0 C) = U_C(\gamma_0) \end{aligned}$$

Looking into 2nd operator

$$(2.33) \quad \begin{aligned} B &= \sum_{k=1}^n \left(\bigotimes_{i=1}^{k-1} I \otimes \sigma_x \otimes \bigotimes_{i=k+1}^n I \right) \\ &= \sigma_x \otimes I^{\otimes(n-1)} + I \otimes \sigma_x \otimes I^{\otimes(n-2)} + \dots + I^{\otimes(n-1)} \otimes \sigma_x \end{aligned}$$

Where $\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$. Further, we have already established

$$(2.34) \quad U_B(\beta) = e^{-i\beta B} = \exp(-i\beta B)$$

In their original publication[10], Farhi and Goldstone introduced the concept of β inside the interval $(0, \pi)$, indicating the existence of an input domain of (γ, β) in the interval $[0, 2\pi] \times [0, \pi]$. The given expression may be simplified to $(\gamma, \beta) \in [0, 2\pi] \times [0, \frac{\pi}{2}]$ or $(\gamma, \beta) \in [0, \pi] \times [0, \pi]$ without any loss of information.

To verify the changes in angles, we shall analyze the values of $0 \leq \gamma_0 \leq 2\pi$ and $0 \leq \beta_0 \leq \pi$. The angles that are obtained are as follows.

$$(2.35) \quad \gamma_1 = -\gamma_0 + 2\pi - (16)$$

$$\beta_1 = -\beta_0 + \pi \dots (17)$$

In this context, the variables γ_0 and β_0 are mirrored over π and $\pi/2$, respectively. Therefore, when the procedure is executed with newer angles γ_1, β_1 , it produces the identical results to γ_0, β_0 . Therefore, decreasing the domain factor by 50% will provide similar outcomes. The multiplication of the operators $U_C(\gamma_1), U_B(\beta_1)$ is

$$(2.36) \quad U_C(\gamma_1)U_B(\beta_1) = e^{-i\gamma_1 C} e^{-i\beta_1 B} = e^{-i(-\gamma_0+2\pi)C} e^{-i(-\beta_0+\pi)B}$$

Since C is diagonal

$$(2.37) \quad e^{-i(-\gamma_0+2\pi)C} = \begin{pmatrix} e^{-i(-\gamma_0+2\pi)C(0)} & 0 & \dots & 0 \\ 0 & e^{i(-\gamma_0+2\pi)C(1)} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & e^{-i(-\gamma_0+2\pi)C(2^n-1)} \end{pmatrix}$$

$$= \begin{pmatrix} e^{i\gamma_0 C(0)} e^{-2\pi i C(0)} & 0 & 0 \\ 0 & e^{i\gamma_0 C(1)} e^{-2\pi i C(1)} & \dots \\ \vdots & \vdots & \ddots \\ 0 & 0 & \dots & e^{i\gamma_0 C(2^n-1)} e^{-2\pi i C(2^n-1)} \end{pmatrix}$$

Given that $C(z) \in \mathbb{Z}$, it follows that $\exp(-2\pi i C(z)) = 1$. The matrix shown above may be simplified to,

$$(2.38) \quad = \begin{pmatrix} e^{i\gamma_0 C(0)} & 0 & \dots & 0 \\ 0 & e^{i\gamma_0 C(1)} & 0 & 0 \\ \vdots & 0 & \ddots & \vdots \\ 0 & 0 & \dots & e^{i\gamma_0 C(2^n-1)} \end{pmatrix} = e^{i\gamma_0 C}$$

In the same way for $U_B(\beta_1)$, it is necessary for all the eigenvalues of B to be either odd or even integers. Therefore, we will proceed to define B using a recursive approach

$$(2.39) \quad \begin{aligned} B_1 &= \sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \\ B_{n+1} &= \begin{pmatrix} B_n & I_n \\ I_n & B_n \end{pmatrix} \end{aligned}$$

The identity matrix of $2n \times 2n$ is denoted as I_n . The recursive definition provided is comparable to the definition previously established for B , which relates to n qubits. For a value of $n = 1$, the equation may be expressed as $B_1 = \sum_{k=1}^1 (\otimes_{i=1}^0 I \otimes \sigma_x \otimes \otimes_{i=2}^1 I) = \sigma_x$

Given the assumption that the definition holds true for $n - 1$ qubits.

Let us now shift our focus towards the eigenvalue of matrix B . By intuition, it is necessary for these values to be odd or even integers. In the case of B_1 , these values are ± 1 . Let's suppose that the eigenvalues of B_{n-1} are integers that are either odd or even. To demonstrate this, let us begin the computation of the eigenvalue of B_n .

$$(2.40) \quad \begin{aligned} 0 &= |B_n - \lambda I_n| = \left| \begin{pmatrix} B_{n-1} & I_{n-1} \\ I_{n-1} & B_{n-1} \end{pmatrix} - \lambda I_n \right| \\ &= \begin{vmatrix} B_{n-1} - \lambda I_{n-1} & I_{n-1} \\ I_{n-1} & B_{n-1} - \lambda I_{n-1} \end{vmatrix} \end{aligned}$$

By using the property of symmetric block matrices

$$(2.41) \quad \det \begin{pmatrix} A & B \\ B & A \end{pmatrix} = \det(A - B) \det(A + B)$$

Thus our eigenvalue calculation becomes

$$(2.42) \quad \begin{aligned} &= \det(B_{n-1} - \lambda I_{n-1} - I_{n-1}) \det(B_{n-1} - \lambda I_{n-1} + I_{n-1}) \\ &= \det(B_{n-1} - (\lambda + 1)I_{n-1}) \det(B_{n-1} - (\lambda - 1)I_{n-1}) \end{aligned}$$

Both determinants are the characteristic polynomial for B_{n-1} with roots shifted ± 1 . Therefore, the eigenvalues of B_n might be equivalent to the eigenvalues of

B_{n-1} following either a subtraction or an addition of 1. Given the assumption that the eigenvalues of B_{n-1} are all odd or even integers, it follows that the eigenvalues of B_n must also be all odd or even integers.

We have

$$(2.43) \quad U_B(\beta_1) = \exp(-i\beta_1 B) = \exp(-i(-\beta_0 + \pi)B)$$

Given that B is diagonalizable, it may be expressed as $B = UB_D U^{-1}$ where U represents a unitary matrix and B_D is a diagonal matrix of B with eigenvalues of B on diagonals and 0 elsewhere.

Hence

$$(2.44) \quad U \exp(-i(-\beta_0 + \pi)B_D) U^{-1} = U \exp(i\beta_0 B_D) \exp(-i\pi B_D) U^{-1}$$

We know $\exp(-i\pi B_D)$ is simply the identity matrix times (-1^n) . Therefore, the equation changes to

$$(2.45) \quad (-1^n) U \exp(i\beta_0 B_D) U^{-1} = (-1^n) \exp(i\beta_0 B_D) = (-1^n) \exp(i\beta_0 B)$$

Therefore, the resultant product of $U_C(\gamma_1) U_B(\beta_1)$ will be as follows.

$$(2.46) \quad U_C(\gamma_1) U_B(\beta_1) = (-1^n) \exp(i\beta_0 B) \exp(i\gamma_0 C)$$

In the final measurement (-1^n) has no effect since it is squared. $\exp(i\beta_0 B) \exp(i\gamma_0 C)$ are clearly matrices $U_B(\beta_0)$ and $U_C(\gamma_0)$ with rotation in the opposite directions. That is, the final measurement will be equal whether we use the angles γ_0, β_0 or γ_1, β_1 .

The squared value of (-1^n) does not have any impact on the final measurement. The matrices $\exp(i\beta_0 B) \exp(i\gamma_0 C)$ may be identified as $U_B(\beta_0)$ and $U_C(\gamma_0)$, respectively, with rotations in opposite directions. The final measurement will be equal regardless of whether the angles γ_0, β_0 or γ_1, β_1 are used.

After considering the Operators, we will now shift our focus to the Hadamard function, which is used to initialise all qubits to the superposition state.

We know that Hadamard gate is

$$(2.47) \quad H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

The state may be constructed using (γ, β) angles in the following manner.

$$(2.48) \quad |\psi\rangle = U_B(\beta)U_C(\gamma)H^{\otimes n}|0\rangle^{\otimes n}$$

Considering $p > 1$ then

$$(2.49) \quad |\psi\rangle = U_B(\beta_{p-1})U_C(\gamma_{p-1})\cdots U_B(\beta_1)U_C(\gamma_1)U_B(\beta_0)U_C(\gamma_0)H^{\otimes n}|0\rangle^{\otimes n}$$

2.0.3 Variational Quantum Eigensolver (VQE)

The Variational Quantum Eigensolver (VQE) is a key hybrid quantum-classical method used to solve eigenvalue and optimization problems on quantum computers. Peruzzo et al. [4] presented one of the first and most referenced papers that introduced the Variational Quantum Eigensolver (VQE) method. Its use in quantum chemistry, especially in determining the ground state energy of molecules, is highly regarded due to its outstanding reputation in the area. The VQE method works by preparing quantum states and measuring observables using a quantum computer, while a classical optimizer tweaks the quantum states' parameters to minimize a cost function, usually the system's energy. VQE is a versatile method for estimating the ground state energy of quantum systems. VQE, combines variational concepts with quantum computing methods, and provides a potential way to solve problems in quantum chemistry, materials research, and optimization.

2.0.3.1 Mathematical Formulation

Let us examine a quantum system characterized by a Hamiltonian \hat{H} , which has the lowest energy state denoted as $|\psi_{\text{gs}}\rangle$ and has an associated energy value of E_{gs} . The objective of Variational Quantum Eigensolver (VQE) is to obtain an estimate of the ground state energy, denoted as E_{gs} , by creating a trial quantum state $|\psi(\boldsymbol{\theta})\rangle$ that is defined by a set of angles $\boldsymbol{\theta}$. The aim is to minimise the expectation value of the Hamiltonian operator \hat{H} with respect to the trial state $|\psi(\boldsymbol{\theta})\rangle$.

$$(2.50) \quad E(\boldsymbol{\theta}) = \langle \psi(\boldsymbol{\theta}) | \hat{H} | \psi(\boldsymbol{\theta}) \rangle$$

The trial state, denoted as $|\psi(\boldsymbol{\theta})\rangle$, is often generated using a parameterized quantum circuit called the variational ansatz. The ansatz circuit consists of many layers of quantum gates, where each gate is parameterized by elements of $\boldsymbol{\theta}$.

2.0.3.2 Variational Ansatz

The ansatz $U(\boldsymbol{\theta})$ is a parameterized quantum circuit that creates trial states by applying operations to a starting state $|\psi_0\rangle$:

$$(2.51) \quad |\psi(\boldsymbol{\theta})\rangle = U(\boldsymbol{\theta})|\psi_0\rangle$$

The ansatz circuit often has many layers of single-qubit rotations and entangling gates, such as the parameterized $R_y(\theta)$ and *CNOT* gates. The selection of ansatz architecture and circuit depth may have a substantial influence on the range of possible solutions and the optimization process of the Variational Quantum Eigensolver (VQE).

2.0.3.3 Optimization Procedure

Variational Quantum Eigensolver (VQE) uses standard optimization techniques, such as gradient descent or variational approaches, to determine the best parameters $\boldsymbol{\theta}^*$ that minimize the function $E(\boldsymbol{\theta})$. The optimization procedure repeatedly modifies the settings using the energy expectation values derived from quantum measurements.

$$(2.52) \quad \boldsymbol{\theta}^{(k+1)} = \boldsymbol{\theta}^{(k)} - \eta \nabla E(\boldsymbol{\theta}^{(k)})$$

where η is the learning rate, and $\nabla E(\boldsymbol{\theta}^{(k)})$ is the gradient of $E(\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}^{(k)}$.

The η represents the learning rate, whereas $\nabla E(\boldsymbol{\theta}^{(k)})$ denotes the gradient of $E(\boldsymbol{\theta})$ with regard to $\boldsymbol{\theta}^{(k)}$.

2.0.3.4 Challenges and Considerations

Even with its potential, VQE has a number of obstacles to overcome, such as:

1. **Hardware Limitations:** The capabilities and noise levels of existing quantum hardware have a major role in the performance of VQE. Energy estimate errors may be caused by noise and mistakes in quantum gates.
2. **Optimization Landscape:** Finding the global energy minimum effectively may be difficult due to the very non-convex and rugged nature of the VQE optimization landscape.
3. **Ansatz Selection:** The effectiveness of VQE depends on selecting a suitable variational ansatz. For the ansatz to accurately represent the ground state of the system, expressiveness and circuit depth must be balanced.

2.0.3.5 Future Directions

Although VQE has potential, it encounters several obstacles such as the ruggedness of the optimization environment, the need for noise and error reduction[13], and the issue of scalability [14]. The optimization landscape of the Variational Quantum Eigensolver (VQE) may exhibit significant non-convexity, which can result in challenges with convergence and the possibility of being trapped in local minima. Furthermore, the presence of flaws in quantum hardware and the phenomenon of decoherence may lead to mistakes that diminish the precision of VQE computations.

In order to tackle these difficulties, current research endeavors are concentrated on creating enhanced optimization methods specifically designed for VQE, improving quantum error correction and mitigation strategies [15, 16], and upgrading quantum hardware technologies. Furthermore, investigating hybrid quantum-classical methods and inventive ansatz constructs show potential for enhancing the efficiency and usefulness of the Variational Quantum Eigensolver (VQE) in real-world scenarios [17].

Overall, VQE is a very promising method for addressing quantum chemistry issues and emulating quantum systems with the use of quantum devices that are expected to be available in the near future. Although there has been notable

advancement, further investigation is required to address the current obstacles and fully use the capabilities of VQE in the field of quantum chemistry and beyond.

2.0.4 Quantum Kernel Methods

Quantum Kernel Methods (QKMs) are a category of machine learning algorithms that use principles from quantum computing to carry out a range of tasks, such as classification, regression, and clustering. Quantum kernel methods (QKMs) enhance classical kernel techniques by integrating quantum characteristics, such as quantum kernel functions and quantum feature maps.

The core of QKMs is centered on the premise of quantum feature spaces, in which data points are transformed into high-dimensional quantum states via the use of quantum feature maps. Subsequently, these quantum states are used to establish quantum kernel functions, which measure the degree of similarity between data points in the feature space.

The quantum kernel function $K(x_i, x_j)$ quantifies the inner product between the quantum states associated with input data points x_i and x_j . A frequently used quantum kernel function is the quantum inner product.

$$(2.53) \quad K(x_i, x_j) = |\langle \psi(x_i) | \psi(x_j) \rangle|^2 = |\langle 0 | U^\dagger(x_i) U(x_j) | 0 \rangle|^2$$

The quantum states $|\psi(x_i)\rangle$ and $|\psi(x_j)\rangle$ correspond to the input data points x_i and x_j , respectively. The quantum inner product quantifies the similarity between quantum states and is used to establish the quantum kernel matrix.

2.0.4.1 Quantum Kernel Estimation

The kernel estimation can be performed using the following steps in algorithm ??.

In this approach, we conduct a total of R measurements and increment the value of r by 1 each time we encounter a string consisting entirely of zeros. The likelihood of seeing a string consisting entirely of zeros is thus equal to the ratio of "r" to "R". The estimated value of the kernel $\kappa_p(x_i, x_j)$ may deviate from the actual value of the kernel $\kappa(x_i, x_j)$ by an additional sampling error that is directly proportional to $\frac{1}{\sqrt{R}}$, where $\kappa_p(x_i, x_j) = \kappa(x_i, x_j) + \frac{1}{\sqrt{R}}$.

Algorithm 1 Quantum Kernel Estimation

```
1: Set  $r = 0$ 
2: for  $i = 1$  to  $R$  do
3:   Prepare  $|0\rangle^{\otimes n}$ 
4:   Apply  $U(x_j)$ 
5:   Apply  $U^\dagger(x_i)$ 
6:   Measure in the  $Z$  basis
7:   if all bits are in 0 state then
8:      $r = r + 1$ 
9:   end if
10: end for
11: Compute  $\kappa(x, y) = \frac{r}{R}$ 
```

2.0.5 Quantum Feature Maps

Quantum feature maps play a crucial role in QKMs by encoding classical data into quantum states. A quantum feature map $\Phi(x)$ is a quantum circuit that transforms an input data point x into a quantum state $|\psi(x)\rangle$ in a high-dimensional feature space.

One commonly used quantum feature map is the Quantum Fourier Transform (QFT), which maps classical data points to quantum states using the Fourier basis. The QFT is defined as:

$$(2.54) \quad \Phi(x) = \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{2\pi i x \cdot k} |k\rangle$$

where x is the classical data point, n is the number of qubits, and $|k\rangle$ represents the basis state in the quantum feature space.

Furthermore, we can use two feature maps based on encoding methods, which are:

$$(2.55) \quad \begin{aligned} \phi_1 : x &\rightarrow |\phi(x)\rangle \\ \phi_2 : x &\rightarrow \rho(x) \end{aligned}$$

We call these two feature maps type 1 and type 2 mapping.

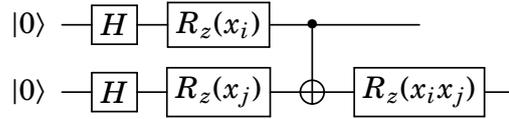


Figure 2.1: Sample circuit for Rotation Encoding

2.0.5.1 Type 1 Encoding

Type 1 encoding includes several types of data encoding, including basis encoding, amplitude encoding, angle encoding, and more.

In basis encoding, a data point is first converted into binary values, which are then further converted into qubits. As a result, this creates vectors that are orthogonal to each other. For instance, let's consider a data point: [2,5]. This data point is then associated with the binary representation [010, 101], which in turn corresponds to the vectors [00000100, 00100000]. Orthogonal vectors may be linearly separated with ease. Nevertheless, there exists a problem related to learning from the training set.

Amplitude encoding is ineffective for processing substantially nonlinear data, making it of moderate usefulness. The use of rotation encoding is beneficial since it enables a non-linear mapping. As an example, let us examine the circuit shown in Figure 2.1. By using entanglements and repeating the circuit shown with dashed lines, we may get an implementation that is very challenging to achieve using traditional methods. Contemporary encoding approaches use a consistent approach, using parametrized rotation gates in combination with entanglement and repetition. For instance, this concept is evident in the Pauli feature map and ZZ feature map created by IBM.

The selection of a feature map for a given problem requires a lot of consideration. A feature map that is readily reproducible conventionally is a bad option. Thus, a traditionally difficult feature map is essential but not sufficient. It is important to avoid feature maps that are easily implemented methodically. We should avoid feature maps based on product states, amplitude encoding, or too sophisticated (noise may complicate things).

2.0.5.2 Type 2 Encoding

In Type 2 encoding, the data is represented by a density matrix $\rho(x)$, which is obtained by taking the outer product of a quantum state

$$(2.56) \quad \rho(x) = |\phi(x)\rangle\langle\phi(x)|.$$

In type 2 encoding, we use a mapping of the variable x to the density matrix $\rho(x)$. The kernel function $\kappa(x_i, x_j)$ may be calculated as follows:

$$(2.57) \quad \begin{aligned} \kappa(x_i, x_j) &= \text{trace}\left(|\rho(x_i)\rangle\langle\rho(x_j)|\right) \\ &= \text{trace}(|\phi(x_i)\rangle\langle\phi(x_i)|\langle\phi(x_j)|\phi(x_j)\rangle) \\ &= \langle\phi(x_i)|\phi(x_j)\rangle\text{trace}(|\phi(x_i)\rangle\langle\phi(x_j)|) \\ &= \langle\phi(x_i)|\phi(x_j)\rangle\text{trace}(\langle\phi(x_j)|\phi(x_i)\rangle) \\ &= \langle\phi(x_j)|\phi(x_i)\rangle\langle\phi(x_i)|\phi(x_j)\rangle = |\langle\phi(x_i)|\phi(x_j)\rangle|^2 \end{aligned}$$

$\kappa(x_i, x_j) = |\langle\phi(x_i)|\phi(x_j)\rangle|^2$ is always a positive quantity (in fact, this is the same as the overlap). Now, the state $|\phi(x_i)\rangle$ can be prepared by using a unitary gate $U(x_i)$, and thus $|\phi(x)\rangle = U(x_i)|0\rangle$. With this, $\kappa(x, x_j) = |\langle 0|U^\dagger(x)U(x_j)|0\rangle|^2$. The computation of this kernel function may be achieved using the swap test, however, the most cost-effective approach to measure this kernel is by employing a quantum kernel estimate.

2.0.5.3 Quantum Support Vector Machine (QSVM)

Assuming we are dealing with a binary classification issue and the data can be separated by a straight line. Linearly separable data refers to a kind of data where the categorization may be achieved by drawing a straight line or a flat surface. Furthermore, for the sake of clarity, let us consider data that exists in two dimensions, as seen in Figure 1. Suppose we have created a line known as the classifier, which is responsible for categorizing the data. The line in Figure 2.2 is shown in the color purple. Next, we choose two data points - one from the positive data (shown as cross) and the other from the negative data (shown as triangle) -

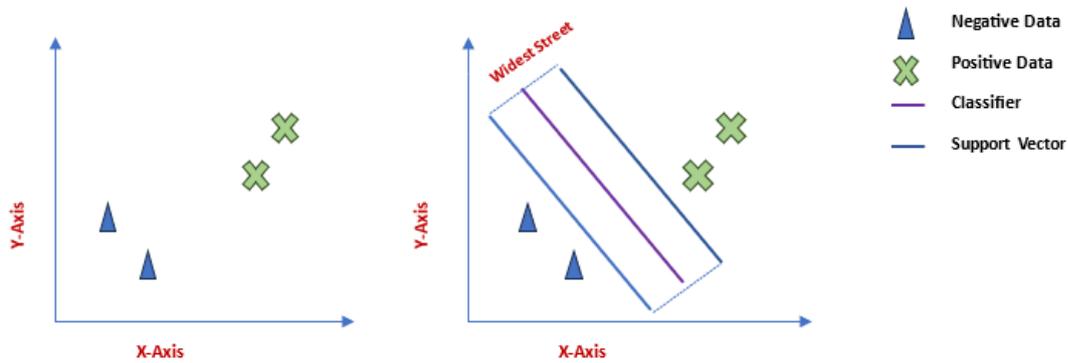


Figure 2.2: (a) Linearly separable data. (b) Classification with SVM.

that are closest to the red line. Next, we will create two more lines (depicted in blue in Figure 1) that are parallel to the red line and intersect with the closest positive and negative data points. The blue lines, referred to as support vectors, constitute a street. The objective of the support vector machine is to identify the street with the maximum width. The selection of the classifier determines the outcome of the street. Therefore, we can assert that the concept of the support vector machine is to locate the classifier (a line in two dimensions and a hyperplane in n -dimensions) that creates the broadest possible street.

In order to proceed with the discussion, it is necessary for us to have some notations. Assuming that w is a vector that is perpendicular to the median of the roadway and has an unknown magnitude, and u is an unknown data point. The classifier is as follows:

$$(2.58) \quad \begin{aligned} H = + \\ w \cdot u \geq c \\ H = - \end{aligned}$$

In this equation, we are calculating the projection of vector u onto vector w , and determining whether the magnitude of this projection is larger than a certain constant c . If the value is larger, we choose our hypothesis (decision rule) as 'cross'; otherwise, we choose the hypothesis as 'triangle.'

Regarding vectors in \mathbb{R}^n , we may express this equation as $f(u) = \text{sign}(w^T u + b)$, with b being equal to $-c$. Let us assume that we have training data denoted by the

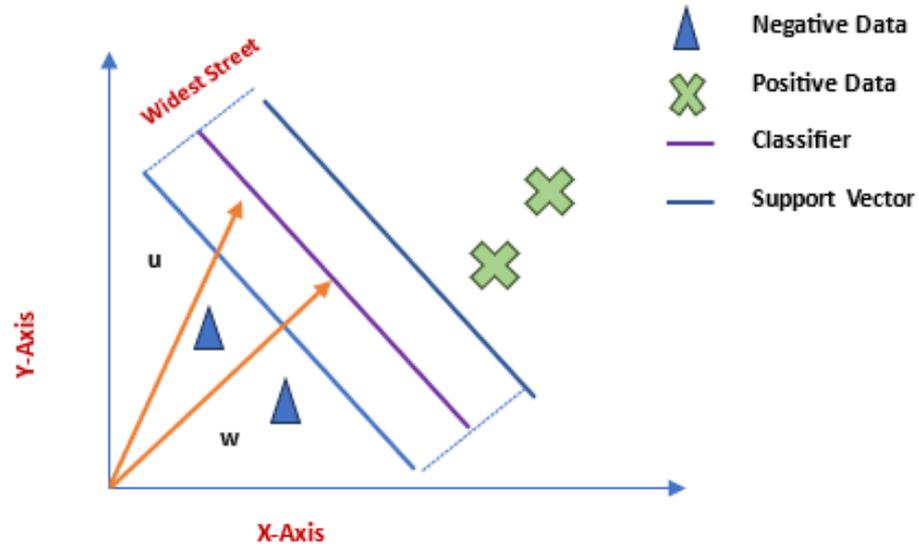


Figure 2.3: Support Vector Machine

variable x . It is important to mention that we have details about the labels of the training data. Therefore, we may use the symbols x_+ and x_- to represent the 'cross' and 'triangle' data, respectively. Additionally, we make the assumption that the provided training data complies with the following equations:

$$(2.59) \quad w^T x_+ + b \geq 1, \text{ and } w^T x_- + b \leq -1.$$

To simplify the equations, we can combine them into a single equation by adding a variable y_i . This variable represents the label of the data points and can be specified as $y_i = \begin{cases} 1 & \text{if } x_+ \\ -1 & \text{if } x_- \end{cases}$. By adding this variable, it is important to note that the previous two equations can be written in a simplified manner:

$$(2.60) \quad y_i (w^T x_i + b) \geq 1$$

Now, let's choose two positions $x_{+,s}$ and $x_{-,s}$ that are located on either the blue line or the support vectors. Therefore, these two points must satisfy the following equations:

$$(2.61) \quad w^T x_{+,s} + b = 1, \text{ and } w^T x_{-,s} + b = -1.$$

The width of the street can be calculated by evaluating the projection of $(x_{+,s} - x_{-,s})$ onto the unit vector that is perpendicular to the street, denoted as $\frac{w}{\|w\|}$.

$$(2.62) \quad \text{width} = \frac{w^T (x_{+,s} - x_{-,s})}{\|w\|} = \frac{1 - b - (-1 - b)}{\|w\|} = \frac{2}{\|w\|}$$

The objective of the support vector algorithm is to maximise the expression $\frac{2}{\|w\|}$, which is equivalent to minimising the expression $\frac{1}{2}\|w\|^2$.

The objective is to minimise the expression $\frac{1}{2}\|w\|^2$ while satisfying the restrictions $y_i (w^T x_i + b) - 1$. To do this, we may use the Lagrange minimization method, which involves the introduction of a constant α_i (generally referred to as the Lagrange multiplier) prior to $y_i (w^T x_i + b) - 1$.

$$(2.63) \quad L = \frac{1}{2}\|w\|^2 - \sum_i \alpha_i (y_i (w^T x_i + b) - 1)$$

By taking the derivative of L with respect to w and setting it equal to zero, we get $\frac{dL}{dw} = w^T - \sum_i \alpha_i y_i x_i^T = 0$. This equation yields the value of w as $\sum_i \alpha_i y_i x_i$. Furthermore, we may compute the derivative of L with respect to b and get $\frac{dL}{db} = \sum_i \alpha_i y_i = 0$. The result is a fresh equation for L , which can be expressed as

$$(2.64) \quad \begin{aligned} L &= \frac{1}{2}\|w\|^2 - \sum_i \alpha_i \left(y_i \left(\sum_j \alpha_j y_j x_j^T x_i \right) \right) + \sum_i \alpha_i \\ &= \frac{1}{2} \left(\sum_i \alpha_i y_i x_i^T \right) \left(\sum_j \alpha_j y_j x_j \right) - \sum_i \sum_j \alpha_i \alpha_j y_i y_j x_j^T x_i + \sum_i \alpha_i \\ &= -\frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j x_j^T x_i + \sum_i \alpha_i \end{aligned}$$

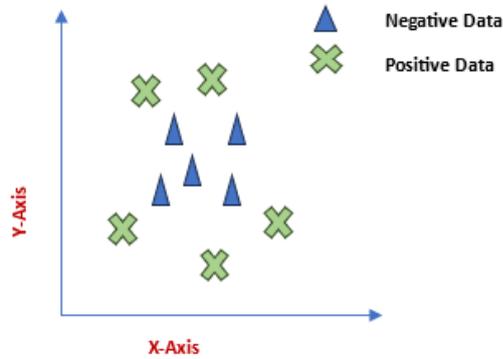


Figure 2.4: Circular Data

Now, the decision rule can be written as

$$(2.65) \quad = \text{sign} \left(\sum_j \alpha_j y_j x_j^T u + b \right)$$

Upon analysing the two equations for L and the decision rule, it becomes clear that both equations depend on the inner product between the data points.

$$(2.66) \quad L = -\frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j \langle x_j, x_i \rangle + \sum_i \alpha_i$$

$$f(u) = \text{sign} \left(\sum_j \alpha_j y_j \langle x_j, u \rangle + b \right)$$

This is important since it allows us to replace a kernel function κ with a generic inner product. The benefit of utilising $\kappa(x, y)$ instead of $\langle x, y \rangle$ is as follows. $\kappa(x, y)$ represents the inner product of feature maps of data points (x, y) . For data point x , feature map $\phi(x)$ increases its dimension. In this higher dimension, linearly inseparable data points may become separable. Take an example to illustrate. Examine the data in Fig ??

The data 2.4 is not separable linearly. Drawing a line will not separate 'crosses' from 'triangles' which is obvious. This data may be separated by drawing a circle, as indicated in Fig 2.5.

In mathematical terms, the effect of this is that the data points $\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ cannot be linearly classified. However, if we convert them to a higher dimension as

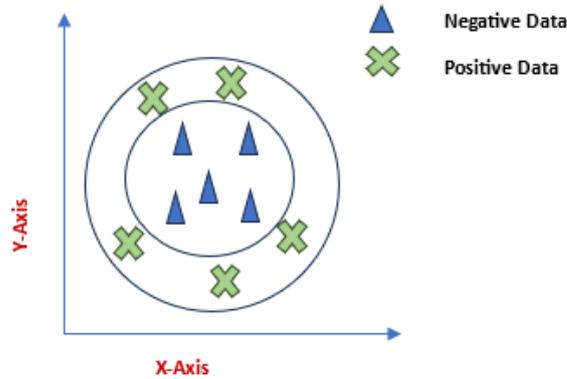


Figure 2.5: Separating Circular data By SVM

$\begin{bmatrix} x_1 \\ x_2 \\ x_1^2 + x_2^2 \end{bmatrix}$, they can be classified. The data points may be readily categorized by comparing the third data characteristic with b . Therefore, this example shows how the data may be divided into distinct groups in the new higher dimension in a linear way. Therefore, the feature space is transformed into our new space, where we can determine the inner product by using a kernel function.

Therefore, the function $k(x, y)$ may be expressed as the inner product of $\phi(x)$ and $\phi(y)$. However, entering a feature space might be difficult since the calculations can get complex due to the greater number of dimensions. Kernel functions are selected in such a manner that they can always be represented using the inner product in the original dimension. Two frequently used choices for kernel functions are:

a) Radial basis function:

$$(2.67) \quad \kappa(x_1, x_2) = \langle \phi(x_1), \phi(x_2) \rangle = e^{-\|x_1 - x_2\|/\sigma}$$

b) Polynomial kernel:

$$(2.68) \quad \kappa(x_1, x_2) = \langle \phi(x_1), \phi(x_2) \rangle = (\langle x_1, x_2 \rangle + 1)^n,$$

where n is an integer.

It is important to understand that the feature space in the radial basis function is infinite-dimensional. Nevertheless, the processing of the kernel is much simpler to understand.

Let us take one example. Suppose $\phi : (X_1, X_2) \rightarrow (1, \sqrt{2}X_1, \sqrt{2}X_2, X_1^2, \sqrt{2}X_1X_2, X_2^2)$. Now,

$$\begin{aligned}
 k(x, x) &= \langle \phi(x) | \phi(x) \rangle = \begin{bmatrix} 1 & \sqrt{2}X_1 & \sqrt{2}X_2 & X_1^2 & \sqrt{2}X_1X_2 & X_2^2 \end{bmatrix} \begin{bmatrix} 1 \\ \sqrt{2}X_1 \\ \sqrt{2}X_2 \\ X_1^2 \\ \sqrt{2}X_1X_2 \\ X_2^2 \end{bmatrix} \\
 (2.69) & \\
 &= 1 + 2X_1^2 + 2X_2^2 + X_1^4 + 2X_1^2X_2^2 + X_2^4 \\
 &= 1 + (X_1^2 + X_2^2)^2 + 2(X_1^2 + X_2^2) = (1 + (X_1^2 + X_2^2))^2 = (1 + \langle x | x \rangle)^2.
 \end{aligned}$$

Therefore, in this particular instance, we have shown that the function $\kappa(x_1, x_2) = (\langle x_1, x_2 \rangle + 1)^2$ can be expressed as the inner product of feature vectors with a dimension of 6. It is important to highlight that feature maps must be nonlinear in order to transform linearly inseparable data into linearly separable data. The primary advantage of SVM is its ability to use kernel functions, enabling it to effectively handle many types of data sets. We want to enhance the precision of these kernel functions by implementing them on quantum computers, with the expectation of achieving improved accuracy in the near future.

Part II

Part II

ANALYSIS OF THE VEHICLE ROUTING PROBLEM SOLVED VIA HYBRID QUANTUM ALGORITHMS IN THE PRESENCE OF NOISY CHANNELS

Abstract

The vehicle routing problem (VRP) is an NP-hard optimization problem that has been an interest of research for decades in science and industry. The objective is to plan routes of vehicles to deliver goods to a fixed number of customers with optimal efficiency. Classical tools and methods provide good approximations to reach the optimal global solution. Quantum computing and quantum machine learning provide a new approach to solving combinatorial optimization of problems faster due to inherent speedups of quantum effects. Many solutions of VRP are offered across different quantum computing platforms using hybrid algorithms such as quantum approximate optimization algorithm and quadratic unconstrained binary optimization. In this work, we build a basic VRP solver for 3 and 4 cities using the variational quantum eigensolver on a fixed ansatz. The work is further extended to evaluate the robustness of the solution in several examples of noisy quantum channels. We find that the performance of the quantum algorithm depends heavily on what

noise model is used. In general, noise is detrimental, but not equally so among different noise sources.

3.1 Introduction

Quantum computers, the next generation of computing technology, are expected to solve complex optimization problems much faster than their traditional counterparts. As parallelism is quantum computing's most notable benefit [18, 19], it is only natural to turn to quantum computing to speed up calculations in complicated optimization problems (such as those described by quantum approximate optimization algorithm (QAOA) [10], adiabatic computation (AC) [5], Grover's algorithm [20], and others). When applied to a multidimensional problem, classical optimization techniques in machine learning (ML) can take a long time to calculate global optimum and consume a lot of CPU and GPU power [21]. In higher dimensional problem spaces, classical algorithms have been shown to be less effective in general [22]. This is due to the fact that NP-hard optimization tasks are often assigned to ML algorithms [21].

VRP comes under the category of routing problems that try to address multiple issues related to fleet management [23]. The objective is always to optimize vehicle movement to minimize the cost or maximize the profit. Notwithstanding the difficulty in delivering quick and dependable solutions to the computationally hard VRP problem, several precise and heuristic techniques have been developed for solving it [23, 24]. Describing the VRP in its simplest form, a single vehicle is tasked to deliver goods at multiple customer locations; also, the vehicle needs to return to pick up additional items when it runs out of goods [25]. The goal is to minimize the cost of service by finding the best feasible combination of routes that begin and terminate at a central location (the depot) while maximizing the reward (often the inverse of the total distance the mean service time). This problem is computationally difficult to solve, even with only a few hundred customer nodes, and is classified as an NP-hard problem [21, 26].

Any VRP (n, k) involves $(n - 1)$ locations with k vehicles and a depot D [23, 27]. Its solution is a set of routes in which all k vehicles begin and terminate at the given depot D , ensuring that each place is visited only once. The route with the

shortest sum total of distance traveled by k vehicles is the ideal one. For a long time, VRP has been studied as an extension of the classical traveling salesman problem [24, 28], where now a group of k salesmen has to service collectively $(n - 1)$ locations, such that each location is serviced exactly once [23]. Constraints such as vehicle capacity or restricted covering time often complicate the VRP issue in practical settings. As a result, a plethora of conventional and quantum approaches have been presented in an effort to effectively solve the problem. Current quantum approaches for solving optimization problems include QAOA [10], Quadratic unconstrained binary optimization (QUBO) [29, 30], quantum annealing [31–33], and Variational quantum eigensolver(VQE) [27], which we will define in detail later.

In this work, we study the VRP in a different light. Here we explore adding controlled noise to an adapted quantum solution to determine if it improves or degrades the overall results. Recent works in QAOA [34–37] and VQE algorithms [38] studied the generic effects of noise in these hybrid algorithms. Our work complements these results by analyzing the effects of noise in a detailed gate-based simulation of an algorithm to solve VRP. We analyze the effect of various noise channels on an existing, yet variable, ansatz developed as a solution to VRP. We apply amplitude damping, bit-flip, phase-flip, bit-phase-flip, and depolarising noise channels to VRP circuits, analyze the effects, and consolidate our findings.

The paper is organized as follows. Sec. 3.2 discusses fundamental mathematical concepts such as combinatorial optimization, adiabatic computation, QAOA, and the Ising model. Sec. 3.3 discusses the formulation of VRP using the concepts discussed in the previous Section. Sec. 3.4 covers the basic building blocks of circuits to solve VRP. Sec. 3.5 covers building an ansatz for VRP. Finally, Sec. 3.6 covers the effects of applying noise models on the VRP circuit. Then Sec. 3.7 presents the observations from the simulation results. In Sec. 3.8, we summarize the effects of various noise models on the VRP circuit and future directions of research.

3.2 Mathematical Background

The fundamental concepts used to solve routing problems involve techniques and procedures from the field of combinatorial optimization. This is followed by

converting the mathematical models to a quantum equivalent mathematical model for formulating the objective function. The solution of the objective function is often achieved by maximization or minimization of the function. In this section, we outline the key concepts.

3.2.1 Combinatorial Optimization

A classical combinatorial optimization (CO) problem is finding an optimal object from a finite set of objects. Exhaustive search is impractical in finding the optimal object due to the potentially high number of objects. Mathematically defining, if s is a string in some set S and m number of clauses, where $s \geq m$, we have a maximization or minimization problem, known as a CO problem. Each clause expects a string parameter and returns a corresponding value [11]. It is the sum over the m clauses that constitute the string's total cost function. If we refer to the input string as z and clauses as C_α , we can write the total cost function as

$$(3.1) \quad C(z) = \sum_{\alpha=1}^m C_\alpha(z).$$

The objective is to identify $\bar{z} \in S$ such that $C(\bar{z}) \geq C(z)$ for all $z \in S$ (or, in the case of minimization, $C(\bar{z}) \leq C(z)$ for all $z \in S$). Here \bar{z} is not required to be unique. $C_\alpha(z)$ can take two values 0 or 1 and z can be written as $z = z_0 z_1 z_2 \dots z_{n-1}$ for $z_i \in \{0, 1\}$. Also, considering maximization problems (assuming $C_\alpha(z)$ is a clause), the minimization problems can be studied as $C'_\alpha(z) = 1 - C_\alpha(z)$

$$(3.2) \quad \begin{aligned} C'(z) &= \sum_{\alpha=0}^{m-1} C'_\alpha(z) = \sum_{\alpha=0}^{m-1} (1 - C_\alpha(z)), \\ &= m - \sum_{\alpha=0}^{m-1} C_\alpha(z) = m - C(z). \end{aligned}$$

3.2.2 Adiabatic Quantum Computation

Adiabatic quantum computation (AQC) is a theoretical framework of a quantum computer [5, 39]. The adiabatic theorem asserts that if the change to the Hamiltonian is sufficiently gradual, the system remains in the ground state of the given

Hamiltonian[40]. The Hamiltonian is an energy operator of a system. In AQC, there are two Hamiltonians: the driver Hamiltonian (H_d) and the problem Hamiltonian (H_p). The driver Hamiltonian (H_d) is the energy operator whose ground state is easy to prepare, whereas the problem Hamiltonian (H_p) is the energy operator whose ground state is obtained after evolution [5]. Interpolation times are proportional to the energy gap between the two lowest states of the Hamiltonian being used.

The procedure begins with an easy-to-prepare ground state (i.e., the ground state of (H_d)) and ends (ideally) with the ground state of (H_p), which is, in general, not directly characterizable. Mathematically, constitute function $s(t)$ on $[0, T]$ where $s(0)=0$ and $s(T)=1$. T is the value of time set high enough for the adiabatic theorem to hold. We define the Hamiltonian, $H(t) = (1-s(t))H_D + s(t)H_P$. According to the adiabatic theorem, a system maintains its ground state of $H(t)$ across the whole interval $[0, T]$, provided a suitable $s(t)$; hence, the system is in the initial ground state (H_d) at time $t = 0$, and it will evolve into the intended ground state (H_p) at time $t = T$. In general, it is challenging to assess the integral describing the temporal evolution under this time-dependent Hamiltonian [12]:

$$(3.3) \quad U(t) = \tau \exp \left\{ \frac{-i}{\hbar} \int_0^t H(T) dT \right\}.$$

It is possible to assess this Hamiltonian using Trotterization methods [41]. We divide $U(T)$ into intervals of δt small enough such that the Hamiltonian is almost constant over them. This permits us to use the much more streamlined formula for the Hamiltonian that is independent of time. Assuming $U(b, a)$ is the time evolution from instant a to instant b .

$$(3.4) \quad \begin{aligned} U(T, 0) &= U(T, T - \delta t)U(T - \delta t, T - 2\delta t) \cdots U(\delta t, 0), \\ &= \prod_{j=1}^p U(j\delta t, (j-1)\delta t), \\ &\approx \prod_{j=1}^p e^{-iH(j\delta t)\delta t}. \end{aligned}$$

Where the approximation gets better as p gets larger (or, in other words, as δt gets smaller), and where δt is measured in \hbar . Now using the approximation

$e^{i(A+B)x} = e^{iAx}e^{iBx} + \mathcal{O}(x^2)$ and adding Hamiltonian $H(j\delta t) = (1-s(j\delta t))H_D + s(j\delta t)H_P$ the integral $U(t)$ becomes,

$$(3.5) \quad U(T, 0) \approx \prod_{j=1}^p e^{-i(1-s(j\delta t))H_D\delta t} e^{-is(j\delta t)H_P\delta t}.$$

AQC may be approximated by allowing the system to develop under H_P for a small $s(j\delta t)\delta t$ and then H_D for a small $(1-s(j\delta t))\delta t$, and unitaries can be derived for these operations using $U = e^{-i\alpha H\delta t}$. Here, α is an integer in the range $[0, 1]$, and this includes the scaling resulting from $s(j\delta t)$. AQC forms the theoretical basis of the variational quantum algorithm QAOA, which is discussed briefly in the next section.

3.2.3 QAOA

The Quantum Approximate Optimization Algorithm (QAOA) is a hybrid variational algorithm, introduced by Farhi *et al.* [5, 10]. It combines principles of adiabatic quantum computation and classical optimization. In adiabatic computation, the system transitions from the driver Hamiltonian's eigenstate to the problem Hamiltonian's eigenstate, useful in optimization contexts.

For optimization, the problem Hamiltonian, C , and the driver Hamiltonian, B , are defined as:

$$(3.6) \quad C|z\rangle = \sum_{\alpha=1}^m C_{\alpha(z)}|z\rangle, \quad B = \sum_{j=1}^n \sigma_j^x,$$

where σ_j^x is the Pauli x operator on bit z_j . The evolution operators $U_C(\gamma) = e^{-i\gamma C}$ and $U_B(\beta) = e^{-i\beta B}$ enable the system to evolve under C and B for specified durations, respectively.

QAOA constructs the state:

$$(3.7) \quad |\boldsymbol{\beta}, \boldsymbol{\gamma}\rangle = e^{-i\beta_p B} e^{-i\gamma_p C} \dots e^{-i\beta_1 B} e^{-i\gamma_1 C} |s\rangle,$$

where $|s\rangle$ is the superposition of all input qubits. The expectation value of the cost function, $\sum_{\alpha=1}^m \langle \boldsymbol{\beta}, \boldsymbol{\gamma} | C_{\alpha} | \boldsymbol{\beta}, \boldsymbol{\gamma} \rangle$, provides an approximate solution to the optimization problem through simplex or gradient optimization [42].

3.2.4 Ising Model

The Ising model of ferromagnetism is a well-established mathematical model used extensively in the field of statistical mechanics [43, 44]. There are two possible states for the magnetic dipole moments of atomic "spins" (+1 and -1), each of which is represented by a discrete variable in the model. Each spin is able to communicate with its neighbors because of how they are organised in a graph, usually a lattice (where the local structure regularly repeats in all directions). The system tends towards the lowest energy state when neighboring spins agree, but heat interrupts this tendency, allowing for the emergence of alternate structural phases. The model serves as a simplification of reality that may be used to spot phase transitions [45]. Using the following Hamiltonian, we can describe the sum of the spin energies:

$$(3.8) \quad H_c = - \sum_{\langle i,j \rangle} J_{ij} \sigma_i \sigma_j - h \sum \sigma_i,$$

where J_{ij} represents the interaction between i and j , which are adjacent spins, and h represents an external magnetic field. If J is positive, the ground state at $h = 0$ is a ferromagnet. If J is negative, the ground state at $h = 0$ is an anti-ferromagnet for a bipartite lattice. Hence for simplification and in the context of this document, we can write the Hamiltonian as

$$(3.9) \quad H_c = - \sum_{\langle i,j \rangle} J_{ij} \sigma_i^z \sigma_j^z - \sum h_i \sigma_i^x.$$

Here σ_z and σ_x represent Pauli z and x operator. For simplification, we can consider the following conditions to be ferromagnetic ($J_{ij} > 0$), $h = 0$ assuming no external influence on the spin. Thus we can rewrite the Hamiltonian as follows,

$$(3.10) \quad H_c = - \sum_{\langle i,j \rangle} J_{ij} \sigma_i^z \sigma_j^z = - \sum_{\langle i,j \rangle} \sigma_i^z \sigma_j^z.$$

3.2.5 VQE

The Variational Quantum Eigensolver (VQE) is a hybrid quantum-classical algorithm designed to determine the eigenvalues of a Hamiltonian H , utilizing a

parameterized trial quantum state $|\psi(\vec{\theta})\rangle$ [46]. The expectation value of H , given the state $|\psi(\vec{\theta})\rangle$, is represented as:

$$(3.11) \quad E(\vec{\theta}) = \langle \psi(\vec{\theta}) | H | \psi(\vec{\theta}) \rangle.$$

The Hamiltonian H is spectrally decomposed into:

$$(3.12) \quad H = \sum_{i=1}^n \lambda_i |\psi\rangle_i \langle \psi|_i,$$

where λ_i are the eigenvalues and $|\psi\rangle_i$ are the orthogonal eigenstates of H .

The trial state $|\psi(\vec{\theta})\rangle$ is a superposition of these eigenstates:

$$(3.13) \quad |\psi(\vec{\theta})\rangle = \sum_{i=1}^n \alpha_i(\vec{\theta}) |\psi\rangle_i,$$

leading to the expectation:

$$(3.14) \quad E(\vec{\theta}) = \sum_{i=1}^n |\alpha_i(\vec{\theta})|^2 \lambda_i.$$

VQE iteratively adjusts $\vec{\theta}$ to minimize $E(\vec{\theta})$, aiming to approximate λ_{\min} , the lowest eigenvalue of H . This process utilizes a quantum circuit, parameterized by $\vec{\theta}$, and a classical optimizer to find the minimum expectation value, which is critical for solving combinatorial optimization problems.

3.3 Modelling VRP in Quantum

To find a solution to the vehicle routing problem, we can map the cost function to an Ising Hamiltonian H_c [45]. The minimization of Ising Hamiltonian H_c gives the solution to the problem. To begin, let us consider an arbitrary connected graph of n vertices and a binary decision variable x_{ij} who has a value 1 if there exists an edge between i and j for edge weight $w_{ij} > 0$ else; the value is 0. To represent the VRP problem, we need $n \times (n - 1)$ decision variables. For every edge from $i \rightarrow j$, we define two sets of nodes $source[i]$ and $target[j]$. The set $source[i]$ contains the nodes j to which i sends an edge $j \in source[i]$. The set $target[j]$ contains the nodes i to which i sends an edge $i \in target[j]$. We define VRP as follows [27, 47],

$$(3.15) \quad VRP(n, k) = \min_{\{x_{ij}\}_{i \rightarrow j} \in \{0,1\}} \sum_{i \rightarrow j} w_{ij} x_{ij},$$

where k is the number of vehicles and n is the total number of locations. Considering the starting location as 0^{th} location or Depot D , we have $n - 1$ locations for vehicles to travel. This is subject to the following constraints,

$$(3.16) \quad \begin{aligned} \sum_{j \in source [i]} x_{ij} &= 1, \forall i \in \{1, \dots, n-1\}, \\ \sum_{j \in target [i]} x_{ji} &= 1, \forall i \in \{1, \dots, n-1\}, \\ \sum_{j \in source [0]} x_{0j} &= k, \\ \sum_{j \in target [0]} x_{j0} &= k \\ u_i - u_j + Qx_{ij} &\leq Q - q_j, \forall i \sim j, i, j \neq 0, \\ q_i \leq u_i &\leq Q, \forall i, i \neq 0. \end{aligned}$$

The first two constraints impose the restriction that the delivering vehicle must visit each node only once. The middle two constraints enforce the restriction that the vehicle must return to the depot after delivering the goods. The last two constraints impose the sub-tour elimination conditions and are bound on u_i , with $Q > q_j > 0$, and $u_i, Q, q_i \in \mathbb{R}$.

For the VRP equation, we can form the Hamiltonian of VRP as follows [27],

$$\begin{aligned}
 H_{VRP} &= H_A + H_B + H_C + H_D + H_E, \\
 H_A &= \sum_{i \rightarrow j} w_{ij} x_{ij}, \\
 H_B &= A \sum_{i \in \{1, \dots, n-1\}} \left(1 - \sum_{j \in \text{source}[i]} x_{ij} \right)^2, \\
 H_C &= A \sum_{i \in \{1, \dots, n-1\}} \left(1 - \sum_{j \in \text{target}[i]} x_{ji} \right)^2, \\
 H_D &= A \left(k - \sum_{j \in \text{source}[0]} x_{0j} \right)^2, \\
 H_E &= A \left(k - \sum_{j \in \text{target}[0]} x_{j0} \right)^2,
 \end{aligned}
 \tag{3.17}$$

where $A > 0$ is a constant.

The set of all binary decision variables x_{ij} can be represented in vector form as,

$$\vec{\mathbf{x}} = [x_{(0,1)}, x_{(0,2)}, \dots, x_{(1,0)}, x_{(1,2)}, \dots, x_{(n-1, n-2)}]^T.
 \tag{3.18}$$

Using the above vector, we can define two additional vectors for each node,

$$\begin{aligned}
 \vec{z}_{S[i]} &= \vec{\mathbf{x}} \ni x_{ij} = 1, x_{kj} = 0, k \neq i, \forall j, k \in \{0, \dots, n-1\}, \\
 \vec{z}_{T[i]} &= \vec{\mathbf{x}} \ni x_{ji} = 1, x_{jk} = 0, k \neq i, \forall j, k \in \{0, \dots, n-1\}.
 \end{aligned}
 \tag{3.19}$$

$$\begin{aligned}
 \sum_{j \in \text{source}[i]} x_{ij} &= \vec{z}_{S[i]}^T \vec{\mathbf{x}}, \\
 \sum_{j \in \text{target}[i]} x_{ji} &= \vec{z}_{T[i]}^T \vec{\mathbf{x}}.
 \end{aligned}
 \tag{3.20}$$

The above vectors will assist in the formulation of the QUBO model of VRP. For a linked graph $G = (N, V)$, the QUBO model [29, 30, 48, 49] is defined as,

$$(3.21) \quad f(x)_{\text{QUBO}} = \min_{x \in \{0,1\}^{(N \times V)}} x^T Q x + g^T x + c ,$$

where Q is a quadratic coefficient of the edge weights, g is a linear coefficient of the node weights, and c is a constant. In order to find these coefficients in the QUBO formations of H_{VRP} given in Eq. 3.17 we first put in Eqs. 3.20 in terms H_B and H_c respectively, then expand and regroup the expression of H_{VRP} according to Eq. 3.21

$$(3.22) \quad \begin{aligned} H &= A \sum_{i=0}^{n-1} \left[z_{S[i]} z_{S[i]}^T + z_{T[i]} z_{T[i]}^T \right] \vec{x}^2 \\ &+ w^T \vec{x} - 2A \sum_{i=1}^{n-1} \left[z_{S[i]}^T + z_{T[i]}^T \right] \vec{x} \\ &- 2Ak \left[z_{S[0]}^T + z_{T[0]}^T \right] \vec{x} + 2A(n-1) + 2Ak^2. \end{aligned}$$

Hence for QUBO formulation of Eq. (3.17) we get the coefficients $Q(n(n-1) \times n(n-1))$, $g(n(n-1) \times 1)$ and c :

$$(3.23) \quad \begin{aligned} Q &= A \left[\left[z_{T[0]}, \dots, z_{T[n-1]} \right]^T \left[z_{T[0]}, \dots, z_{T[n-1]} \right] \right. \\ &\quad \left. + (\mathbb{I}_n \otimes \mathbb{J}(n-1, n-1)) \right], \\ g &= W - 2Ak \left((e_0 \otimes \mathbb{J}_{n-1}) + \left[z_{T[0]} \right]^T \right), \\ &\quad + 2A (\mathbb{J}_n \otimes \mathbb{J}_{n-1}), \\ c &= 2A(n-1) + 2Ak^2. \end{aligned}$$

Here, \mathbb{J} is the matrix of all ones, \mathbb{I} is the identity matrix, and $e_0 = [1, 0, \dots, 0]^T$.

The binary decision variable x_{ij} is transformed to spin variable $s_{ij} \in \{-1, 1\}$ as $x_{ij} = (s_{ij} + 1)/2$.

From the above Eqs, we can expand Eq. (3.21) to form the Ising Hamiltonian of VRP [29],

$$(3.24) \quad H_{\text{Ising}} = - \sum_i \sum_{i < j} J_{ij} s_i s_j - \sum_i h_i s_i + d.$$

Here, the terms J_{ij}, h_i and d are defined as follows,

$$(3.25) \quad \begin{aligned} J_{ij} &= -\frac{Q_{ij}}{2}, \forall i < j, \\ h_i &= \frac{g_i}{2} + \sum \frac{Q_{ij}}{4} + \sum \frac{Q_{ji}}{4}, \\ d &= c + \sum_i \frac{g_i}{2} + \sum_i \sum_j \frac{Q_{ij}}{4}. \end{aligned}$$



Figure 3.1: (a) Sample circuit showing gate selections for H_{cost} . (b) Sample circuit showing gate selections with additional U gate after barrier for H_{mixer} . Note: The sample circuits displayed in the figures represent the building blocks of the actual circuit and do not represent actual angles that are obtained as a solution of VRP using VQE.

Note: The sample circuits displayed in the figures represent the building blocks of the actual circuit and do not represent actual angles that are obtained as a solution of VRP using VQE.

3.4 Analysis And Circuit Building

In this section, we create a gate-based circuit to realize the above formulation using the IBM gate model, which we have implemented using the Qiskit framework [50]. For any arbitrary VRP problem using qubits, we begin with the state of $|+\rangle^{\otimes n(n-1)}$ the ground state of H_{mixer} by applying the Hadamard to all qubits initialized as zero states, and we prepare the following state.

$$(3.26) \quad \begin{aligned} |\beta, \gamma\rangle &= e^{-iH_{\text{mixer}}\beta_p} e^{-iH_{\text{cost}}\gamma_p} \dots \\ &\dots e^{-iH_{\text{mixer}}\beta_0} e^{-iH_{\text{cost}}\gamma_0} |+\rangle^{n(n-1)}. \end{aligned}$$

The energy E of the state $|\beta, \gamma\rangle$ is calculated by the expectation of H_{cost} from Eq. (??). Once again, the H_{cost} term may be expressed in terms of Pauli operators using the Ising model, as

$$(3.27) \quad H_{cost} = -\sum_i \sum_{i < j} J_{ij} \sigma_i^z \sigma_j^z - \sum_i h_i \sigma_i^z - d.$$

Thus for a single term of state in $|\beta, \gamma\rangle$ as β_0, γ_0 , the expression reads, $e^{-iH_{mixer}\beta_0} e^{-iH_{cost}\gamma_0}$. The first term H_{cost} can be expanded to following,

$$(3.28) \quad \begin{aligned} e^{iJ_{ij}\gamma_0\sigma_i\sigma_j} &= \cos J_{ij}\gamma_0 I + i \sin J_{ij}\gamma_0 \sigma_i \sigma_j, \\ &= \begin{bmatrix} e^{iJ_{ij}\gamma_0} & 0 & 0 & 0 \\ 0 & e^{-iJ_{ij}\gamma_0} & 0 & 0 \\ 0 & 0 & e^{-iJ_{ij}\gamma_0} & 0 \\ 0 & 0 & 0 & e^{iJ_{ij}\gamma_0} \end{bmatrix}, \\ &= M. \end{aligned}$$

Applying $CNOT$ gate on, before, and after the above matrix ' M ' we can swap the diagonal elements,

$$(3.29) \quad CNOT(M)CNOT = \begin{bmatrix} e^{iJ_{ij}\gamma_0} & 0 & 0 & 0 \\ 0 & e^{-iJ_{ij}\gamma_0} & 0 & 0 \\ 0 & 0 & e^{iJ_{ij}\gamma_0} & 0 \\ 0 & 0 & 0 & e^{-iJ_{ij}\gamma_0} \end{bmatrix}.$$

Observing the upper and lower blocks of the matrix, we can rewrite,

$$(3.30) \quad \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \otimes \begin{bmatrix} e^{iJ_{ij}\gamma_0} & 0 \\ 0 & e^{-iJ_{ij}\gamma_0} \end{bmatrix} = I \otimes e^{iJ_{ij}\gamma_0} \begin{bmatrix} 1 & 0 \\ 0 & e^{-2iJ_{ij}\gamma_0} \end{bmatrix}.$$

$\begin{bmatrix} 1 & 0 \\ 0 & e^{-2iJ_{ij}\gamma_0} \end{bmatrix}$ is a phase gate. Looking at the 2nd term of H_{cost} we get,

$$\begin{aligned}
 H_{cost} &= \sum_i h_i \sigma_i^z, \\
 e^{ih_i \gamma_0 \sigma_i^z} &= \cos h_i \gamma_0 I + i \sin h_i \gamma_0 \sigma_i^z, \\
 &= \cos h_i \gamma_0 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + i \sin h_i \gamma_0 \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \\
 &= \begin{bmatrix} e^{ih_i \gamma_0} & 0 \\ 0 & e^{-ih_i \gamma_0} \end{bmatrix}, \\
 (3.31) \quad &= e^{ih_i \gamma_0} \begin{bmatrix} 1 & 0 \\ 0 & e^{-2ih_i \gamma_0} \end{bmatrix}.
 \end{aligned}$$

Fig. 3.1 depicts the basic circuit with two qubits along with gate selections for H_{cost} .

Similarly H_{mixer} can be derived as follows

$$(3.32) \quad H_{mixer} = -\sum_i \sigma_x.$$

Considering a single term of H_{mixer} and taking the unitary,

$$\begin{aligned}
 e^{-iH_{mixer}\beta_0} &= e^{-i(-\sigma_x)\beta_0}, \\
 &= e^{i\sigma_x\beta_0}, \\
 &= \cos \beta_0 \mathbb{1} + i \sin \beta_0 \sigma_x, \\
 (3.33) \quad &= \begin{bmatrix} \cos \beta_0 & i \sin \beta_0 \\ i \sin \beta_0 & \cos \beta_0 \end{bmatrix}.
 \end{aligned}$$

The IBMQ U gate is defined as follows.

$$(3.34) \quad U = \begin{bmatrix} \cos \theta/2 & -e^{i\lambda} \sin \theta/2 \\ e^{i\phi} \sin \theta/2 & e^{i(\lambda+\phi)} \cos \theta/2 \end{bmatrix}.$$

Comparing Eqs. (3.30), (3.31), (3.33) and (3.34), we can establish relation of circuit parameters with γ, β to U gate which will form building blocks of circuit. From Fig. 3.1, the circuit represents H_{cost} term, where the first U gate takes the parameters, $\theta = 0, \phi = -2J_{ij}\gamma_0$, and $\lambda = 0$, and the second U gate takes the parameters, $\theta = 0, \phi = -2h_i\gamma_0$, and $\lambda = 0$. Similarly, in Fig. 3.1(b), the U gate after the barrier represents the H_{mixer} term having the parameters $\theta = 2\beta_0, \phi = \pi/2$, and $\lambda = -\pi/2$.

3.5 VQE Simulation of VRP

We construct the VRP circuit using the above equations and create the Hamiltonians for 3-city and 4-city scenarios. Since we need $n(n - 1)$ qubits, we end up with only Hamiltonians and circuits with 6 (Fig. 3.2) and 12 qubits. Beyond four cities, it is impossible to simulate in a classical desktop computer due to memory limitations. We create the ansatz using a quantum circuit defined in the previous section and run it across various VQE optimizers available in the IBM Qiskit framework: COBYLA, L_BFGS_B, SPSA, and SLSQP. We run the circuit up to 4 layers across all the optimizers and obtain the results depicted in Fig. 3.3 and Fig. 3.4. The primary difference between the Figs. 3.3 and 3.4 is that while the former represents average energy value the later represents minimum energy value of 15 consecutive runs at each layer.

The tables for these figures are presented in Supplementary material section 3.9 with tables 3.6 and 3.7. The figures are derived from 15 consecutive runs of the VRP circuit, each with all the optimizers and four layers. While the average energy values of VRP simulations decrease as optimizations increase across layers for most of the six qubit circuits, the same trend is not observed for 12 qubit circuits. Also, the energy curves vary significantly across all optimizers. Similarly, for minimum energy graphs Fig.3.4, energy values have no clear decreasing trend as optimization layers increase. Yet from the minimum energy graphs, we can reliably say that the protocol has achieved the minimum energy value, However they do not always follow the downward trend or stay at the same level as optimization layers increase. This, of course, is heavily dependent on the optimizer. Thus when selecting an optimizer for simulation, we chose the optimizer that achieves the lowest minimum, the fewest number of optimization layers. In summary, we have found that COBYLA is the best-performing optimizer, followed by SPSA, L_BFGS_B, and SLSQP. However, in the following, when we pass the circuit through various noise models, we will use only the COBYLA optimizer.

CHAPTER 3. ANALYSIS OF THE VEHICLE ROUTING PROBLEM SOLVED VIA HYBRID QUANTUM ALGORITHMS IN THE PRESENCE OF NOISY CHANNELS

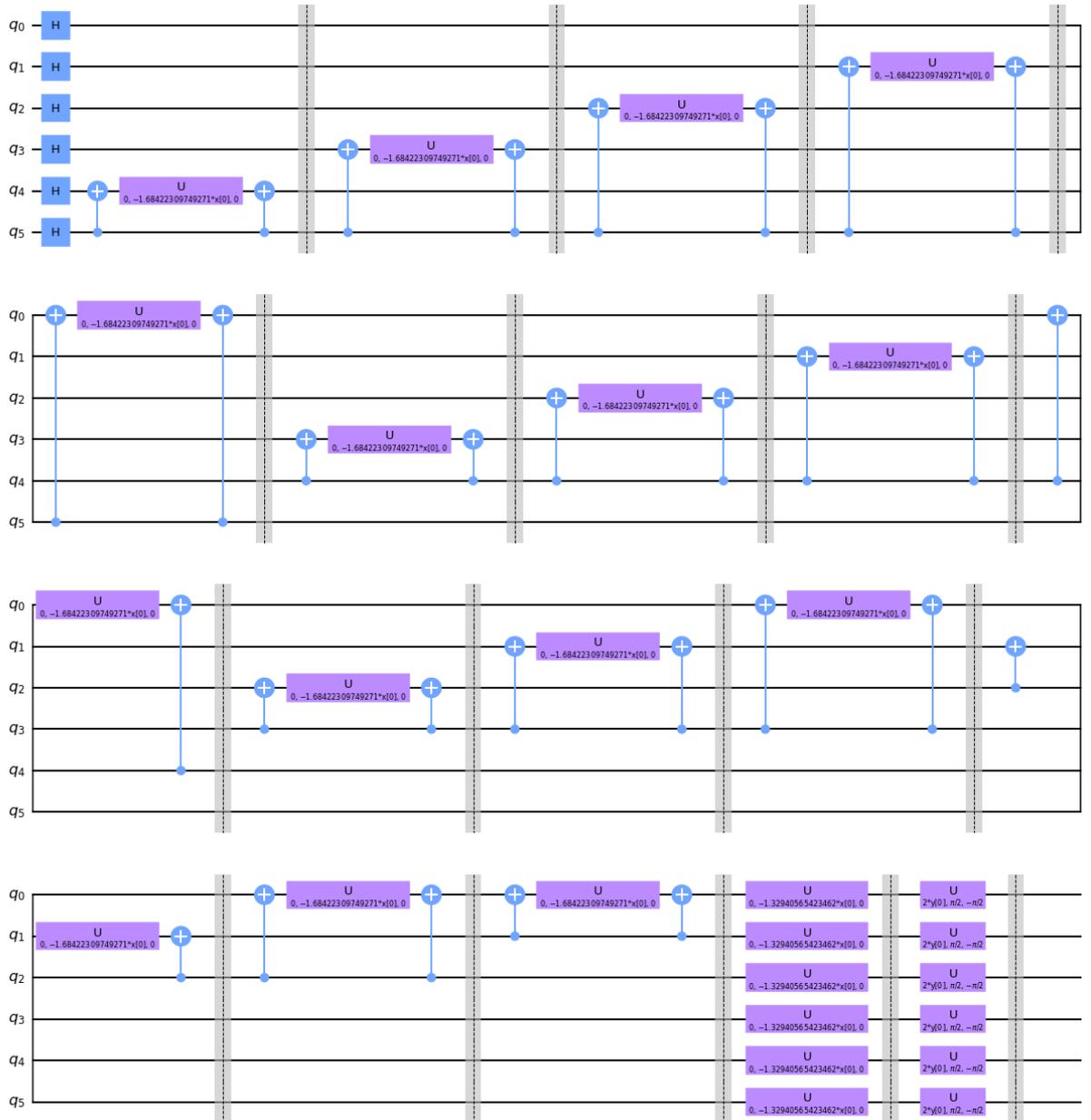


Figure 3.2: VRP Simulation circuit for 6 Qubits

3.5. VQE SIMULATION OF VRP

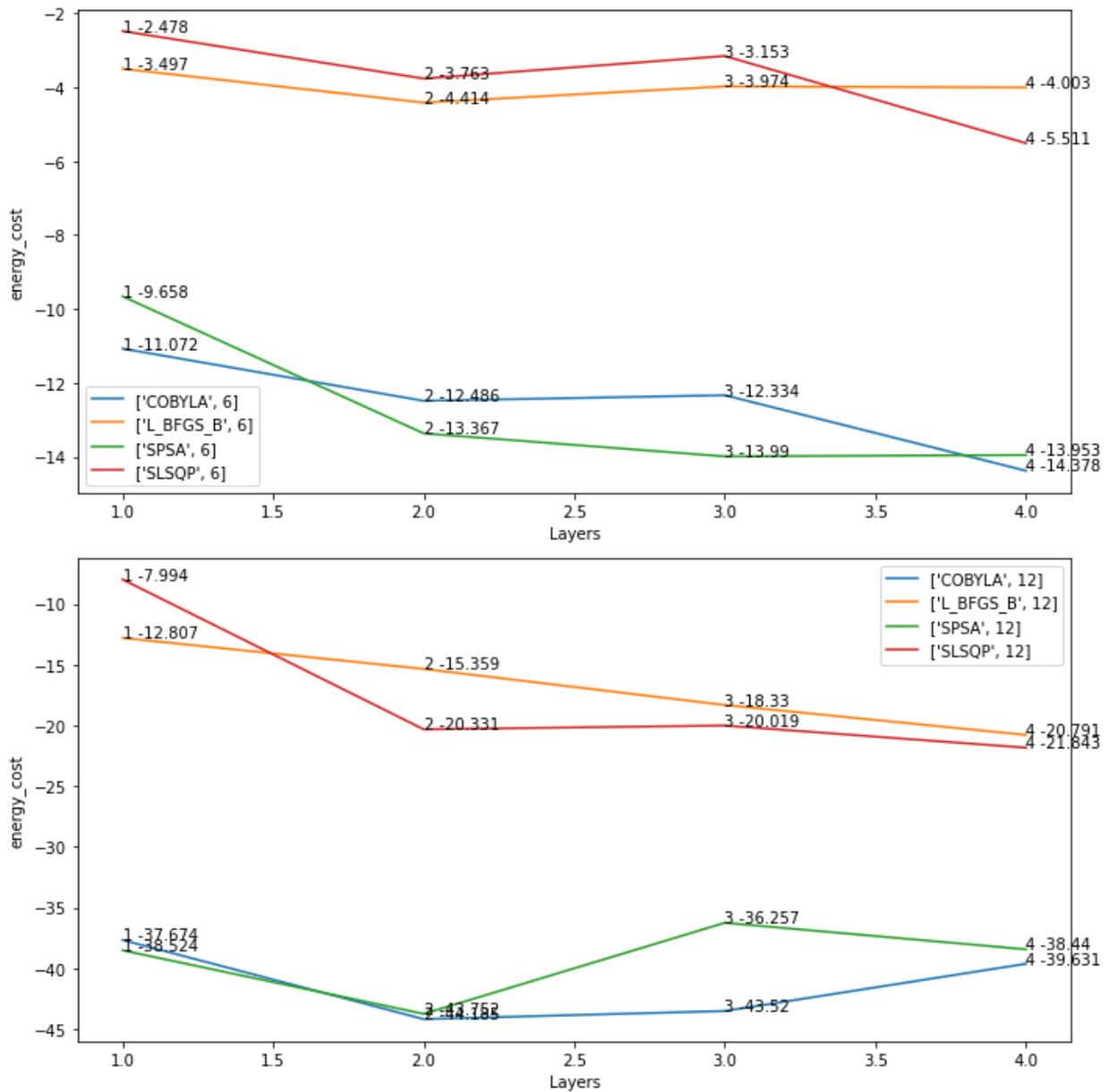


Figure 3.3: Plot illustrating the circuit simulation of VRP with 5 layers using various optimizers (COBYLA, L_BFGS_B, SLSQP, SPSA). The plot consists of two separate graphs depicting the simulation output of 6 qubit and 12 qubit circuits each executed with 1024 shots, respectively. Each plot, in turn, consists of four lines indicating energy values for different optimizers. The average value at each Layer is represented in pairs with (layer, average energy) format.

CHAPTER 3. ANALYSIS OF THE VEHICLE ROUTING PROBLEM SOLVED VIA HYBRID QUANTUM ALGORITHMS IN THE PRESENCE OF NOISY CHANNELS

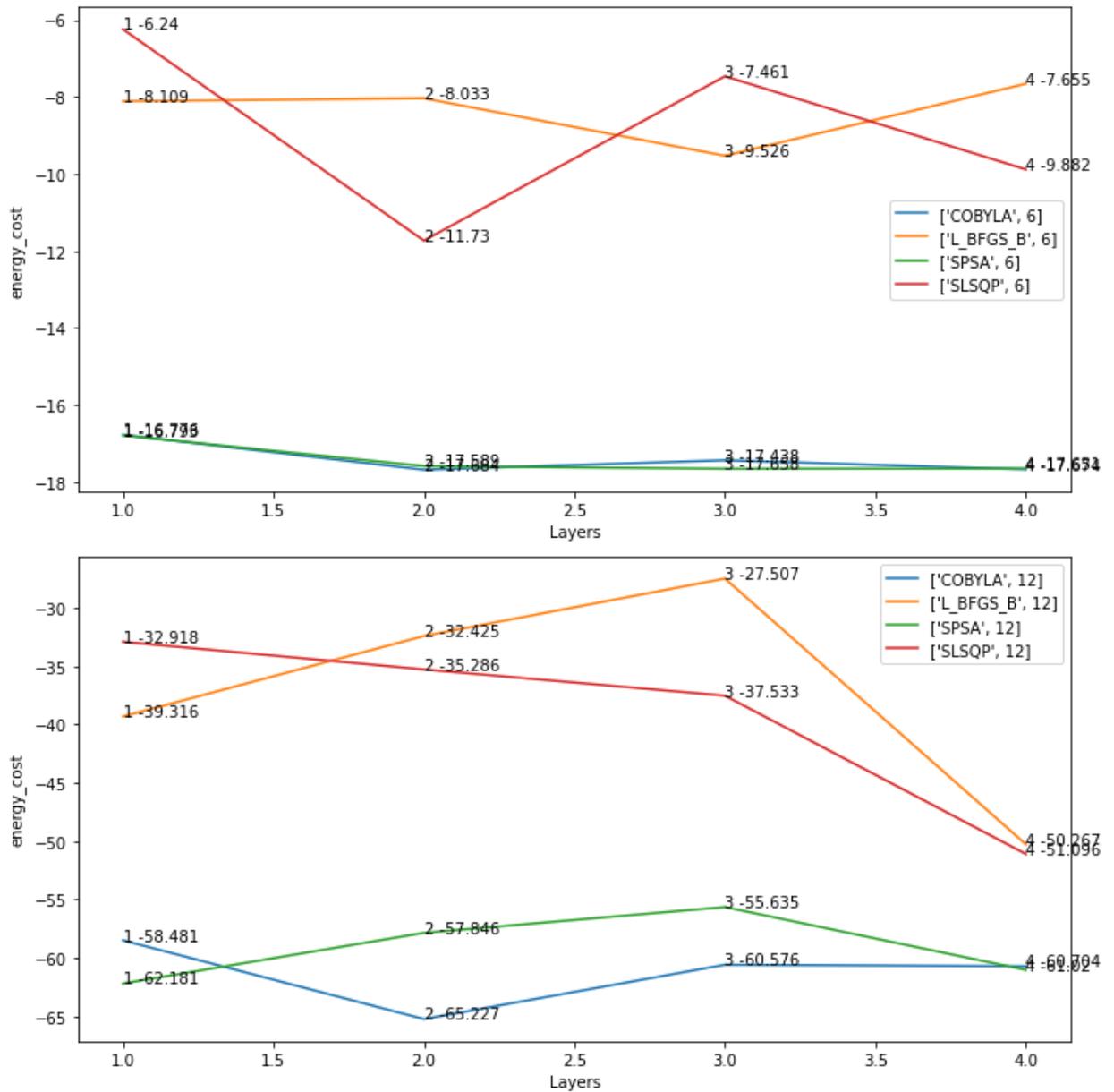


Figure 3.4: Plot illustrating the circuit simulation of VRP with 5 layers using various optimizers (COBYLA, L_BFGS_B, SLSQP, SPSA). The plot consists of two graphs depicting the simulation output of 6 qubit and 12 qubit circuits each executed with 1024 shots, respectively. Each plot consists of four lines indicating energy values for different optimizers. The minimum value at each Layer is represented in pairs with (layer, minimum energy) format.

3.6 Noise Model Simulation of VQE

In a noisy quantum environment, a pure input state will be transformed into a mixed state represented as a density matrix [36, 51]. In the case of a 6-qubit state pure state $|\psi\rangle_{q_0q_1q_2q_3q_4q_5}$ the density matrix can be defined as $\rho = |\psi\rangle_{q_0q_1q_2q_3q_4q_5} \langle\psi|_{q_0q_1q_2q_3q_4q_5}$. After the implementation of the noise model, the density matrix takes the following form,

$$(3.35) \quad \xi_r(\rho) = \sum_m (E_m^{rq0})(E_m^{rq1}) \dots (E_m^{rq5}) \rho \times (E_m^{xq0})^\dagger (E_m^{xq1})^\dagger \dots (E_m^{xq5})^\dagger,$$

where $r \in \{A, B, W, F, D\}$. The elements of the noise channels are described as follows, A is amplitude damping noise, B is bit-flip noise, W is phase-flip noise, F is bit-phase-flip noise, and D is depolarising noise. We apply these noise channels to our VRP circuit and ansatz, which is variable based on the number of qubits (6 or 12) and layers (1 to 5). For simulation purposes, we choose the optimizer COBYLA as it has the best performance characteristics in the simulation of VQE. We restrict the noise probability to 0.5 as noisy environments beyond this noise level are unlikely and irrelevant in practice. The following subsections discuss the noise channels and operators we experimented on in the VRP circuit.

3.6.1 Amplitude Damping

The energy dissipation is a consequence of the interaction of the quantum system with an amplitude-damping channel. A quantum system gaining or losing energy from or to its environment is described as a change in amplitude rather than phase [51, 52]. If κ_A is the probability of gain or loss of amplitude or decoherence rate, the Kraus operators of amplitude damping channel can be described as follows,

$$(3.36) \quad \begin{aligned} E_0^A &= \begin{bmatrix} 1 & 0 \\ 0 & \sqrt{1-\kappa_A} \end{bmatrix}, \\ E_1^A &= \sqrt{\kappa_A} \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}. \end{aligned}$$

3.6.2 Bit-Flip Noise

Random bit-flip errors characterize bit-flip noise [52] with probability κ_B . Thus, the Kraus operators of the bit-flip noise channel can be described as,

$$(3.37) \quad \begin{aligned} E_0^B &= \sqrt{1-\kappa_B} I, \\ E_1^B &= \sqrt{\kappa_B} X = \sqrt{\kappa_B} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}. \end{aligned}$$

3.6.3 Phase Flip Noise

Phase-flip noise alters the phase parameter of the quantum system without exchange of energy [51, 52]. The decoherence rate or the phase-flip noise parameter also follows the simple Bernoulli distribution with probability parameter κ_W . thus the Kraus operators of phase-flip noise channel can be defined as Eq. (3.38),

$$(3.38) \quad \begin{aligned} E_0^W &= \sqrt{1-\kappa_W} I, \\ E_1^W &= \sqrt{\kappa_W} Z = \sqrt{\kappa_W} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}. \end{aligned}$$

3.6.4 Bit-Phase Flip Noise

Bit-phase flip noise channel is characterized by a combination of random bit-flip errors and a change in the quantum system's phase information without energy loss [52]. Like other noise channels, the decoherence rate or the combined probability of bit-phase flip error follows the distribution κ_F . The Kraus operator of the bit-phase flip channel could be given by,

$$(3.39) \quad \begin{aligned} E_0^F &= \sqrt{1-\kappa_F} I, \\ E_1^F &= \sqrt{\kappa_F} Y = \sqrt{\kappa_F} \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}. \end{aligned}$$

3.6.5 Depolarizing Noise

A depolarizing noise channel leaves the system untouched or replaces it with a maximally mixed state of I/d for a d -level quantum system. The decoherence rate or the depolarization noise probability follows the distribution with parameter κ_D . The Kraus operators are as follows,

$$\begin{aligned}
 E_0^D &= \sqrt{1-\kappa_D} I, \\
 E_1^D &= \sqrt{\frac{\kappa_D}{3}} X = \sqrt{\frac{\kappa_D}{3}} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \\
 E_2^D &= \sqrt{\frac{\kappa_D}{3}} Y = \sqrt{\frac{\kappa_D}{3}} \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \\
 E_3^D &= \sqrt{\frac{\kappa_D}{3}} Z = \sqrt{\frac{\kappa_D}{3}} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}.
 \end{aligned}
 \tag{3.40}$$

It is to be noted that, in all cases, the noise channel is applied locally to each qubit in the circuit.

CHAPTER 3. ANALYSIS OF THE VEHICLE ROUTING PROBLEM SOLVED VIA HYBRID QUANTUM ALGORITHMS IN THE PRESENCE OF NOISY CHANNELS

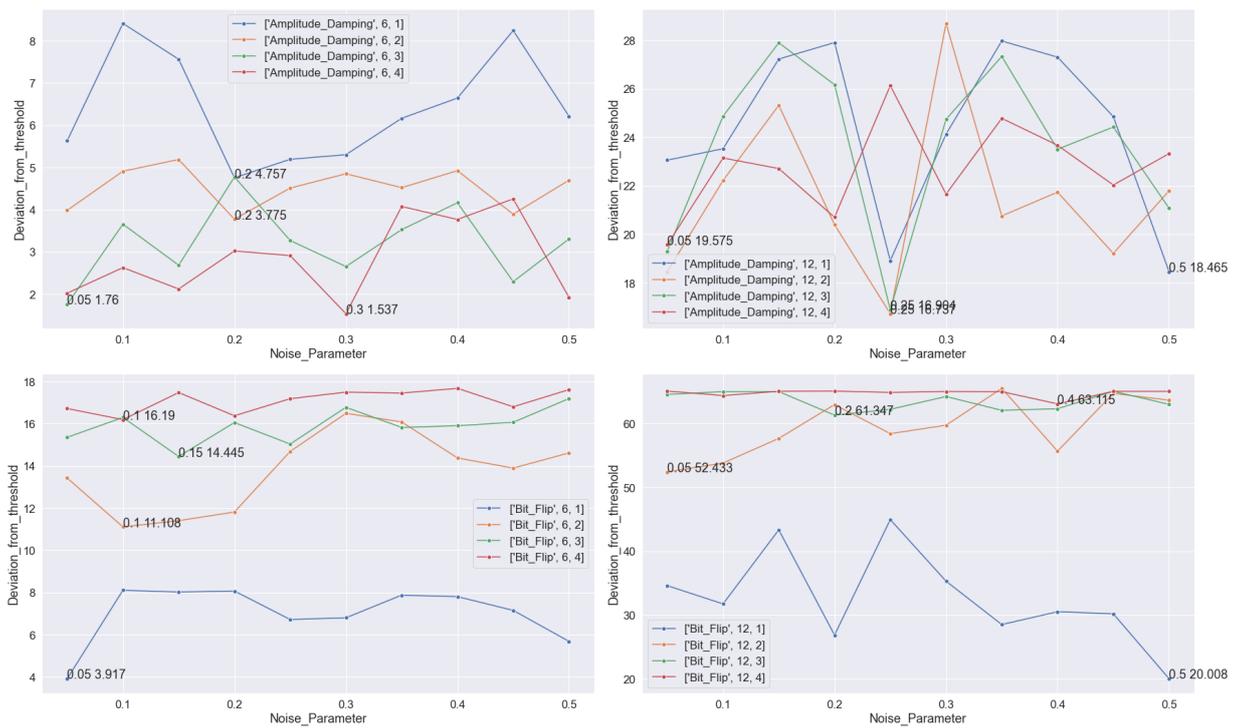


Figure 3.5: Plot illustrating the average Deviation of the energy cost of VRP with 4 layers using various Amplitude damping and Bit-flip noise models. The plot consists of two charts depicting the simulation output of 6 qubit and 12 qubit circuits, respectively. The Average Deviation at each Layer is represented in pairs with (Noise parameter, Average Deviation of Energy cost) format.

3.6. NOISE MODEL SIMULATION OF VQE

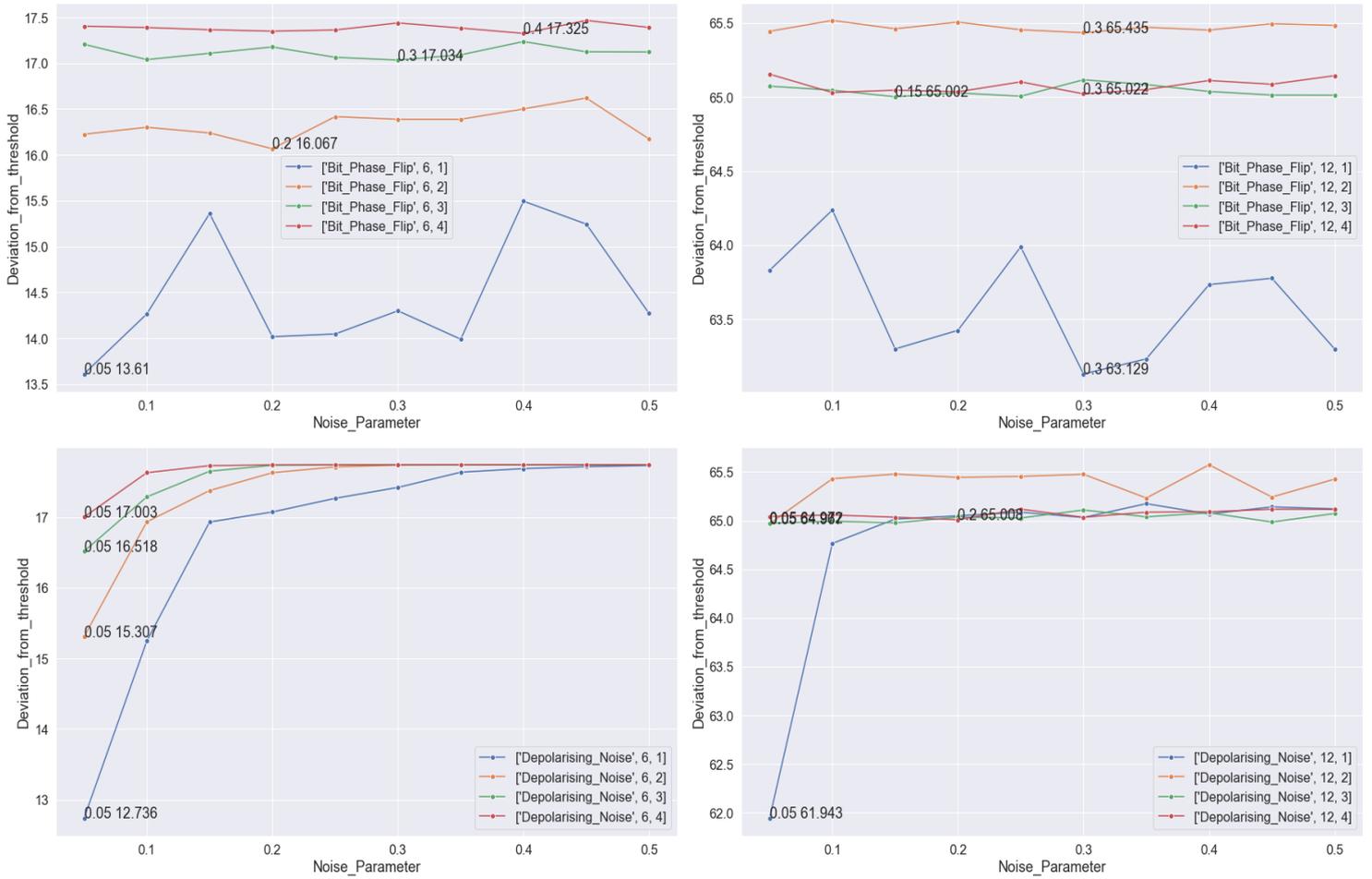


Figure 3.6: Plot illustrating the average Deviation of the energy cost of VQE with 4 layers using Bit-phase-flip, and Depolarising noise. The plot consists of two charts depicting the simulation output of 6 qubit and 12 qubit circuits, respectively. The Average Deviation at each Layer is represented in pairs with (Noise parameter, Average Deviation of Energy cost) format.

CHAPTER 3. ANALYSIS OF THE VEHICLE ROUTING PROBLEM SOLVED VIA HYBRID QUANTUM ALGORITHMS IN THE PRESENCE OF NOISY CHANNELS

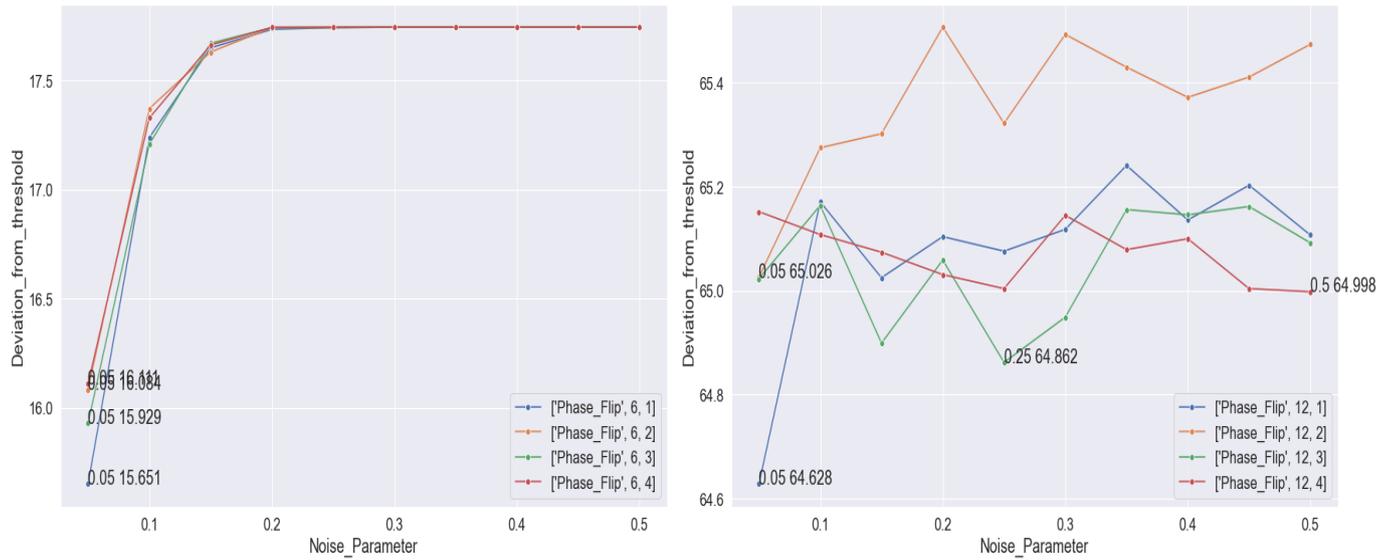


Figure 3.7: Plot illustrating the average energy cost of VRP with 4 layers using the Phase-flip noise model. The plot consists of two charts depicting the simulation output of 6 qubit and 12 qubit circuits, respectively. The average deviation at each Layer is represented in pairs with (Noise parameter, Average Deviation of Energy cost) format.

3.6. NOISE MODEL SIMULATION OF VQE

Qubits	Layers	Noise_Parameter	Noise_Model				
			Amplitude_Damping	Bit_Flip	Bit_Phase_Flip	Depolarising_Noise	Phase_Flip
6	1	0.05	-12.053	-13.768	-4.247	-4.949	-2.033
		0.1	-9.273	-9.577	-2.571	-2.437	-0.447
		0.15	-10.121	-9.661	-3.383	-0.750	-0.036
		0.2	-12.928	-9.614	-5.297	-0.608	0.049
		0.25	-12.491	-10.965	-4.170	-0.416	0.058
		0.3	-12.381	-10.883	-3.761	-0.261	0.059
		0.35	-11.519	-9.809	-4.724	-0.045	0.059
		0.4	-11.040	-9.881	-3.535	0.006	0.059
		0.45	-9.440	-10.531	-2.724	0.034	0.059
	0.5	-11.470	-11.999	-3.118	0.050	0.059	
	2	0.05	-13.698	-4.262	-1.668	-2.377	-1.601
		0.1	-12.773	-6.576	-1.118	-0.748	-0.313
		0.15	-12.495	-6.282	-1.313	-0.305	-0.055
		0.2	-13.909	-5.872	-1.411	-0.052	0.058
		0.25	-13.171	-3.002	-1.621	0.029	0.058
		0.3	-12.834	-1.198	-1.699	0.055	0.059
		0.35	-13.162	-1.603	-1.468	0.058	0.059
		0.4	-12.760	-3.024	-1.304	0.059	0.059
		0.45	-13.789	-3.790	-1.236	0.059	0.059
	0.5	-12.986	-3.075	-1.237	0.059	0.059	
	3	0.05	-15.924	-2.332	-0.551	-1.167	-1.755
		0.1	-14.026	-1.392	-0.605	-0.396	-0.476
		0.15	-14.999	-3.239	-0.539	-0.031	-0.014
		0.2	-12.909	-1.634	-0.492	0.053	0.058
		0.25	-14.413	-2.651	-0.407	0.058	0.059
		0.3	-15.029	-0.910	-0.600	0.059	0.059
		0.35	-14.246	-1.864	-0.723	0.059	0.059
		0.4	-13.519	-1.784	-0.574	0.059	0.059
		0.45	-15.388	-1.621	-0.620	0.059	0.059
	0.5	-14.377	-0.498	-0.693	0.059	0.059	
	4	0.05	-15.659	-0.969	-0.411	-0.681	-1.574
		0.1	-15.055	-1.494	-0.446	-0.052	-0.357
		0.15	-15.560	-0.213	-0.343	0.047	-0.022
		0.2	-14.661	-1.312	-0.101	0.059	0.059
		0.25	-14.771	-0.502	-0.219	0.059	0.059
		0.3	-16.147	-0.193	-0.368	0.059	0.059
		0.35	-13.608	-0.238	-0.440	0.059	0.059
		0.4	-13.916	-0.012	-0.439	0.059	0.059
		0.45	-13.427	-0.886	-0.241	0.059	0.059
	0.5	-15.757	-0.074	-0.221	0.059	0.059	

Table 3.1: VQE simulation of amplitude damping, bit-flip, phase-flip, bit-phase-flip, and depolarizing channel for 6 qubits with 4 layers involving optimizer COBYLA, where energy costs are averaged over 10 simulations.

CHAPTER 3. ANALYSIS OF THE VEHICLE ROUTING PROBLEM SOLVED VIA HYBRID QUANTUM ALGORITHMS IN THE PRESENCE OF NOISY CHANNELS

Qubits	Layers	Noise_Parameter	Noise_Model				
			Amplitude_Damping	Bit_Flip	Bit_Phase_Flip	Depolarising_Noise	Phase_Flip
12	1	0.05	-42.626	-31.086	-1.522	-3.741	-1.056
		0.1	-42.157	-33.947	-1.743	-0.919	-0.514
		0.15	-38.465	-22.328	-2.430	-0.669	-0.660
		0.2	-37.787	-38.835	-3.391	-0.637	-0.581
		0.25	-46.766	-20.722	-2.142	-0.599	-0.609
		0.3	-41.555	-30.384	-0.982	-0.653	-0.567
		0.35	-37.722	-37.187	-2.105	-0.512	-0.444
		0.4	-38.388	-35.172	-2.454	-0.618	-0.548
		0.45	-40.835	-35.500	-3.122	-0.545	-0.482
	0.5	-47.219	-45.676	-1.586	-0.565	-0.577	
	2	0.05	-47.215	-13.251	-0.416	-0.718	-0.658
		0.1	-43.464	-11.857	-0.419	-0.255	-0.410
		0.15	-40.368	-7.993	-0.299	-0.208	-0.383
		0.2	-45.283	-2.685	-0.417	-0.242	-0.178
		0.25	-48.947	-7.242	-0.359	-0.231	-0.363
		0.3	-36.422	-5.921	-0.372	-0.209	-0.192
		0.35	-44.922	-0.137	-0.431	-0.454	-0.255
		0.4	-43.944	-9.986	-0.400	-0.113	-0.313
		0.45	-46.463	-0.931	-0.556	-0.444	-0.274
	0.5	-43.890	-1.993	-0.434	-0.259	-0.211	
	3	0.05	-46.368	-1.086	-0.409	-0.713	-0.662
		0.1	-40.817	-0.656	-0.461	-0.690	-0.520
		0.15	-37.797	-0.649	-0.493	-0.709	-0.786
		0.2	-39.519	-4.338	-0.465	-0.649	-0.626
		0.25	-48.781	-3.428	-0.402	-0.658	-0.823
		0.3	-40.945	-1.416	-0.349	-0.576	-0.736
		0.35	-38.358	-3.589	-0.288	-0.646	-0.529
		0.4	-42.179	-3.328	-0.447	-0.605	-0.539
		0.45	-41.264	-0.603	-0.585	-0.700	-0.523
	0.5	-44.605	-2.647	-0.345	-0.613	-0.593	
	4	0.05	-46.110	-0.565	-0.509	-0.643	-0.533
		0.1	-42.539	-1.276	-0.630	-0.628	-0.577
		0.15	-42.971	-0.596	-0.616	-0.650	-0.611
		0.2	-44.977	-0.560	-0.553	-0.676	-0.653
		0.25	-39.561	-0.781	-0.508	-0.567	-0.680
		0.3	-44.023	-0.624	-0.531	-0.651	-0.539
		0.35	-40.907	-0.657	-0.507	-0.599	-0.606
		0.4	-42.012	-2.570	-0.597	-0.594	-0.585
		0.45	-43.650	-0.604	-0.736	-0.569	-0.681
	0.5	-42.357	-0.608	-0.608	-0.571	-0.686	

Table 3.2: VQE simulation of amplitude damping, bit-flip, phase-flip, bit-phase-flip, and depolarizing channel for 12 qubits with 4 layers involving optimizer COBYLA, where energy costs are averaged over 10 simulations.

3.6. NOISE MODEL SIMULATION OF VQE

Qubits	Layers	Noise_Parameter	Noise_Model				
			Amplitude_Damping	Bit_Flip	Bit_Phase_Flip	Depolarising_Noise	Phase_Flip
6	1	0.05	5.631	3.917	13.437	12.736	15.651
		0.1	8.412	8.107	15.113	15.247	17.238
		0.15	7.563	8.023	14.301	16.934	17.649
		0.2	4.757	8.070	12.388	17.076	17.734
		0.25	5.194	6.719	13.514	17.269	17.742
		0.3	5.303	6.801	13.923	17.423	17.744
		0.35	6.165	7.875	12.961	17.639	17.744
		0.4	6.644	7.803	14.149	17.690	17.744
		0.45	8.244	7.153	14.960	17.718	17.744
		0.5	6.215	5.686	14.567	17.734	17.744
	2	0.05	3.986	13.423	16.017	15.307	16.084
		0.1	4.911	11.108	16.567	16.936	17.371
		0.15	5.189	11.402	16.372	17.379	17.630
		0.2	3.775	11.813	16.273	17.632	17.742
		0.25	4.513	14.683	16.063	17.714	17.742
		0.3	4.851	16.487	15.985	17.739	17.744
		0.35	4.522	16.081	16.217	17.422	17.744
		0.4	4.925	14.661	16.380	17.744	17.744
		0.45	3.895	13.894	16.449	17.744	17.744
		0.5	4.698	14.610	16.447	17.744	17.744
	3	0.05	1.760	15.352	17.133	16.518	15.929
		0.1	3.658	16.292	17.080	17.289	17.208
		0.15	2.686	14.445	17.146	17.653	17.670
		0.2	4.776	16.051	17.192	17.737	17.742
		0.25	3.271	15.033	17.278	17.742	17.743
		0.3	2.655	16.774	17.084	17.744	17.744
		0.35	3.438	15.820	16.961	17.744	17.744
		0.4	4.165	15.901	17.110	17.744	17.744
		0.45	2.297	16.063	17.065	17.744	17.744
		0.5	3.307	17.187	16.991	17.744	17.744
	4	0.05	2.025	16.715	17.273	17.003	16.111
		0.1	2.629	16.190	17.238	17.632	17.328
		0.15	2.124	17.472	17.342	17.732	17.662
		0.2	3.023	16.373	17.583	17.743	17.743
		0.25	2.914	17.182	17.465	17.744	17.744
		0.3	1.537	17.492	17.316	17.744	17.744
		0.35	4.076	17.447	17.244	17.744	17.744
		0.4	3.769	17.672	17.246	17.744	17.744
		0.45	4.257	16.798	17.443	17.744	17.744
		0.5	1.927	17.610	17.463	17.744	17.744

Table 3.3: For 6 qubits with 4 layers and using COBYLA as an optimizer, the table above shows the average deviation from the classical minimum (over 10 simulations) for VQE simulations utilizing amplitude damping, bit-flip, phase-flip, bit-phase-flip, and depolarising channels.

CHAPTER 3. ANALYSIS OF THE VEHICLE ROUTING PROBLEM SOLVED VIA HYBRID QUANTUM ALGORITHMS IN THE PRESENCE OF NOISY CHANNELS

Qubits	Layers	Noise_Parameter	Noise_Model				
			Amplitude_Damping	Bit_Flip	Bit_Phase_Flip	Depolarising_Noise	Phase_Flip
12	1	0.05	23.058	34.599	64.163	61.943	64.628
		0.1	23.528	31.738	63.942	64.765	65.171
		0.15	27.220	43.356	63.254	65.015	65.025
		0.2	27.898	26.850	62.294	65.048	65.104
		0.25	18.919	44.963	63.543	65.086	65.076
		0.3	24.129	35.301	64.703	65.031	65.118
		0.35	27.963	28.498	63.579	65.173	65.241
		0.4	27.297	30.513	63.231	65.067	65.136
		0.45	24.850	30.185	62.563	65.140	65.203
	0.5	18.465	20.008	64.099	65.120	65.107	
	2	0.05	18.469	52.433	65.268	64.967	65.026
		0.1	22.220	53.827	65.266	65.429	65.275
		0.15	25.317	57.691	65.385	65.477	65.302
		0.2	20.402	62.999	65.268	65.443	65.507
		0.25	16.737	58.443	65.325	65.453	65.322
		0.3	29.263	59.764	65.313	65.475	65.493
		0.35	20.763	65.548	65.254	65.231	65.430
		0.4	21.740	55.698	65.285	65.572	65.372
		0.45	19.222	64.753	65.129	65.241	65.411
	0.5	21.795	63.692	65.251	65.425	65.474	
	3	0.05	19.317	64.599	65.275	64.972	65.023
		0.1	24.868	65.029	65.224	64.994	65.164
		0.15	27.888	65.035	65.191	64.976	64.899
		0.2	26.165	61.347	65.220	65.036	65.059
		0.25	16.904	62.256	65.283	65.026	64.862
		0.3	24.740	64.269	65.335	65.108	64.949
		0.35	27.327	62.096	65.397	65.038	65.156
		0.4	23.506	62.356	65.238	65.079	65.146
		0.45	24.421	65.082	65.100	64.985	65.162
	0.5	21.079	63.038	65.339	65.072	65.092	
	4	0.05	19.575	65.120	65.175	65.042	65.152
		0.1	23.145	64.408	65.055	65.057	65.108
		0.15	22.713	65.088	65.068	65.034	65.074
		0.2	20.707	65.124	65.132	65.008	65.031
		0.25	26.124	64.903	65.177	65.117	65.004
		0.3	21.662	65.061	65.153	65.033	65.145
		0.35	24.778	65.028	65.178	65.086	65.079
		0.4	23.672	63.115	65.088	65.090	65.100
		0.45	22.034	65.080	64.949	65.116	65.004
	0.5	23.328	65.076	65.077	65.114	64.998	

Table 3.4: For 12 qubits with 4 layers and using COBYLA as an optimizer, the table above shows the average deviation from the classical minimum (over 10 simulations) for VQE simulations utilizing amplitude damping, bit-flip, phase-flip, bit-phase-flip, and depolarising channels.

3.7 Inferences from Simulation

In the experiment of simulating VRP across various noise channels, we vary the noise probability from 0.05 to 0.5 and observe the energy values of VQE. We execute the VRP circuit with 1 to 4 layers on our chosen optimizer COBYLA for both 6 qubit and 12 qubit configurations. The experiment is repeated ten times for each noise model with different noise realizations. In the same experiment, we calculate the

minimum eigenvalue of classical Hamiltonian and record the difference in energy cost after noise induction. The state's energy is recorded for each layer from 1 to 4 of the QAOA circuit. These values are averaged over the ten iterations to arrive at the average energy cost for each value of the noise parameter and each layer number. The results are shown in Table 3.1 and Table 3.2. The deviation from the optimal value is shown in Table 3.3 and Table 3.4. Finally, table 3.5 summarizes the inferences on deviation from the classical minimum for various noise models. We observe that VQE results are impacted due to the induction of noise. In the below subsections, we will describe our observations briefly for each noise model.

3.7.1 Amplitude damping Noise

Amplitude damping noise shows values range between 50% to 75% of classical minimum for both 6 qubit (-17.68) and 12 qubit (-65.684) circuits. There are a few outliers where the algorithm can reach very close to the classical minimum for 6 qubit circuits. For 12 qubit circuits, the values are mostly above 50% of classical minimum but never reach classical minimum as close as in 6 qubit circuits. This trend is seen across multiple layers for amplitude damping channels. It is noticed that the global minimum across layers is observed at the 2nd layer at -48.947 , but it is very close to the minimum of the 1st layer at -47.219 . Hence we can infer that increasing layers does not necessarily improve the results for the amplitude-damping channel. Table 3.8 summarizes the amplitude damping average energy values. Figure 3.5 refers to the average deviation of energy cost from the classical minimum at each noise parameter across layers. This again confirms that deviation from classical minimum energy cost due to amplitude damping noise remains within 25% to 50%, which is recorded in the table 3.5.

3.7.2 Bit-Flip Noise

For the bit-flip noise channel, we note that the VQE values are 50% or above the classical minimum for the first layer, but it degrades to around 25% for 2nd layer, falling further on 3rd and finally close to zero on 4th layer. Since this trend is seen in both 6 qubit and 12 qubit circuits, we can infer that increasing the number of layers degrades the VQE values for the bit-flip noise channel. We have summarized

the bit-flip noise channel average energy values in the table 3.10. Figure 3.5 refers to the average deviation of energy cost from the classical minimum at each noise parameter across layers. This again confirms that deviation from classical minimum energy cost due to bit-flip noise remains within the range of 50% to 75% for the first two Layers before deteriorating further, which is recorded in the table 3.5.

3.7.3 Bit-Phase-Flip Noise

There is a similar observation for the bit-phase-flip channel. The VQE values are 25% (or above) of the classical minimum for the first layer, but they degrade as the layers increase for 6 qubit. For 12 qubit circuits, the VQE values are consistently poor. We have summarized bit-phase-flip noise channel average energy values in the table 3.14. Figure 3.6 refers to the average deviation of energy cost from the classical minimum at each noise parameter across layers. This again confirms that deviation from classical minimum energy cost due to bit-phase-flip noise remains close to 100%; this is recorded in the table 3.5.

3.7.4 Phase-Flip and Depolarizing Noise Channel

Finally, for both depolarizing and phase-flip channels, the VQE values remain close to zero for both 6 qubit and 12 qubit circuits. It appears that phase-flip and depolarizing noise channels are the most detrimental in VQE circuits (Supplementary Tables 3.12 and 3.16). The figures 3.6 and 3.7 refer to the average deviation of energy cost from the classical minimum at each noise parameter across layers. This again confirms that deviation from classical minimum energy cost due to depolarising and phase-flip noise remains close to 100%, recorded in the table 3.5.

3.7.5 Data gathering and Statistics Collection

In all the simulations, we have used a quantum instance object and a fixed random seed in the Qiskit framework to avoid VQE terminating early and mitigate statistical fluctuations. Hence, all the noise models used here are applied to the quantum instance object, which in turn applies noise to the circuit whose parameters are

varied by VQE to arrive at a result. We have executed ten iterations of VQE using various noise channels described above. From the results of the ten simulations, we have taken the average energy value of each noise parameter at each layer. Our figure of merit is the difference between the layer’s classical minimum and average energy cost. We remind the reader that gate-based simulations are extremely expensive. The results reported here amounted to 219 hours of CPU time on a standard laptop computer using Qiskit’s built-in simulators [50]. While more iterations would improve the variability of the average energy calculations, some clear trends have already been observed.

Noise Channel	Qubits	Classical Minimum	1st Layer Dev. %	2nd Layer Dev.%	3rd Layer Dev. %	4th Layer Dev.%
Amplitude Damping	6	-17.68	25% - 50%	25% - 50%	25% - 50%	25% - 50%
Bit Flip	6	-17.68	50%	75%	nearly 100%	nearly 100%
BitPhase Flip	6	-17.68	nearly 100%	nearly 100%	nearly 100%	nearly 100%
Phase Flip	6	-17.68	nearly 100%	nearly 100%	nearly 100%	nearly 100%
Depolarising	6	-17.68	75%	nearly 100%	nearly 100%	nearly 100%

Noise Channel	Qubits	Classical Minimum	1st Layer Dev. %	2nd Layer Dev.%	3rd Layer Dev. %	4th Layer Dev.%
Amplitude Damping	12	-65.684	25% - 50%	25% - 50%	25% - 50%	25% - 50%
Bit Flip	12	-65.684	50%	75%	nearly 100%	nearly 100%
BitPhase Flip	12	-65.684	nearly 100%	nearly 100%	nearly 100%	nearly 100%
Phase Flip	12	-65.684	nearly 100%	nearly 100%	nearly 100%	nearly 100%
Depolarising	12	-65.684	nearly 100%	nearly 100%	nearly 100%	nearly 100%

Table 3.5: Table summarizing the inferences on deviation from the classical minimum for VQE simulation VRP using amplitude damping, bit-flip, phase-flip, bit-phase-flip, and depolarizing channels for 6 and 12 qubits.

3.7.6 Choice Of COBYLA

The COBYLA optimizer, a gradient-free method, is particularly effective in quantum computing scenarios where derivatives are not available or reliable. It operates by linearly approximating both the objective and constraint functions within a trust-region framework, which allows it to handle noisy optimization environments effectively [53]. Unlike gradient-based methods like L-BFGS [54] and SLSQP [55], COBYLA does not require derivative information, reducing the risk of error amplification due to noisy data. This characteristic makes it highly suitable for quantum variational algorithms, such as those used in the vehicle routing problem,

where noise is a significant challenge. In contrast, methods like the Simultaneous Perturbation Stochastic Approximation (SPSA) optimizer also operate without gradients but might require more tuning and can show slower convergence in certain conditions [56]. Overall, COBYLA’s ability to efficiently manage constraints and adaptively refine its optimization strategy in the absence of accurate gradient information makes it a superior choice for handling the complexities of quantum noise [57].

3.8 Discussion and Conclusion

Detailed simulation results, first for a noiseless case (Figs. 3.3 and 3.4) and then for each noise channel, are provided in the new section 3.9. Each noise channel’s simulation results are divided into two tables: an average and a minimum. These tables make it easy to see that the deviation from the classical minimum does not change linearly when the noise parameter is changed from 0.05 to 0.5 for each given noise channel. It is also clear that the deviation from the classical minimum does not grow linearly with the number of circuit layers used for optimization. Consider the instance of a bit-flip noise channel with a noise parameter of 0.25; for layers 1-3, the minimum energy cost is almost the classical minimum at -16.8 , but at 4th layer, it abruptly lowers to -1.9 . In the same scenario, the average energy continues a downward trend, going from 10.9 to 0.5. As we go over the tables for all of the noise channels, we observe several similar patterns. As a corollary, this validates making all noise channels use the same 0.05 to 0.5 broad noise parameter range. When we choose a lesser range or a range of noise parameters that do not ruin the results (different for each noise channel), we will see similar tendencies as we have found here, despite the fact that separate noise channels impact the circuit and the findings differently. We can see similar behaviors and range of noise parameters (though the case and experiment is different) for the application of noise channels in quantum teleportation [52].

The work we have presented in this paper provides an interesting avenue for evaluating the effect of noise on detailed gate-based simulations of hybrid quantum algorithms for real-world applications. Noise is considered the most problematic aspect of today’s intermediate-scale devices, and hence understanding the details

of the effects of noise is critical in understanding how to make the most effective use of them.

In most cases, the effects of noise are minimal at the first layer of the VRP circuits. While additional layers improve upon the results in the noiseless case, the opposite is valid with the induction of noise. Since some noise will always be present in quantum circuits, an empirical finding of our results is that the COBYLA optimizer performs better for VQE circuits compared to the other available optimizers. Yet, there is room to study other optimizers, such as SPSA. As we had come across prior work on noise simulations in quantum circuits, one clear trend is that the results are heavily influenced by the optimizer used for the simulations, while some optimizers perform well at first for lower values of noise, as the noise probabilities increase the performance degrades yet among them COBYLA performs well as seen by multiple experiments[37].

We also aim to test and compare these results on more significant VRP instances in physical devices, which is beyond the ability to simulate classically. Future work is also needed to analyze more detailed noise models guided by the measured characteristics from the real NISQ devices proposed to solve problems such as VRP.

3.9 Supplementary Tables and Statistics

3.9.1 No Noise VQE Simulation Stats

Tables for VRP Simulation without noise 3.6, 3.7

Qubits	Layers	Average of energy_cost			
		COBYLA	L_BFGS_B	SLSQP	SPSA
6	1	-9.919	-3.367	-2.381	-8.753
	2	-10.985	-4.257	-3.692	-11.854
	3	-11.467	-3.792	-3.037	-12.850
	4	-13.711	-3.931	-5.420	-13.057
12	1	-30.920	-9.480	-5.941	-32.290
	2	-38.107	-12.802	-16.892	-37.090
	3	-38.461	-17.027	-18.203	-32.380
	4	-37.779	-18.312	-18.916	-32.460

Table 3.6: Table summarizing the Average Energy Cost (out of 15 runs) of VQE simulation using 5 Optimizers.

Qubits	Layers	Minimum of energy_cost			
		COBYLA	L_BFGS_B	SLSQP	SPSA
6	1	-16.776	-8.109	-6.240	-16.793
	2	-17.684	-8.033	-11.730	-17.589
	3	-17.438	-9.526	-7.461	-17.658
	4	-17.674	-7.655	-9.882	-17.651
12	1	-58.481	-39.316	-32.918	-62.181
	2	-65.227	-32.425	-35.286	-57.846
	3	-60.576	-27.507	-37.533	-55.635
	4	-60.704	-50.267	-51.096	-61.020

Table 3.7: Table containing the Minimum Energy value (out of 15 runs) across each layer of VQE simulation using 5 Optimizers.

3.9.2 Amplitude Damping Tables

Tables for Amplitude damping Noise channel containing Average and minimum energy values at various noise parameter over 10 iterations (Tables 3.8 and 3.9).

3.9. SUPPLEMENTARY TABLES AND STATISTICS

Qubits	Layers	Average Energy Cost (Amplitude Damping)									
		Noise Parameter									
		0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5
6	1	-12.053	-9.273	-10.121	-12.928	-12.491	-12.381	-11.519	-11.040	-9.440	-11.470
	2	-13.698	-12.773	-12.495	-13.909	-13.171	-12.834	-13.162	-12.760	-13.789	-12.986
	3	-15.924	-14.026	-14.999	-12.909	-14.413	-15.029	-14.246	-13.519	-15.388	-14.377
	4	-15.659	-15.055	-15.560	-14.661	-14.771	-16.147	-13.608	-13.916	-13.427	-15.757
12	1	-42.626	-42.157	-38.465	-37.787	-46.766	-41.555	-37.722	-38.388	-40.835	-47.219
	2	-47.215	-43.464	-40.368	-45.283	-48.947	-36.422	-44.922	-43.944	-46.463	-43.890
	3	-46.368	-40.817	-37.797	-39.519	-48.781	-40.945	-38.358	-42.179	-41.264	-44.605
	4	-46.110	-42.539	-42.971	-44.977	-39.561	-44.023	-40.907	-42.012	-43.650	-42.357

Table 3.8: Table containing the Average Energy values (out of 10 runs) for Amplitude Damping Noise Channel across each layer of VQE simulation Using COBYLA.

Qubits	Layers	Minimum Energy Cost (Amplitude Damping)									
		Noise Parameter									
		0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5
6	1	-16.796	-16.796	-16.799	-16.799	-16.806	-16.790	-16.799	-16.799	-16.783	-16.796
	2	-17.684	-17.684	-17.684	-17.684	-17.684	-17.638	-17.684	-17.388	-17.648	-17.684
	3	-17.467	-17.349	-17.655	-17.497	-17.043	-17.642	-17.520	-16.780	-17.684	-17.642
	4	-17.546	-17.609	-17.684	-17.480	-17.178	-17.530	-17.138	-16.609	-16.076	-17.467
12	1	-62.027	-61.905	-61.971	-61.889	-61.964	-61.961	-61.882	-61.866	-61.961	-61.879
	2	-61.908	-61.375	-55.915	-64.155	-64.033	-53.234	-62.422	-61.047	-64.905	-55.145
	3	-65.491	-55.402	-48.300	-51.375	-65.320	-53.079	-48.096	-51.954	-65.586	-65.303
	4	-63.899	-50.359	-64.895	-53.408	-48.994	-53.405	-62.089	-51.583	-56.198	-51.556

Table 3.9: Table containing the Minimum Energy values (out of 10 runs) for Amplitude Damping Noise Channel across each layer of VQE simulation Using COBYLA.

3.9.3 Bit-Flip Tables

Tables for BitFlip Noise channel containing Average and minimum energy values at various noise parameters over 10 iterations (Tables 3.10 and 3.11).

Qubits	Layers	Average Energy Cost (Bit Flip)									
		Noise Parameter									
		0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5
6	1	-13.768	-9.577	-9.661	-9.614	-10.965	-10.883	-9.809	-9.881	-10.531	-11.999
	2	-4.262	-6.576	-6.282	-5.872	-3.002	-1.198	-1.603	-3.024	-3.790	-3.075
	3	-2.332	-1.392	-3.239	-1.634	-2.651	-0.910	-1.864	-1.784	-1.621	-0.498
	4	-0.969	-1.494	-0.213	-1.312	-0.502	-0.193	-0.238	-0.012	-0.886	-0.074
12	1	-31.086	-33.947	-22.328	-38.835	-20.722	-30.384	-37.187	-35.172	-35.500	-45.676
	2	-13.251	-11.857	-7.993	-2.685	-7.242	-5.921	-0.137	-9.986	-0.931	-1.993
	3	-1.086	-0.656	-0.649	-4.338	-3.428	-1.416	-3.589	-3.328	-0.603	-2.647
	4	-0.565	-1.276	-0.596	-0.560	-0.781	-0.624	-0.657	-2.570	-0.604	-0.608

Table 3.10: Table containing the Average Energy values (out of 10 runs) for BitFlip Noise Channel across each layer of VQE simulation Using COBYLA.

CHAPTER 3. ANALYSIS OF THE VEHICLE ROUTING PROBLEM SOLVED VIA HYBRID QUANTUM ALGORITHMS IN THE PRESENCE OF NOISY CHANNELS

Qubits	Layers	Minimum Energy Cost (Bit Flip)									
		Noise Parameter Energy									
		0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5
6	1	-16.806	-16.783	-16.799	-16.783	-16.799	-16.799	-16.799	-16.793	-16.790	-16.799
	2	-8.424	-16.602	-11.780	-11.859	-16.793	-3.497	-11.786	-11.790	-10.842	-16.790
	3	-9.951	-5.599	-10.799	-9.681	-16.799	-2.658	-8.158	-15.346	-8.322	-3.109
	4	-4.089	-7.556	-1.013	-9.082	-1.898	-1.891	-2.020	-0.520	-3.635	-0.520
12	1	-62.556	-62.537	-62.547	-62.507	-56.500	-62.550	-62.566	-62.570	-62.533	-62.510
	2	-38.609	-41.921	-25.464	-25.559	-42.451	-50.056	-0.497	-56.892	-8.194	-9.020
	3	-5.632	-0.819	-0.786	-14.905	-26.730	-9.385	-24.846	-28.303	-0.727	-20.372
	4	-0.882	-6.806	-0.819	-0.895	-2.286	-0.809	-0.888	-21.122	-1.059	-0.938

Table 3.11: Table containing the Minimum Energy values (out of 10 runs) for BitFlip Noise Channel across each layer of VQE simulation Using COBYLA.

3.9.4 Phase-Flip Tables

Tables for PhaseFlip Noise channel containing Average and minimum energy values at various noise parameters over 10 iterations (Tables 3.12 and 3.13).

Qubits	Layers	Average Energy Cost (Phase Flip)									
		Noise Parameter									
		0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5
6	1	-2.033	-0.447	-0.036	0.049	0.058	0.059	0.059	0.059	0.059	0.059
	2	-1.601	-0.313	-0.055	0.058	0.058	0.059	0.059	0.059	0.059	0.059
	3	-1.755	-0.476	-0.014	0.058	0.059	0.059	0.059	0.059	0.059	0.059
	4	-1.574	-0.357	-0.022	0.059	0.059	0.059	0.059	0.059	0.059	0.059
12	1	-1.056	-0.514	-0.660	-0.581	-0.609	-0.567	-0.444	-0.548	-0.482	-0.577
	2	-0.658	-0.410	-0.383	-0.178	-0.363	-0.192	-0.255	-0.313	-0.274	-0.211
	3	-0.662	-0.520	-0.786	-0.626	-0.823	-0.736	-0.529	-0.539	-0.523	-0.593
	4	-0.533	-0.577	-0.611	-0.653	-0.680	-0.539	-0.606	-0.585	-0.681	-0.686

Table 3.12: Table containing the Average Energy values (out of 10 runs) for Phase-Flip Noise Channel across each layer of VQE simulation Using COBYLA.

Qubits	Layers	Minimum Energy Cost (Phase Flip)									
		Noise Parameter									
		0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5
6	1	-2.309	-0.493	-0.092	0.043	0.056	0.059	0.059	0.059	0.059	0.059
	2	-2.243	-0.602	-0.086	0.056	0.056	0.059	0.059	0.059	0.059	0.059
	3	-2.234	-0.599	-0.089	0.056	0.056	0.059	0.059	0.059	0.059	0.059
	4	-2.240	-0.609	-0.086	0.056	0.059	0.059	0.059	0.059	0.059	0.059
12	1	-1.559	-0.539	-0.849	-0.664	-0.822	-0.717	-0.543	-0.589	-0.589	-0.645
	2	-1.135	-0.536	-0.516	-0.451	-0.510	-0.365	-0.385	-0.438	-0.576	-0.362
	3	-0.914	-0.602	-1.013	-0.855	-1.089	-0.921	-0.618	-0.641	-0.615	-0.763
	4	-0.914	-0.914	-0.711	-0.743	-0.849	-0.622	-0.717	-0.796	-0.987	-0.730

Table 3.13: Table containing the Minimum Energy values (out of 10 runs) for PhaseFlip Noise Channel across each layer of VQE simulation Using COBYLA.

3.9.5 Bit-Phase-Flip Table

Tables for Bit-PhaseFlip Noise channel containing Average and minimum energy values at various noise parameters over 10 iterations (Tables 3.14 and 3.15).

Qubits	Layers	Average Energy Cost (Bit-PhaseFlip)									
		Noise Parameter									
		0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5
6	1	-4.241	-2.571	-3.370	-5.296	-4.159	-3.756	-4.722	-3.519	-2.716	-3.104
	2	-1.668	-1.118	-1.313	-1.411	-1.621	-1.699	-1.468	-1.304	-1.236	-1.237
	3	-0.551	-0.605	-0.550	-0.492	-0.407	-0.600	-0.723	-0.574	-0.620	-0.693
	4	-0.411	-0.446	-0.343	-0.101	-0.219	-0.368	-0.440	-0.439	-0.241	-0.221
12	1	-1.486	-1.713	-2.390	-3.355	-2.124	-0.978	-2.046	-2.405	-3.078	-1.565
	2	-0.417	-0.419	-0.299	-0.417	-0.358	-0.372	-0.431	-0.400	-0.555	-0.433
	3	-0.409	-0.460	-0.493	-0.464	-0.401	-0.349	-0.289	-0.446	-0.584	-0.345
	4	-0.510	-0.630	-0.617	-0.553	-0.508	-0.499	-0.526	-0.596	-0.736	-0.607

Table 3.14: Table containing the Average Energy values (out of 10 runs) for Bit-Phase-Flip Noise Channel across each layer of VQE simulation Using COBYLA.

Qubits	Layers	Minimum Energy Cost (Bit-Phase-Flip)									
		Noise Parameter									
		0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5
6	1	-5.355	-5.069	-5.378	-5.368	-5.352	-5.378	-5.355	-5.365	-5.378	-5.378
	2	-1.819	-1.845	-1.839	-1.852	-1.799	-1.763	-1.826	-1.770	-1.780	-1.855
	3	-0.776	-0.776	-0.799	-0.766	-0.556	-0.793	-0.796	-0.724	-0.783	-0.793
	4	-0.503	-0.533	-0.447	-0.438	-0.447	-0.530	-0.507	-0.516	-0.431	-0.533
12	1	-4.171	-4.332	-4.197	-4.523	-4.352	-2.632	-4.523	-4.516	-4.516	-4.033
	2	-0.638	-0.510	-0.368	-0.470	-0.447	-0.500	-0.609	-0.477	-0.780	-0.539
	3	-0.497	-0.717	-0.711	-0.714	-0.609	-0.408	-0.477	-0.599	-0.776	-0.582
	4	-0.737	-0.691	-0.829	-0.796	-0.609	-0.727	-0.720	-0.757	-0.885	-0.832

Table 3.15: Table containing the Minimum Energy values (out of 10 runs) for Bit-PhaseFlip Noise Channel across each layer of VQE simulation Using COBYLA.

3.9.6 Depolarising Table

Tables for Depolarising Noise channel containing Average and minimum energy values at various noise parameters over 10 iterations (Tables 3.16 and 3.17).

CHAPTER 3. ANALYSIS OF THE VEHICLE ROUTING PROBLEM SOLVED VIA HYBRID QUANTUM ALGORITHMS IN THE PRESENCE OF NOISY CHANNELS

Qubits	Layers	Average Energy Cost (Depolarising)									
		Noise Parameter									
		0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5
6	1	-4.949	-2.437	-0.750	-0.608	-0.416	-0.261	-0.045	0.006	0.034	0.050
	2	-2.377	-0.748	-0.305	-0.052	0.029	0.055	0.058	0.059	0.059	0.059
	3	-1.167	-0.396	-0.031	0.053	0.058	0.059	0.059	0.059	0.059	0.059
	4	-0.681	-0.052	0.047	0.059	0.059	0.059	0.059	0.059	0.059	0.059
12	1	-3.741	-0.919	-0.669	-0.637	-0.599	-0.653	-0.512	-0.618	-0.545	-0.565
	2	-0.718	-0.255	-0.208	-0.242	-0.231	-0.209	-0.454	-0.113	-0.444	-0.259
	3	-0.713	-0.690	-0.709	-0.649	-0.658	-0.576	-0.646	-0.605	-0.700	-0.613
	4	-0.643	-0.628	-0.650	-0.676	-0.567	-0.651	-0.599	-0.594	-0.569	-0.571

Table 3.16: Table containing the Average Energy values (out of 10 runs) for Depolarising Noise Channel across each layer of VQE simulation Using COBYLA.

Qubits	Layers	Minimum Energy Cost (Depolarising)									
		Noise Parameter									
		0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5
6	1	-7.632	-3.464	-1.651	-0.701	-0.530	-0.336	-0.102	-0.059	0.013	0.039
	2	-3.507	-1.020	-0.516	-0.089	-0.010	0.046	0.056	0.059	0.059	0.059
	3	-1.822	-0.533	-0.086	0.039	0.056	0.059	0.059	0.059	0.059	0.059
	4	-1.105	-0.115	0.030	0.056	0.059	0.059	0.059	0.059	0.059	0.059
12	1	-7.888	-1.447	-0.783	-1.000	-0.750	-0.763	-0.579	-0.793	-0.655	-0.826
	2	-1.849	-0.421	-0.461	-0.582	-0.391	-0.477	-0.855	-0.359	-0.628	-0.589
	3	-1.102	-1.039	-0.819	-0.803	-0.987	-0.743	-0.934	-0.862	-0.931	-0.793
	4	-0.862	-0.977	-0.829	-0.964	-0.750	-0.954	-0.813	-0.760	-0.697	-0.796

Table 3.17: Table containing the Minimum Energy values (out of 10 runs) for Depolarising Noise Channel across each layer of VQE simulation Using COBYLA.

SOLVING THE VEHICLE ROUTING PROBLEM VIA QUANTUM SUPPORT VECTOR MACHINES

Abstract

The Vehicle Routing Problem (VRP) is an example of a combinatorial optimization problem that has attracted academic attention due to its potential use in various contexts. VRP aims to arrange vehicle deliveries to several sites in the most efficient and economical manner possible. Quantum machine learning offers a new way to obtain solutions by harnessing the natural speedups of quantum effects, although many solutions and methodologies are modified using classical tools to provide excellent approximations of the VRP. In this paper, we employ 6 and 12 qubit circuits, respectively, to build and evaluate a hybrid quantum machine learning approach for solving VRP of 3 and 4-city scenarios. The approach employs quantum support vector machines (QSVMs) trained using a variational quantum eigensolver on a static or dynamic ansatz. Different encoding strategies are used in the experiment to transform the VRP formulation into a QSVM and solve it. Multiple optimizers from the IBM Qiskit framework are also evaluated and compared.

4.1 Introduction

4.1.1 Quantum Computing

Quantum computing has provided novel approaches for solving computationally complex problems over the last decade by leveraging the inherent speedup(s) of quantum calculations compared to classical computing. Quantum superposition and entanglement are two key factors that give a massive speed up to calculations in the quantum domain compared to classical counterparts [18, 19, 58]. Because of this, addressing Optimization problems by quantum computing is an appealing prospect. Multiple approaches, such as Grover’s algorithm [20], adiabatic computation (AC) [5], and quantum approximate optimization algorithm (QAOA) [10], have been proposed to use quantum effects and, as such, have served as the basis for solving mathematically complex problems using quantum computing. The performance of classical algorithms has generally been found to be subpar when applied to larger dimensional problem spaces [22]. On a multidimensional problem, classical machine learning optimization techniques frequently require a significant amount of CPU and GPU resources and long computation time. The reason for this is that ML techniques are needed to resolve NP-hard optimization problems [21].

4.1.2 Vehicle Routing Problem

The vehicle routing problem is an intriguing optimization problem because of its many uses in routing and fleet management [23], but its computational complexity is NP-hard [59, 60]. Moving automobiles as quickly and cheaply as feasible is always the objective. VRP has inspired a plethora of precise and heuristic approaches [23, 24], all of which struggle to provide fast and trustworthy solutions. The VRP’s bare bones implementation comprises sending a single vehicle to deliver items to many client locations before returning to the depot to restock [25]. By optimizing a collection of routes that are available and all initiate and conclude at a single node called the depot, the maximum reward sought by VRP is often expressed as the inverse of total distance traveled or mean service time. Even with just a few hundred delivery locations, finding the best solution to this problem is computationally challenging.

To be precise, in every VRP with parameters (n, k) , there are $(n - 1)$ locations, k motor vehicles, and a depot D [23, 27]. The solution is a collection of paths whereby each vehicle takes exactly one journey, and all k vehicles start and conclude at the same location, D . The best route is one that requires k vehicles to drive the fewest total miles. This problem may be thought of as a generalization of the well-known “traveling salesman” problem, whereby a set of k salesmen must service an aggregate of $(n - 1)$ sites with a single visit to each of those places being guaranteed [23]. In most practical settings, the VRP issue is complicated by other constraints, such as limited vehicle capacity or limited time for coverage. As a consequence, several other approaches, both classical and quantum, have been proposed as potential ways forwards. Currently, available quantum approaches for optimizing a system include the Quantum Approximate Optimization Algorithm (QAOA) [27], the Quadratic Unconstrained Binary Optimization (QUBO) [29], and quantum annealing [31–33].

4.1.3 Quantum Support Vector Machine (QSVM)

The goal of the support vector machine (SVM) technique is to find the best line (or decision boundary) between two classes in n -dimensional space so that new data may be classified quickly. This optimum decision boundary is referred to as a hyperplane. The most extreme vectors and points that help construct the hyperplane are selected using SVM. The SVM method is based on support vectors, which are used to represent these extreme instances. Typically, a hyperplane cannot divide a data point in its original space. In order to find this hyperplane, a nonlinear transformation is applied to the data as a function. A feature map is a function that transforms the features of provided data into the inner product of data points, also known as the kernel [9, 61, 62].

Quantum computing produces implicit calculations in high-dimensional Hilbert spaces using kernel techniques by physically manipulating quantum systems. Feature vectors for SVM in the quantum realm are represented by density operators, which are themselves encodings of quantum states. The kernel of a quantum support vector machine (QSVM) is made up of the fidelities between different feature vectors, as opposed to a classical SVM; the kernel conducts an encoding of classical

input into quantum states [9, 63].

4.1.4 Novelty and Contribution

- In this work, we propose a new method to solve the VRP using a machine-learning approach through the use of QSVM.
- In this context, we came across recent and older works in QSVM [61, 62, 64] and VQE algorithms [38], which are used to solve optimization problems such as VRP. However, none of them use a hybrid approach to arrive at a solution.
- Our work implements this new approach of solving VRP using gate-model simulation of a 3-city or 4-city problem on a 6-qubit or 12-qubit system, respectively, using a parameterized circuit that is proposed as a solution to VRP.
- We apply quantum encoding techniques such as amplitude encoding, angle encoding, higher order encoding, IQP Encoding, and quantum algorithms such as QSVM, VQE, and QAOA to construct circuits for VRP and assess the results and summarize our findings.
- We evaluate our solution using a variety of classical optimizers, as well as fixed and variable Hamiltonians to draw statistical conclusions.

4.1.5 Organization

The paper is organized as follows. Sec. 4.2 discusses the fundamental mathematical concepts such as QAOA, the Ising model, quantum support vector machine, Amplitude encoding, Angle encoding, Higher order encoding, IQP encoding, and VQE. Sec. 4.3 discusses the formulation and solution of VRP utilizing the concepts covered in the preceding Section. Sub-Sec. 4.3.2 includes the fundamental components of circuits required to solve VRP using QSVM. Sec. 4.4 covers the outcomes of the QSVM simulation consisting of two sub-sections. Sub-Sec 4.4.1 covers the outcome of simulation results of all the encoding schemes used, Finally in Sub Sec. 4.4.2, we conclude by comparing the results of QSVM solutions using various optimizers

in the Qiskit platform on the VRP circuit and discuss the feasibility of higher qubit solutions as the future directions of research.

4.2 Background

Dealing with methods and processes for resolving combinatorial optimization problems is the foundation of solving routing challenges. The objective function is then derived by converting the mathematical models into their quantum counterparts. We arrive at the objective function's solution by iteratively maximizing or minimizing the mathematical model. In this section, we provide an outline of our solution strategy's key concepts.

4.2.1 QAOA

The Quantum Approximate Optimization Algorithm (QAOA) was proposed by Farhi *et al.* in 2014 [5, 10] using an adiabatic quantum computation framework as the algorithm's foundation. It is a hybrid algorithm because both classical and quantum approaches are utilized. Quantum adiabatic computation entails transitioning between the eigenstates of the driver Hamiltonian and the problem Hamiltonian. The Hamiltonian problem can be expressed as,

$$(4.1) \quad C|z\rangle = \sum_{\alpha=1}^m C_{\alpha}|z\rangle.$$

It is well known that the combinatorial optimization problem may be effectively addressed by determining the eigenstate of C with the maximum energy. Likewise, we use driver Hamiltonian as

$$(4.2) \quad B = \sum_{j=1}^n \sigma_j^x,$$

here σ_j^x denotes the σ^x Pauli operator on bit z_j and B is the mixing operator. Let's additionally define two operators

$$(4.3) \quad U_C(\gamma) = e^{-i\gamma C}, U_B(\beta) = e^{-i\beta B}.$$

This enables the system to develop under C for γ time and B for β time. Essentially, QAOA creates the following state,

$$(4.4) \quad |\beta, \gamma\rangle = U_B(\beta_p) U_C(\gamma_p) \cdots U_B(\beta_2) U_C(\gamma_2) U_B(\beta_1) U_C(\gamma_1) |s\rangle$$

Here $|s\rangle$ represents the superposition state of all inputs. The expectation value of the cost function $\sum_{\alpha=1}^m \langle \beta, \gamma | C_\alpha | \beta, \gamma \rangle$ provides the solution, or an approximate solution to the problem [42].

4.2.2 Ising Model

The Ising model is a well-known mathematical representation of ferromagnetism in statistical mechanics [43, 44]. In the model, discrete variables (+1 or -1) represent magnetic dipole moments of spins in one of two possible states. Each spin can interact with its neighbours because they are organised in a network, commonly a lattice (when there is periodic repetition in all directions of the local structure). The spins interact in pairs, with one value of energy when the two spins are identical and another value when they are dissimilar. However, heat reverses this tendency, permitting the formation of alternative structural phases. The model is a condensed representation of reality that allows phase transitions to be identified. The subsequent Hamiltonian describes the entire spin energy:

$$(4.5) \quad H_c = - \sum_{\langle i,j \rangle} J_{ij} \sigma_i \sigma_j - h \sum \sigma_i,$$

where J_{ij} is the interaction between adjacent spins i and j , and h is an external magnetic field. At $h = 0$, the ground state is ferromagnetic if J is positive. At $h = 0$, the ground state is antiferromagnetic if J is negative in a bipartite lattice. Thus, for the purpose of clarity and within the scope of this paper, the Hamiltonian can be expressed as

$$(4.6) \quad H_c = - \sum_{\langle i,j \rangle} J_{ij} \sigma_i^z \sigma_j^z - \sum h_i \sigma_i^x.$$

Here, σ_z and σ_x represent the z and x Pauli operators, respectively. For the sake of simplification, we can presume the following conditions to be ferromagnetic

($J_{ij} > 0$) if there is no external impact on the spin: $h = 0$. The Hamiltonian may therefore be reformulated as follows:

$$(4.7) \quad H_c = - \sum_{\langle i,j \rangle} J_{ij} \sigma_i^z \sigma_j^z = - \sum_{\langle i,j \rangle} \sigma_i^z \sigma_j^z.$$

4.2.3 Quantum Support Vector Machine

SVM [61, 62] is a supervised algorithm that constructs hyper-plane with $\vec{w} \cdot \vec{x} + b = 0$ such that $\vec{w} \cdot \vec{x} + b \geq 1$ for a training point \vec{x}_i in the positive class, and $\vec{w} \cdot \vec{x} + b \leq -1$ for a training point \vec{x}_i in the negative class. During the training process, the algorithm aims to maximize the gap between the two classes, which is intuitive as we want to separate two classes as far as possible, in order to get a sharper estimate for the classification result of new data samples like \vec{x}_0 . Mathematically, we can see the objective of SVM is to find a hyper-plane that maximizes the distance $2/|\vec{w}|$ constraint to $\vec{y}_i(\vec{w} \cdot \vec{x}_i + b) \geq 1$. The normal vector \vec{w} can be written as $\vec{w} = \sum_{i=1}^M \alpha_i \vec{x}_i$ where α_i is the weight of the i^{th} training vector \vec{x}_i . Thus, obtaining optimal parameters b and α_i is the same as finding the optimal hyper-plane. To classify the new vector is analogous to knowing which side of the hyper-plane it lies, i.e., $y_i(\vec{x}_0) = \text{sign}(\vec{w} \cdot \vec{x} + b)$. After having the optimal parameters, classification now becomes a linear operation. From the least-squares approximation of SVM, the optimal parameters can be obtained by solving a linear equation,

$$(4.8) \quad \vec{F}(b, \alpha_1, \alpha_2, \alpha_3, \dots, \alpha_M)^T = (0, y_1, y_2, y_3, \dots, y_M)^T.$$

In a general form of F , we adopt the linear kernels $K_{i,j} = \kappa(\vec{x}_i, \vec{x}_j) = \vec{x}_i \cdot \vec{x}_j$. Thus, to find the hyper-plane parameters we use matrix inversion of $F : (b, \vec{\alpha}_i^T)^T = \vec{F}^{-1}(0, \vec{y}_i^T)^T$.

4.2.3.1 Quantum Kernels

The main inspiration for a quantum Support vector machine is to consider quantum feature maps that lead to quantum kernel functions, which are hard to simulate in classical computers. In this case, the quantum computer is only used to estimate a

quantum kernel function, which can be later used in kernel-based algorithms. For simplicity assuming the datapoints $x, z \in \mathcal{X}$, the nonlinear feature map of any data point is

$$(4.9) \quad \Phi(\mathbf{x}) = U(\mathbf{x})|0^n\rangle\langle 0^n|U^\dagger(\mathbf{x}).$$

The kernel function $\kappa(x, z)$ can be computed as

$$(4.10) \quad \kappa(x, y) = |\langle \phi(x) | \phi(z) \rangle|^2.$$

The state $|\phi(x)\rangle$ can be prepared by using a unitary gate $U(x)$, and thus $|\phi(x)\rangle = U(x)|0\rangle$. Thus the kernel function becomes ,

$$(4.11) \quad \kappa(x, z) = \left| \langle 0 | U^\dagger(x)U(z) | 0 \rangle \right|^2.$$

From the above we can say that the kernel $\kappa(x, z)$ is simply the probability of getting an all-zero string when the circuit $U^\dagger(x)U(z)|0\rangle$ is measured, or this kernel is an $|0^n\rangle$ to $|0^n\rangle$ transition probability of a particular unitary quantum circuit on n qubits [9, 65]. This can be implemented using the following kernel estimation circuit (Fig. 4.1).

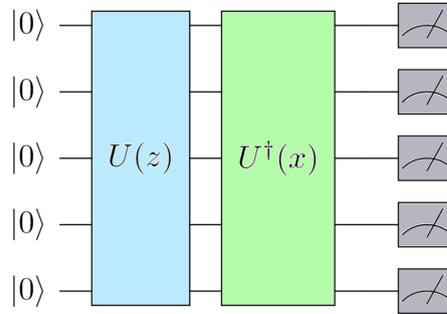


Figure 4.1: Schematic diagram depicting quantum circuit for Kernel estimation.

4.2.4 Amplitude Encoding(AE)

In the process of amplitude-embedding [66], data is encoded into the amplitudes of a quantum state. A N-dimensional classical datapoint x is represented by the amplitudes of an n-qubit quantum state $|\psi_x\rangle$ as

$$(4.12) \quad |\psi_x\rangle = \sum_{i=1}^N x_i |i\rangle$$

where $N = 2^n$, x_i is the i -th element of x and $|i\rangle$ is the i -th computational basis state.

In order to encode any data point x into an amplitude-encoded state, we must normalize the same by following

$$(4.13) \quad |\psi_{x_{norm}}\rangle = \frac{1}{x_{norm}} \sum_{i=1}^N x_i |i\rangle,$$

$$\text{where } x_{norm} = \sqrt{\sum_{i=1}^N |x_i|^2}.$$

4.2.5 Angle Encoding (AgE)

While the above-described amplitude encoding expands into a complicated quantum circuit with huge depths, the angle encoding employs N qubits and a quantum circuit with fixed depth, making it favorable to NISQ computers [67, 68]. We define angle encoding as a method of classical information encoding that employs rotation gates (the rotation could be chosen along x , y or z axis). In our scenario, the classical information consists of the node and edge weights assigned to the vehicle's nodes and pathways, which are further assigned as parameters to ansatz.

$$(4.14) \quad |\mathbf{x}\rangle = \bigotimes_i^n R(\mathbf{x}_i) |0^n\rangle,$$

Where x_i represents the classical information stored on the angle parameter of rotation operator R .

4.2.6 Higher Order Encoding(HO)

Higher order encoding is a variation of angle encoding where we have an entangled layer and an additional sequential operation of rotation angles of two entangled qubits [68]. This can be loosely defined as the following

$$(4.15) \quad |\mathbf{x}\rangle = \bigotimes_{i=2}^n R(x_{i-1}, x_i) \bigotimes_{i=2}^{n-1} CX_{i,i+1} \bigotimes_{i=1}^n R(x_i) |0^n\rangle.$$

Similar to angle encoding we are free to choose the rotation.

4.2.7 IQP Encoding(IqpE)

IQP-style encoding is a relatively complicated encoding strategy. We encode classical information [69]

$$(4.16) \quad |x\rangle = (\mathbf{U}_Z(x)\mathbf{H}^{\otimes n})^r |0^n\rangle,$$

where r is the depth of the circuit, indicating the repeating times of $\mathbf{U}_Z(x)\mathbf{H}^{\otimes n}$. $\mathbf{H}^{\otimes n}$ is a layer of Hadamard gates acting on all qubits. $\mathbf{U}_Z(\mathbf{x})$ is the key step in IQP encoding scheme:

$$(4.17) \quad \mathbf{U}_Z(\mathbf{x}) = \prod_{[i,j] \in S} R_{Z_i Z_j}(x_i x_j) \bigotimes_{k=1}^n R_z(x_k),$$

where S is the set containing all pairs of qubits to be entangled using R_{ZZ} gates. First, we consider a simple two-qubit gate: $R_{Z_1 Z_2}(\theta)$. Its mathematical form $e^{-i\frac{\theta}{2}Z_1 \otimes Z_2}$ can be seen as a two-qubit rotation gate around ZZ , which makes these two qubits entangled.

4.2.8 VQE

Variational Quantum Eigensolver (VQE) is another hybrid quantum-classical algorithm used for the estimation of the eigenvalue of a matrix or Hamiltonian H [46] of significant size. The primary objective of this approach is to ascertain a trial qubit state from a wave function $|\psi(\vec{\theta})\rangle$ that relies on a collection of parameters $\vec{\theta} = \theta_1, \theta_2, \dots$, which are often referred to as the variational parameters. The expectation of an observable or Hamiltonian H in a state $|\psi(\vec{\theta})\rangle$ can be expressed as follows,

$$(4.18) \quad E(\vec{\theta}) = \langle \psi(\vec{\theta}) | H | \psi(\vec{\theta}) \rangle.$$

By spectral decomposition

$$H = \lambda_1 |\psi\rangle_1 \langle\psi|_1 + \lambda_2 |\psi\rangle_2 \langle\psi|_2 + \dots + \lambda_n |\psi\rangle_n \langle\psi|_n, \quad (4.19)$$

where λ_i and $|\psi\rangle_i$ are the matrix H 's eigenvalues and eigenstates, respectively. Additionally, because H 's eigenstates are orthogonal, $\langle\psi_i | \psi_j\rangle = 0$ If $i \neq j$. The wave function $|\psi(\vec{\theta})\rangle$ can be expressed as a superposition of eigenstates.

$$|\psi(\vec{\theta})\rangle = \alpha_1(\vec{\theta}) |\psi\rangle_1 + \alpha_2(\vec{\theta}) |\psi\rangle_2 + \dots + \alpha_n(\vec{\theta}) |\psi\rangle_n. \quad (4.20)$$

Hence the expectation is given by,

$$E(\vec{\theta}) = |\alpha_1(\vec{\theta})|^2 \lambda_1 + |\alpha_2(\vec{\theta})|^2 \lambda_2 + \dots + |\alpha_n(\vec{\theta})|^2 \lambda_n. \quad (4.21)$$

Hence, $E(\vec{\theta}) \geq \lambda_{\min}$. The VQE method involves the iterative adjustment of the parameters $\vec{\theta} = \theta_1, \theta_2, \dots$ in order to minimise the value of $E(\vec{\theta})$. This property of VQE is advantageous when attempting to solve combinatorial optimization problems. Specifically, the approach involves using a parameterized circuit to establish the trial state of the algorithm, with the cost function denoted as $E(\vec{\theta})$, which is also the expected value of the Hamiltonian in this state. It is possible to derive the ground state of the desired Hamiltonian by iteratively minimizing the cost function. A classical optimizer uses a quantum computer to calculate its gradient and assess the cost function at each step of the optimization process.

4.3 Methodology

4.3.1 Modelling VRP in QSVM

By mapping the cost function to an Ising Hamiltonian H_c , the vehicle routing problem can be solved [45]. The solution to the problem is determined by minimizing the Ising Hamiltonian H_c . Consider a graph with n vertices and $n - 1$ edges and an arbitrary connectivity. Assuming we must route a vehicle between two

non-adjacent vertices in the graph, consider a binary decision variable x_{ij} whose value is 1 if there is an edge between i and j with an edge weight $w_{ij} > 0$ and 0 otherwise. Now, the VRP problem necessitates $n \times (n - 1)$ selection variables. We define two sets of nodes for each edge from $i \rightarrow j$: source $s[i]$ and target $t[j]$. $s[i]$ contains the nodes j to which i sends an edge $j \in s[i]$. The collection $t[j]$ comprises the nodes i to which the node i delivers the edge $i \in t[j]$. The VRP is defined as follows[27, 47]:

$$(4.22) \quad VRP(n, k) = \min_{\{x_{ij}\}_{i \rightarrow j} \in \{0,1\}} \sum_{i \rightarrow j} w_{ij} x_{ij},$$

where k and n represent the number of vehicles and locations respectively, there are $n - 1$ locations for vehicles to traverse if the starting point is considered to be the $0th$ location or Depot D . Notably, the following restrictions apply to this [?]:

$$(4.23) \quad \begin{aligned} \sum_{j \in s[i]} x_{ij} &= 1, \forall i \in \{1, \dots, n-1\}, \\ \sum_{j \in t[i]} x_{ji} &= 1, \forall i \in \{1, \dots, n-1\}, \\ \sum_{j \in s[0]} x_{0j} &= k, \\ \sum_{j \in t[0]} x_{j0} &= k \\ u_i - u_j + Qx_{ij} &\leq Q - q_j, \forall i \sim j, i, j \neq 0, \\ q_i \leq u_i &\leq Q, \forall i, i \neq 0. \end{aligned}$$

The first two restrictions establish that each node may only be visited once by the delivering vehicle. The middle two limitations enforce the requirement that after product delivery, the vehicle must return to the depot. The last two restrictions enforce the requirements for eliminating sub-tours and are constrained on u_i , with $Q > q_j > 0$, and $u_i, Q, q_i \in \mathbb{R}$. For the VRP equation and constraints, the VRP Hamiltonian can be expressed as follows[27].

$$\begin{aligned}
 H_{VRP} &= H_a + H_b + H_c + H_d + H_e, \\
 H_a &= \sum_{i \rightarrow j} w_{ij} x_{ij}, \\
 H_b &= A \sum_{i \in 1, \dots, n-1} \left(1 - \sum_{j \in s[i]} x_{ij} \right)^2, \\
 H_c &= A \sum_{i \in 1, \dots, n-1} \left(1 - \sum_{j \in t[i]} x_{ji} \right)^2, \\
 H_d &= A \left(k - \sum_{j \in s[0]} x_{0j} \right)^2, \\
 H_e &= A \left(k - \sum_{j \in t[0]} x_{j0} \right)^2.
 \end{aligned}
 \tag{4.24}$$

$A > 0$ is indicative of a constant. The vector representation of the collection of all binary decision variables x_{ij} is

$$\vec{\mathbf{x}} = [x_{(0,1)}, x_{(0,2)}, \dots, x_{(1,0)}, x_{(1,2)}, \dots, x_{(n-1,n-2)}]^T.
 \tag{4.25}$$

Using the antecedent vector, we can construct two new vectors for each node: $\vec{z}_{S[i]}$ and $\vec{z}_{T[i]}$ (at the start of the section, we defined two sets for source and target nodes, so two vectors will represent them).

$$\begin{aligned}
 \vec{z}_{S[i]} &= \vec{\mathbf{x}} \ni x_{ij} = 1, x_{kj} = 0, k \neq i, \forall j, k \in \{0, \dots, n-1\}, \\
 \vec{z}_{T[i]} &= \vec{\mathbf{x}} \ni x_{ji} = 1, x_{jk} = 0, k \neq i, \forall j, k \in \{0, \dots, n-1\}.
 \end{aligned}
 \tag{4.26}$$

$$\begin{aligned}
 \sum_{j \in s[i]} x_{ij} &= \vec{z}_{S[i]}^T \vec{\mathbf{x}}, \\
 \sum_{j \in t[i]} x_{ji} &= \vec{z}_{T[i]}^T \vec{\mathbf{x}}.
 \end{aligned}
 \tag{4.27}$$

These vectors will contribute to the development of the QUBO model of VRP [29, 30, 48, 49]. The QUBO model of a connected graph $G = (N, V)$ is specified as follows:

$$(4.28) \quad f(x)_{QUBO} = \min_{x \in \{0,1\}^{(N \times V)}} x^T Q x + g^T x + c,$$

where, Q is a quadratic edge weight coefficient, g is a linear node weight coefficient, and c is a constant. To determine the coefficients in the QUBO formulations of H_{VRP} as shown in Eq. 4.24, the equations in Eq. 4.27 are first substituted in terms of H_b and H_c , respectively. Subsequently, the expression of H_{VRP} is expanded and rearranged in accordance with Eq. 4.28.

$$(4.29) \quad \begin{aligned} H &= A \sum_{i=0}^{n-1} \left[z_{S[i]} z_{S[i]}^T + z_{T[i]} z_{T[i]}^T \right] \vec{x}^2 \\ &+ w^T \vec{x} - 2A \sum_{i=1}^{n-1} \left[z_{S[i]}^T + z_{T[i]}^T \right] \vec{x} \\ &- 2Ak \left[z_{S[0]}^T + z_{T[0]}^T \right] \vec{x} + 2A(n-1) + 2Ak^2. \end{aligned}$$

Hence, in the QUBO formulation of the Eq. (4.24), the coefficients $Q(n(n-1) \times n(n-1))$, $g(n(n-1) \times 1)$, and c are derived. The coefficients associated with the QUBO formulation of Eq. (4.24) are shown below.

$$(4.30) \quad \begin{aligned} Q &= A \left[[z_{T[0]}, \dots, z_{T[n-1]}]^T [z_{T[0]}, \dots, z_{T[n-1]}] \right. \\ &\quad \left. + (\mathbb{1}_n \otimes \mathbb{J}(n-1, n-1)) \right], \\ g &= W - 2Ak \left((e_0 \otimes \mathbb{J}_{n-1}) + [z_{T[0]}]^T \right), \\ &\quad + 2A(\mathbb{J}_n \otimes \mathbb{J}_{n-1}), \\ c &= 2A(n-1) + 2Ak^2. \end{aligned}$$

\mathbb{J} is the matrix containing all ones, $\mathbb{1}$ and $e_0 = [1, 0, \dots, 0]^T$ are the identity matrices. The binary decision variable x_{ij} is converted to the spin variable $s_{ij} \in \{-1, 1\}$ using the formula $x_{ij} = (s_{ij} + 1)/2$.

From the aforementioned equations, we may expand Eq. (4.28) to form the Ising Hamiltonian of VRP [29].

$$(4.31) \quad H_{Is} = -\sum_i \sum_{i<j} J_{ij} s_i s_j - \sum_i h_i s_i + d.$$

Following are definitions for the terms J_{ij} , h_i , and d :

$$(4.32) \quad \begin{aligned} J_{ij} &= -\frac{Q_{ij}}{2}, \forall i < j, \\ h_i &= \frac{g_i}{2} + \sum \frac{Q_{ij}}{4} + \sum \frac{Q_{ji}}{4}, \\ d &= c + \sum_i \frac{g_i}{2} + \sum_i \sum_j \frac{Q_{ij}}{4}. \end{aligned}$$

4.3.2 Analysis And Circuit Building

4.3.2.1 VRP

In the current section, we proceed to create a circuit based on gates using the IBM gate model. The implementation of this model is carried out using the Qiskit framework [50], enabling us to effectively execute the aforementioned formulation. In the context of a given Vehicle Routing Problem (VRP) that incorporates qubits, the initial state is established as $|+\rangle^{\otimes n(n-1)}$. This state represents the ground state of $H_{m_{xr}}$, which is achieved by applying the Hadamard gate to each qubit that has been initialized to the zero state. Subsequently, we proceed to build the subsequent state.

$$(4.33) \quad \begin{aligned} |\beta, \gamma\rangle &= e^{-iH_{m_{xr}}\beta_p} e^{-iH_c\gamma_p} \dots \\ &\dots e^{-iH_{m_{xr}}\beta_0} e^{-iH_c\gamma_0} |+\rangle^{n\otimes(n-1)}. \end{aligned}$$

The energy E of the state $|\beta, \gamma\rangle$ is computed using the expectation of H_c from Equation (4.18). Again based on the Ising model, the term H_c can be expressed in terms of Pauli operators as follows:

$$(4.34) \quad H_c = -\sum_i \sum_{i<j} J_{ij} \sigma_i^z \sigma_j^z - \sum_i h_i \sigma_i^z - d.$$

Thus, the expression for a single term of state in $|\beta, \gamma\rangle$ as β_0, γ_0 reads:
 $e^{-iH_{mxr}\beta_0} e^{-iH_c\gamma_0}$. The first term H_c can be expanded to following,

$$\begin{aligned}
 e^{iJ_{ij}\gamma_0\sigma_i\sigma_j} &= \cos J_{ij}\gamma_0 I + i \sin J_{ij}\gamma_0 \sigma_i \sigma_j, \\
 &= e^{iJ_{ij}\gamma_0} |00\rangle\langle 00| + e^{-iJ_{ij}\gamma_0} |01\rangle\langle 01| + e^{-iJ_{ij}\gamma_0} |10\rangle\langle 10| + e^{iJ_{ij}\gamma_0} |11\rangle\langle 11|, \\
 (4.35) \quad &= M
 \end{aligned}$$

It can be noted that by the application of $CNOT(CX)$ gate before and after ‘ M ’, the diagonal elements of the above matrix can be swapped.

$$\begin{aligned}
 CX(M)CX &= e^{iJ_{ij}\gamma_0} |00\rangle\langle 00| - e^{-iJ_{ij}\gamma_0} |01\rangle\langle 01| + e^{-iJ_{ij}\gamma_0} |10\rangle\langle 10| - e^{iJ_{ij}\gamma_0} |11\rangle\langle 11|, \\
 (4.36)
 \end{aligned}$$

Expanding the matrix and observing the upper and lower blocks we can rewrite,

$$\begin{aligned}
 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \otimes \begin{bmatrix} e^{iJ_{ij}\gamma_0} & 0 \\ 0 & e^{-iJ_{ij}\gamma_0} \end{bmatrix} &= I \otimes e^{iJ_{ij}\gamma_0} \begin{bmatrix} 1 & 0 \\ 0 & e^{-2iJ_{ij}\gamma_0} \end{bmatrix}. \\
 (4.37)
 \end{aligned}$$

$\begin{bmatrix} 1 & 0 \\ 0 & e^{-2iJ_{ij}\gamma_0} \end{bmatrix}$ is a phase gate. Looking at the 2nd term of H_c we get,

$$\begin{aligned}
 H_c &= \sum_i h_i \sigma_i^z, \\
 e^{ih_i\sigma_i} &= \cos h_i\gamma_0 I + i \sin h_i\gamma_0 \sigma_i, \\
 &= \cos h_i\gamma_0 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + i \sin h_i\gamma_0 \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \\
 (4.38) \quad &= \begin{bmatrix} e^{ih_i\gamma_0} & 0 \\ 0 & e^{-ih_i\gamma_0} \end{bmatrix}.
 \end{aligned}$$

Fig. 4.2(a) picturizes the fundamental circuit with two qubits and gate selections for H_c . Similarly, H_{mxr} is merely a rotation along the X axis, as depicted by the U gate in Fig. 4.2(b), Fig. 4.2(a).

The above sample circuits can be used for the solution of VRP combined with the VQE and QAOA approach. However, in this paper, we are focusing on a machine-learning solution of VRP by use of QSVM; thus we need to construct a QSVM circuit using various encoding schemes. Simple interpretation and implementation of encoding schemes are described in upcoming subsections.

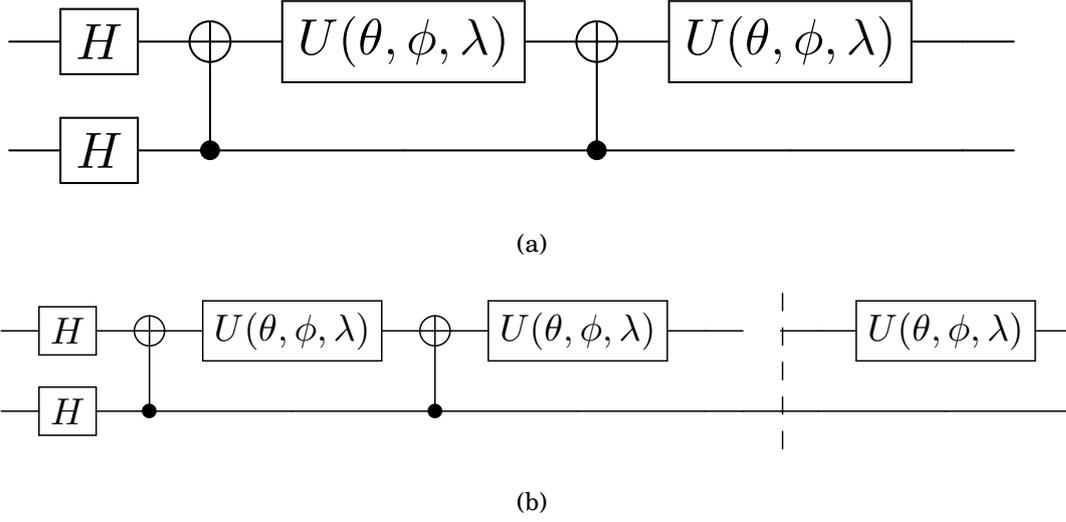


Figure 4.2: (a) Circuit example illustrating gate operations for H_c . (b) Circuit example displaying gate selections with an additional u gate for H_{mxr} .

4.3.2.2 Amplitude Encoding

As we look into AE, a single qubit state is represented by

$$(4.39) \quad |\psi\rangle(\theta) = \cos(\theta/2)|0\rangle + \sin(\theta/2)|1\rangle,$$

for two qubits

$$(4.40) \quad \begin{aligned} |\psi(\theta)\rangle &= \alpha|00\rangle + \beta|01\rangle + \gamma|10\rangle + \delta|11\rangle, \\ &= |0\rangle(\alpha|0\rangle + \beta|1\rangle) + |1\rangle(\gamma|0\rangle + \delta|1\rangle), \\ &= |0\rangle\sqrt{\alpha^2 + \beta^2} \left(\frac{\alpha|0\rangle + \beta|1\rangle}{\sqrt{\alpha^2 + \beta^2}} \right) \\ &\quad + |1\rangle\sqrt{\gamma^2 + \delta^2} \frac{\gamma|0\rangle + \delta|1\rangle}{\sqrt{\gamma^2 + \delta^2}}. \end{aligned}$$

Now applying Ctrl U and Anti-CTRL U on the above state we achieve

$$\begin{aligned}
 & |0\rangle\sqrt{\alpha^2 + \beta^2} |0\rangle + |1\rangle\sqrt{\gamma^2 + \delta^2} |0\rangle \\
 (4.41) \quad & = \left(\sqrt{\alpha^2 + \beta^2} |0\rangle + \sqrt{\gamma^2 + \delta^2} |1\rangle \right) |0\rangle.
 \end{aligned}$$

Here $\theta_1 = \tan^{-1} \frac{\sqrt{\gamma^2 + \delta^2}}{\sqrt{\alpha^2 + \beta^2}}$, $\theta_2 = \tan^{-1} \frac{\delta}{\gamma}$, $\theta_3 = \tan^{-1} \frac{\beta}{\alpha}$ Combining VRP and amplitude encoding circuit eliminates the need for Hadamard gates and $H_{m \times r}$ components and we end up with the following skeleton circuits Figure ??.

4.3.2.3 Angle Encoding

For a 2-qubit scenario, angle encoding translates to the following example. We define the R_y gate as follows

$$\begin{aligned}
 R_y(\theta) & = e^{-iY\theta/2} = \cos \frac{\theta}{2} - i \sin \theta/2Y, \\
 (4.42) \quad & = \begin{bmatrix} \cos \theta/2 & -\sin \theta/2 \\ \sin \theta/2 & \cos \theta/2 \end{bmatrix}.
 \end{aligned}$$

$$\begin{aligned}
 & |00\rangle \\
 & \xrightarrow{R_y(\theta_1)} \left(\cos \frac{\theta_1}{2} |0\rangle + \sin \frac{\theta_1}{2} |1\rangle \right) \left(\cos \frac{\theta_2}{2} |0\rangle + \sin \frac{\theta_2}{2} |1\rangle \right), \\
 & = \cos \frac{\theta_1}{2} \cdot \cos \frac{\theta_2}{2} |00\rangle + \cos \frac{\theta_1}{2} \cdot \sin \frac{\theta_2}{2} |01\rangle \\
 & + \sin \frac{\theta_1}{2} \cdot \cos \frac{\theta_2}{2} |10\rangle + \sin \frac{\theta_1}{2} \sin \frac{\theta_2}{2} |11\rangle \\
 & \xrightarrow{CNOT} \cos \frac{\theta_1}{2} \cdot \cos \frac{\theta_2}{2} |00\rangle + \cos \frac{\theta_1}{2} \cdot \sin \frac{\theta_2}{2} |01\rangle \\
 (4.43) \quad & + \sin \frac{\theta_1}{2} \cdot \cos \frac{\theta_2}{2} |11\rangle + \sin \frac{\theta_1}{2} \sin \frac{\theta_2}{2} |10\rangle.
 \end{aligned}$$

4.3.2.4 Higher Order Encoding

For a 2qubit scenario, HO encoding translates to the following We define the R_y gate as follows

$$(4.44) \quad \begin{aligned} R_y(\theta) &= e^{-iY\theta/2} = \cos\frac{\theta}{2} - i \sin\theta/2Y, \\ &= \begin{bmatrix} \cos\theta/2 & -\sin\theta/2 \\ \sin\theta/2 & \cos\theta/2 \end{bmatrix}. \end{aligned}$$

$$(4.45) \quad \begin{aligned} &|00\rangle \\ &\frac{R_y(\theta_1)}{R_y(\theta_2)} \left(\cos\frac{\theta_1}{2}|0\rangle + \sin\frac{\theta_1}{2}|1\rangle \right) \left(\cos\frac{\theta_2}{2}|0\rangle + \sin\frac{\theta_2}{2}|1\rangle \right), \\ &= \cos\frac{\theta_1}{2} \cdot \cos\frac{\theta_2}{2}|00\rangle + \cos\frac{\theta_1}{2} \cdot \sin\frac{\theta_2}{2}|01\rangle \\ &+ \sin\frac{\theta_1}{2} \cdot \cos\frac{\theta_2}{2}|10\rangle + \sin\frac{\theta_1}{2} \sin\frac{\theta_2}{2}|11\rangle \\ &\frac{CNOT}{R_y(\theta_1, \theta_2)} \cos\frac{\theta_1}{2} \cdot \cos\frac{\theta_2}{2}|0\rangle \left(\cos\frac{\theta_1 \cdot \theta_2}{2}|0\rangle + \sin\frac{\theta_1 \cdot \theta_2}{2}|1\rangle \right) \\ &+ \cos\frac{\theta_1}{2} \cdot \sin\frac{\theta_2}{2}|0\rangle \left(-\sin\frac{\theta_1 \cdot \theta_2}{2}|0\rangle + \cos\frac{\theta_1 \cdot \theta_2}{2}|1\rangle \right) \\ &+ \sin\frac{\theta_1}{2} \cos\frac{\theta_2}{2}|1\rangle \left(\cos\frac{\theta_1 \cdot \theta_2}{2}|0\rangle + \sin\frac{\theta_1 \cdot \theta_2}{2}|1\rangle \right) \\ &+ \sin\frac{\theta_1}{2} \cdot \sin\frac{\theta_2}{2}|1\rangle \left(-\sin\frac{\theta_1 \cdot \theta_2}{2}|0\rangle + \cos\frac{\theta_1 \cdot \theta_2}{2}|1\rangle \right). \end{aligned}$$

4.3.2.5 IQP Encoding

For a 2qubit scenario IqpE translates to the following

$$\begin{aligned}
 & |00\rangle \xrightarrow{H_1 H_2} |++\rangle, \\
 & = \frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle), \\
 & \xrightarrow{\substack{R_Z(\theta_1) \\ R_Z(\theta_2)}} \frac{1}{2} \left(|00\rangle + e^{i\theta_2}|01\rangle + e^{i\theta_1}|10\rangle + e^{i(\theta_1+\theta_2)}|11\rangle \right) \\
 (4.46) \quad & \xrightarrow{\text{CNOT}} \frac{1}{2} \left(|00\rangle + e^{i\theta_2}|01\rangle + e^{i\theta_1}|11\rangle + e^{i(\theta_1+\theta_2)}|10\rangle \right) \\
 & \xrightarrow{R_Z(\theta_1 \cdot \theta_2)} \frac{1}{2} \left(|00\rangle + e^{i\theta_2(1+\theta_1)}|01\rangle + e^{i\theta_1(1+\theta_2)}|11\rangle \right. \\
 & \left. + e^{i\theta_1(1+\theta_2)}|10\rangle + e^{i(\theta_1+\theta_2)}|11\rangle \right).
 \end{aligned}$$

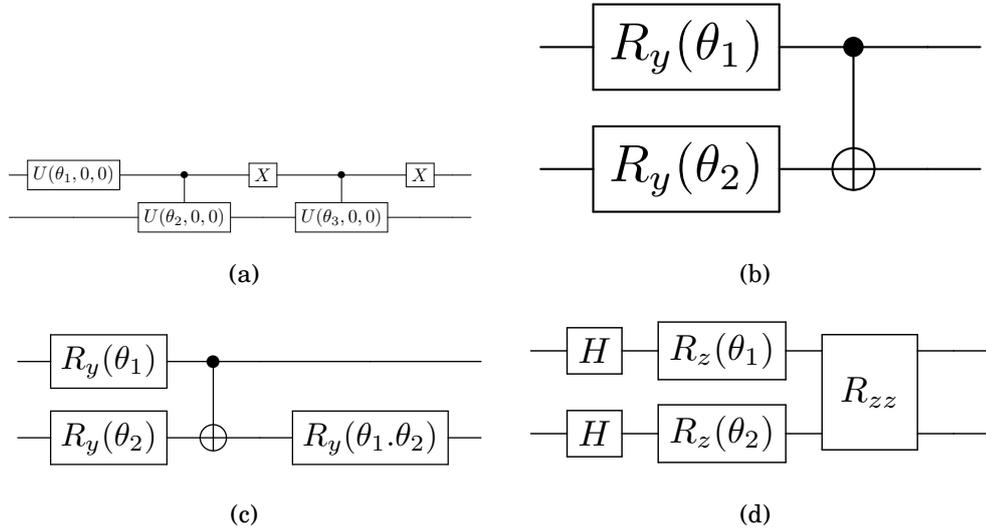


Figure 4.3: Plot illustrating different encoding methods for two qubits. (a) Amplitude encoding, (b) angle encoding, (c) Higher order encoding, (d) IQP encoding.

4.4 Results

4.4.1 VQE Simulation of QSVM and VRP

We build the Hamiltonian with a uniform distribution of weights between 0 and 1, and then run it along with the ansatz via IBM's three available VQE optimizers (COBYLA, L_BFGS_B, and SLSQP). We run the circuit up to two layers and gather data using all of the available optimizers. We run the experiment again with a fixed Hamiltonian and, subsequently, a set of variable Hamiltonians to see whether the QSVM and encoding approach can effectively reach the classical minimum. Our results indicate that COBYLA is the most efficient optimizer, followed by SLSQP and L BFGS B. In the sections that follow, we'll have a look at the results obtained using various QSVM encoding schemes. We define two terms, "Accuracy" and "Error," in the context of outcomes' interpretability. An error occurs when the solution deviates from the classical minimum more often than it reaches it, whereas accuracy is defined as the number of times the solution reaches the classical minimum. Percentages based on the distribution of the outcomes are used to evaluate both terms.

$$(4.47) \quad \begin{aligned} Acc &= \frac{N}{T}, \\ Err &= \frac{T-N}{T}. \end{aligned}$$

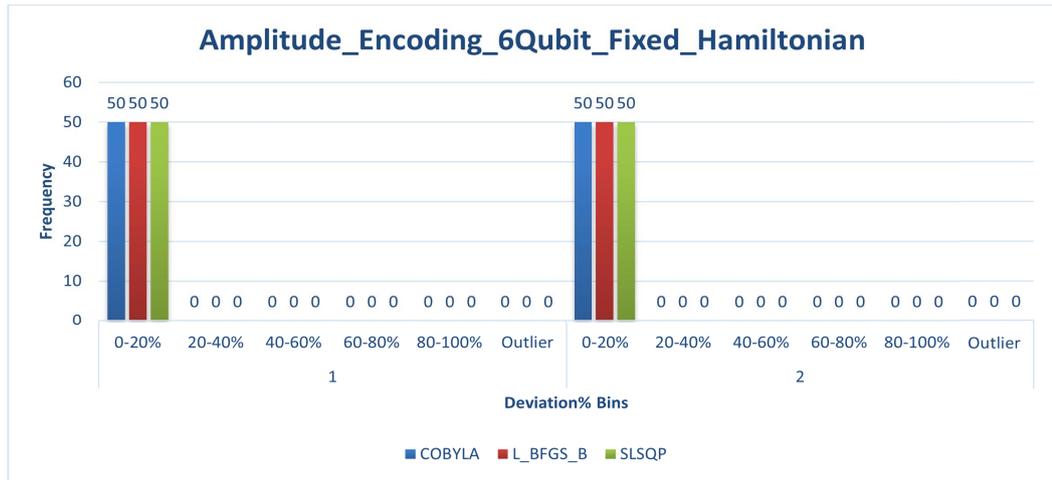
T = Total number of Simulation runs

N = Total number of times solution reaches classical minimum

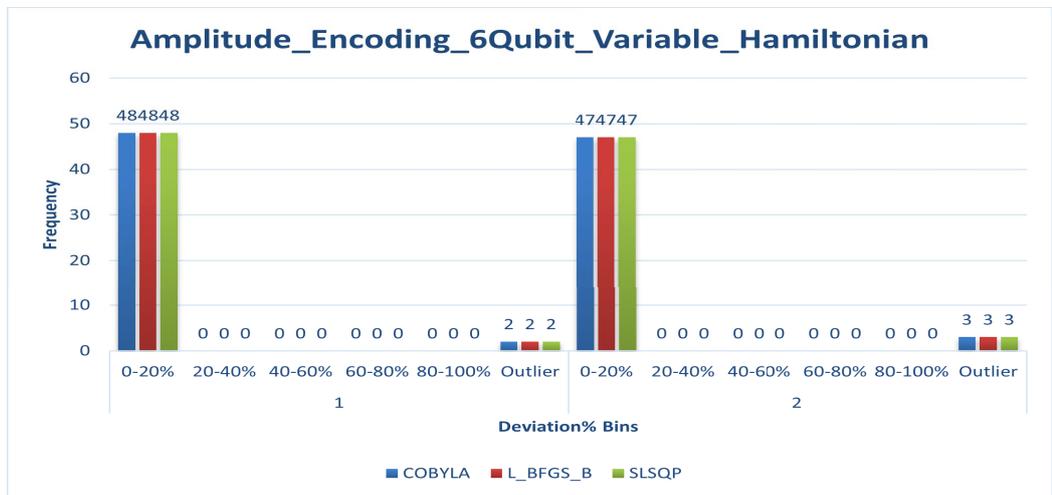
4.4.1.1 Amplitude Encoding

With a large number of gates, the AE circuit has proven to be the most complex of all encoding circuits. We can simulate no more than six qubit computations due to this complexity. Despite its complexity, AE has a high, nearly perfect accuracy rate (100%) and a very low error rate (0%) for 50-iteration fixed Hamiltonian simulations. The trend is present in both the first and second layers. The first layer accuracy for a variable Hamiltonian simulation is 96%, and the second

layer accuracy is 94% across all optimizers. Figures 4.4 depicts the results of 50 iterations of simulating SVM with amplitude encoding on a VRP circuit with fixed and variable Hamiltonian. The decline in accuracy, however, can be attributed to simulation or computational errors, as all the errors are greater than 100 percent and are therefore considered aberrations. Most likely, the simulation hardware cannot accommodate the VQE procedure.



(a)



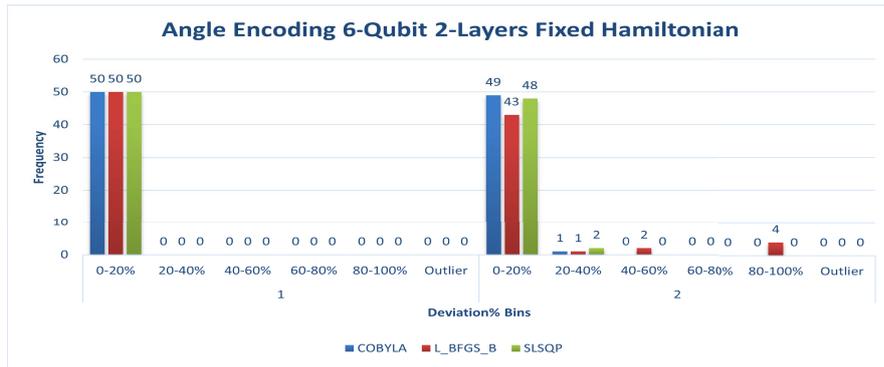
(b)

Figure 4.4: Plot illustrating Amplitude encoding results for QSVM solution of VRP. (a) Amplitude encoding 6 qubits Fix Hamiltonian, (b) Amplitude encoding 6 qubits Variable hamiltonian

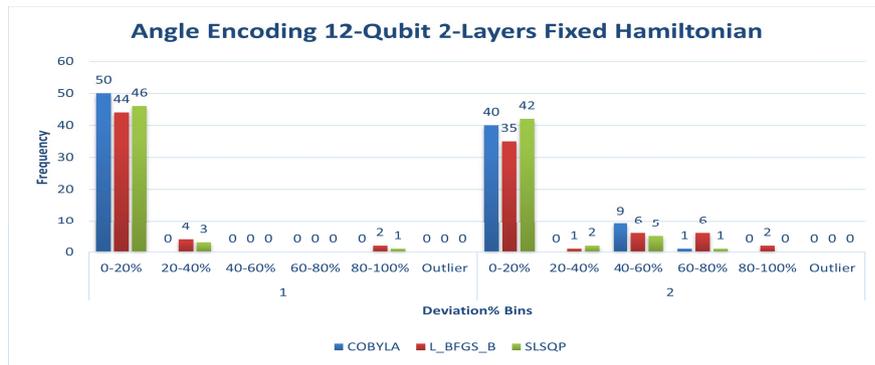
4.4.1.2 Angle Encoding

Angle encoding is the second encoding, following amplitude encoding; we have experimented with SVM VRP simulation, which yields high accuracy and low error rates. Observing tables I and II, and 4.5 angle encoding is the second most precise encoding employed in our investigations. For fixed Hamiltonian simulations over 50 iterations with 6 qubits angle encoding, the first layer, including all optimizers, achieves 100 percent accuracy and zero percent error. In the 2nd layer simulation (over 50 iterations), the accuracy decreases to 98% for COBYLA, 96% for SLSQP, and 86% for L_BGFS_B, which is a greater decrease than the other two. These declines are attributable to optimizer-dependent statistical errors. Similarly, for 12 qubit simulations of SVM VRP, the accuracy rates are higher in the first layer, which consists of COBYLA at 100%, SLSQP at 92%, and L_BGFS_B at 88%, reiterating that the accuracy is highly dependent on the optimizer. As we move to the second layer of 12 qubit simulations on Fixed hamiltonian, we observe a decline in precision as the level of optimization rises. In this case, COBYLA winds up with 80%, L_BGFS_B with 70%, and SLSQP with 84%. Here, SLSQP's accuracy loss is less than that of the other two optimizers. The variable Hamiltonian with 12 qubits demonstrates a comparable trend. On the initial layer, we observe high accuracy with COBYLA at 96%, L_BGFS_B at 86%, and SLSQP at 90%. Moving to the second stratum, the accuracy figures drop significantly, with COBYLA at 76% and L_BGFS_B at 62%, while SLSQP maintains excellent accuracy at 86%. In every scenario of our investigation, it is evident that over-optimization reduces accuracy rates.

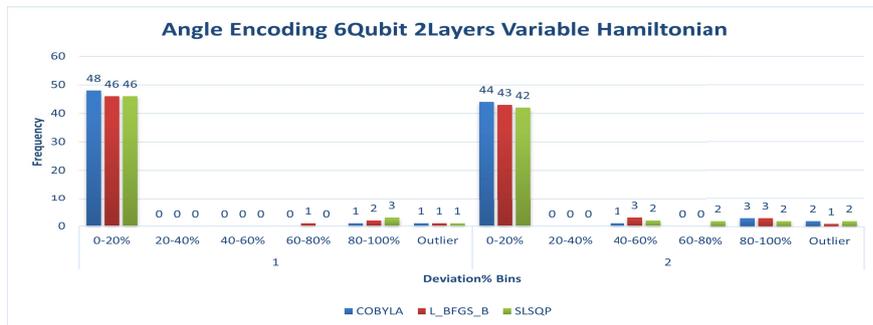
CHAPTER 4. SOLVING THE VEHICLE ROUTING PROBLEM VIA QUANTUM SUPPORT VECTOR MACHINES



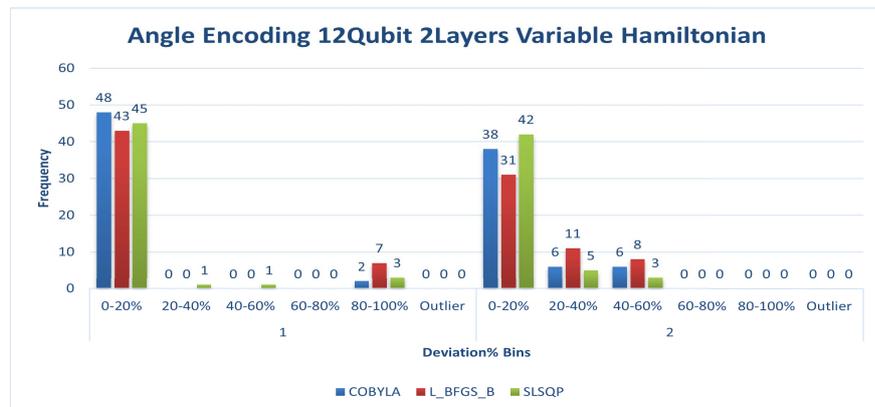
(a)



(b)



(c)



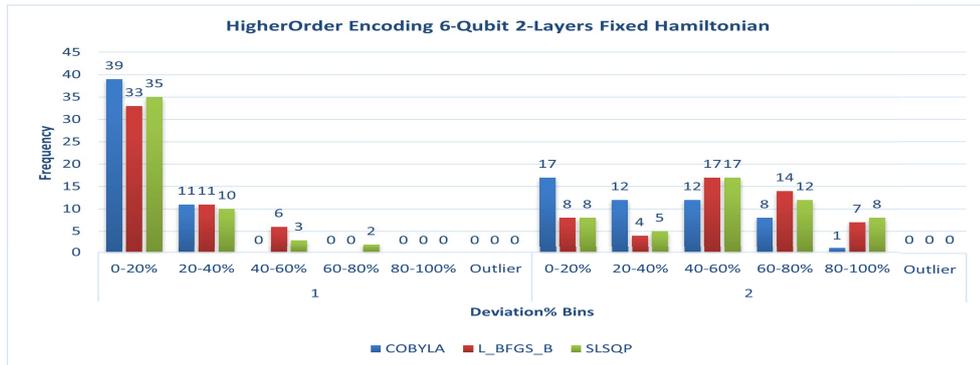
(d)

Figure 4.5: Plot illustrating angle encoding results for QSVM solution of VRP. (a) Angle encoding 6 qubits Fix hamiltonian, (b) Angle encoding 12 qubits Fix hamiltonian, (c) Angle encoding 6 qubits Variable hamiltonian, (d) Angle encoding 12 qubits Variable hamiltonian.

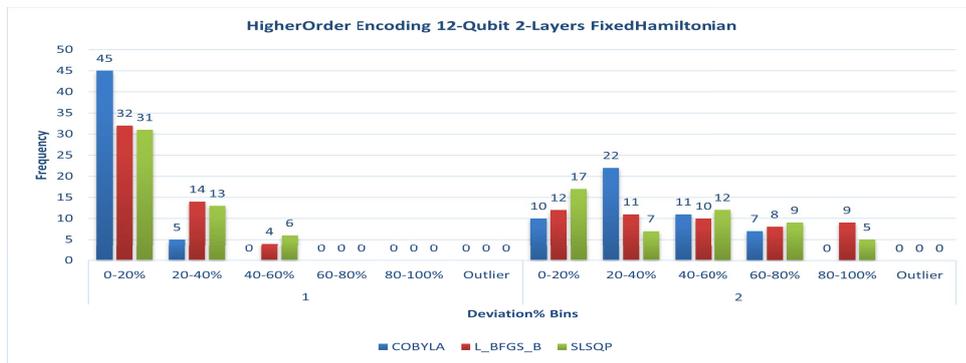
4.4.1.3 Higher Order Encoding

After Amplitude and Angle Encoding, Higher Order Encoding is the third most prevalent encoding in our SVM VRP simulation experiment. This is also the third most accurate encoding in our experiment. For both 6 qubit and 12 qubit simulations, HO encoding yields moderately accurate results; however, as the number of circuit layers is increased, the accuracy of the HO encoding scheme deteriorates, rendering it inappropriate. Figure 4.6 depicts the statistics of the HO encoding scheme for fixed and variable hamiltonian simulations of SVM VRP circuits over 50 iterations for both 6 qubit and 12 qubit simulations. COBYLA achieves 78% accuracy for a 6-qubit HO encoding circuit on a fixed Hamiltonian, while L_BGFS_B achieves 66% accuracy and SLSQP achieves 70% accuracy. As we proceed to the second layer, the accuracy considerably decreases, with COBYLA at 34% and SLSQP, L_BGFS_B at 16%, respectively. Similar trends can be observed in variable Hamiltonian simulations of HO encoding with 6 qubits, with COBYLA at 76%, SLSQP at 62%, and L_BGFS_B at 58% for the first layer; for the second layer, the accuracy drops to 36%, 34%, and 36% for COBYLA, L_BGFS_B, and SLSQP, respectively. The 12 qubit simulation yields superior results than the 6 qubit simulation and improves COBYLA's accuracy. For fixed hamiltonian simulations, COBYLA achieves an accuracy of 92%, compared to 78% for 6qubit. For variable hamiltonian simulations, COBYLA stores 76% for 6 qubit in the first layer, and 92% for 12 qubit in the first layer. The tendencies for L_BGFS_B and SLSQP are ambiguous for both cases (fixed and variable hamiltonian simulations); it is reassuring to conclude that an increase in layer decreases accuracy and that COBYLA outperforms the other two optimizers and ensures stable performance.

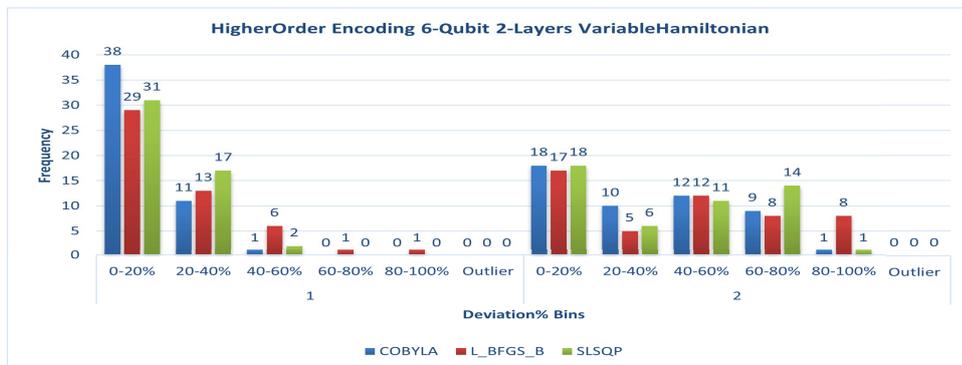
CHAPTER 4. SOLVING THE VEHICLE ROUTING PROBLEM VIA QUANTUM SUPPORT VECTOR MACHINES



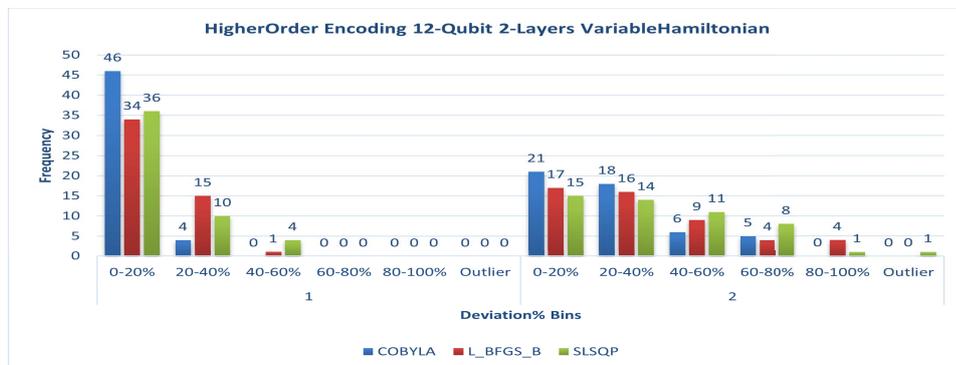
(a)



(b)



(c)



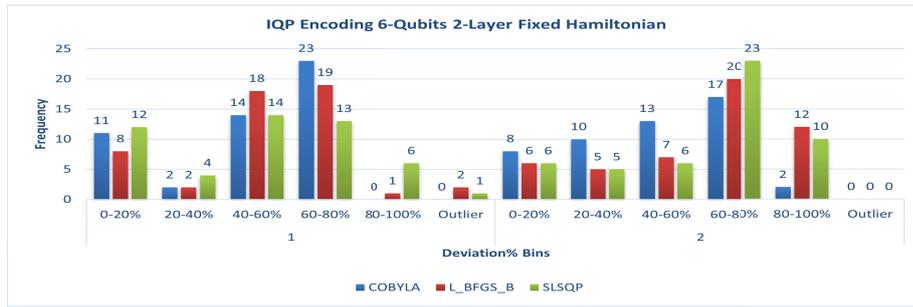
(d)

Figure 4.6: Plot illustrating Higherorder encoding results for QSVM solution of VRP. (a) Higherorder encoding 6 qubits Fix hamiltonian, (b) Higherorder encoding 12 qubits Fix hamiltonian, (c) Higherorder encoding 6 qubits Variable hamiltonian, (d) Higherorder encoding 12 qubits Variable hamiltonian.

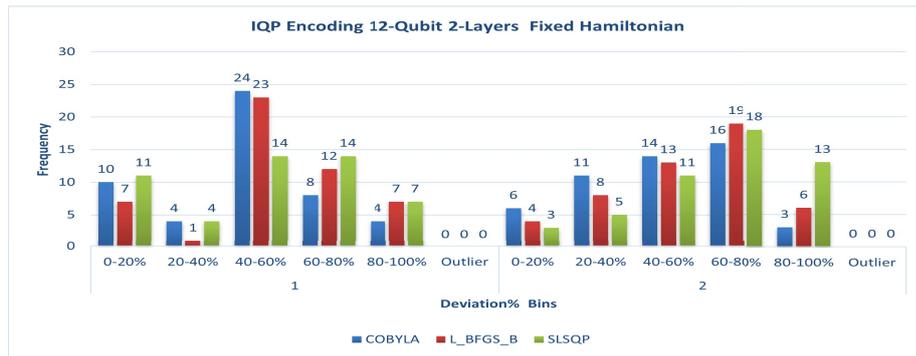
4.4.1.4 IQP Encoding

IQP encoding is the last and least accurate encoding in our experiment to simulate an SVM VRP circuit. The results are plotted in Figures 4.7 and in tables 4.1 and tables 4.2. As we can see from the figures and tables that accuracy is consistently poor for fixed and variable hamiltonian simulations in both 6 qubit and 12 qubit circuits. The accuracy further declines as layers increase. Hence this encoding is unsuitable in our experiment of SVM VRP circuits.

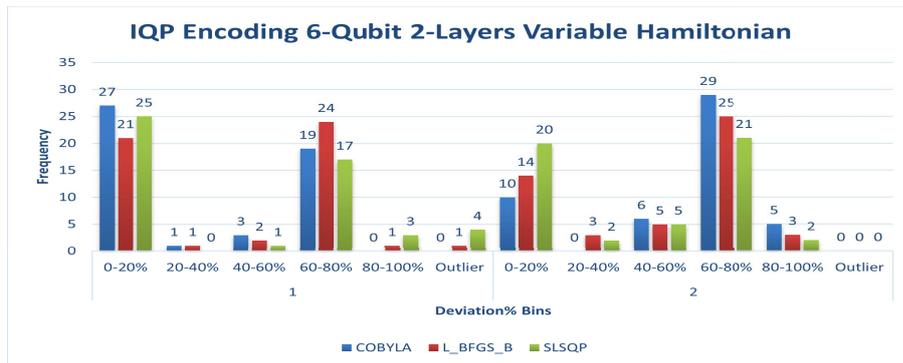
CHAPTER 4. SOLVING THE VEHICLE ROUTING PROBLEM VIA QUANTUM SUPPORT VECTOR MACHINES



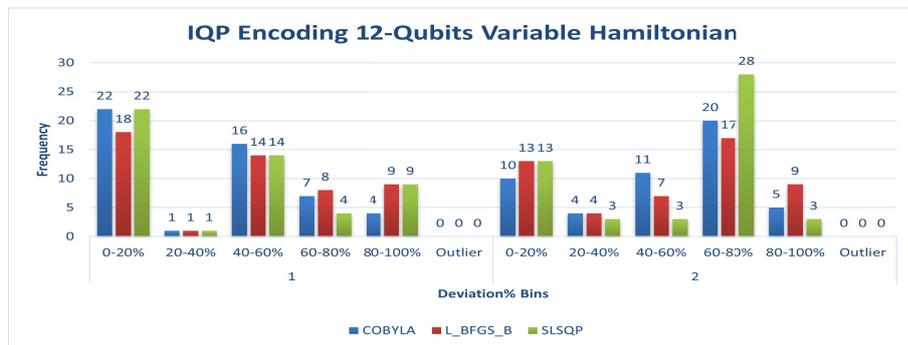
(a)



(b)



(c)



(d)

Figure 4.7: Plot illustrating IQP encoding results for QSVM solution of VRP. (a) IQP encoding 6 qubits Fix hamiltonian, (b) IQP encoding 12 qubits Fix hamiltonian, (c) IQP encoding 6 qubits Variable hamiltonian, (d) IQP encoding 12 qubits Variable hamiltonian.

Qubits				6				12			
Encoding	Optimizer	Layers	Iterations	No Devn.	With Devn	Acc	Err	No Devn.	With Devn	Acc	Err
Amplitude Enc.	COBYLA	1	50	50	0	100%	0%				
	L_BGFS_B	1	50	50	0	100%	0%				
	SLSQP	1	50	50	0	100%	0%				
	COBYLA	2	50	50	0	100%	0%				
	L_BGFS_B	2	50	50	0	100%	0%				
	SLSQP	2	50	50	0	100%	0%				
Angle Enc.	COBYLA	1	50	50	0	100%	0%	50	0	100%	0%
	L_BGFS_B	1	50	50	0	100%	0%	44	6	88%	12%
	SLSQP	1	50	50	0	100%	0%	46	4	92%	8%
	COBYLA	2	50	49	1	98%	2%	40	10	80%	20%
	L_BGFS_B	2	50	43	7	86%	14%	35	15	70%	30%
	SLSQP	2	50	48	2	96%	4%	42	8	84%	16%
HO Enc.	COBYLA	1	50	39	11	78%	22%	45	5	90%	10%
	L_BGFS_B	1	50	33	17	66%	34%	32	18	64%	36%
	SLSQP	1	50	35	15	70%	30%	31	19	62%	38%
	COBYLA	2	50	17	33	34%	66%	10	40	20%	80%
	L_BGFS_B	2	50	8	42	16%	84%	12	38	24%	76%
	SLSQP	2	50	8	42	16%	84%	17	33	34%	66%
IQP Enc.	COBYLA	1	50	11	39	22%	78%	10	40	20%	80%
	L_BGFS_B	1	50	8	42	16%	84%	7	43	14%	86%
	SLSQP	1	50	12	38	24%	76%	11	49	22%	98%
	COBYLA	2	50	8	42	16%	84%	6	44	12%	88%
	L_BGFS_B	2	50	6	44	12%	88%	4	46	8%	92%
	SLSQP	2	50	6	44	12%	88%	4	46	8%	92%

Table 4.1: For 6 and 12 qubit VRP circuits using SVM with 2 layers, the table above shows the Accuracy and Error with reference to classical minimum (over 50 iterations) for VQE simulations over a fixed Hamiltonian; utilizing Amplitude, Angle, Higher-Order, and IQP encoding schemes, Over the use of COBYLA, SLSQP and L_BGFS_B optimizers.

CHAPTER 4. SOLVING THE VEHICLE ROUTING PROBLEM VIA QUANTUM SUPPORT VECTOR MACHINES

Qubits				6				12			
Encoding	Optimizer	Layers	Iterations	No Devn.	With Devn	Acc	Err	No Devn.	With Devn	Acc	Err
Amplitude Enc.	COBYLA	1	50	48	2	96%	4%				
	L_BGFS_B	1	50	48	2	96%	4%				
	SLSQP	1	50	48	2	96%	4%				
	COBYLA	2	50	47	3	94%	6%				
	L_BGFS_B	2	50	47	3	94%	6%				
	SLSQP	2	50	47	3	94%	6%				
Angle Enc.	COBYLA	1	50	48	2	96%	4%	48	2	96%	4%
	L_BGFS_B	1	50	46	4	92%	8%	43	7	86%	14%
	SLSQP	1	50	46	4	92%	8%	45	5	90%	10%
	COBYLA	2	50	44	6	88%	12%	38	12	76%	24%
	L_BGFS_B	2	50	43	7	86%	14%	31	19	62%	38%
	SLSQP	2	50	42	8	84%	16%	42	8	84%	16%
HO Enc.	COBYLA	1	50	38	12	76%	24%	46	4	92%	8%
	L_BGFS_B	1	50	29	21	58%	42%	34	16	68%	32%
	SLSQP	1	50	31	19	62%	38%	36	14	72%	28%
	COBYLA	2	50	18	32	36%	64%	21	29	42%	58%
	L_BGFS_B	2	50	17	33	34%	66%	17	33	34%	66%
	SLSQP	2	50	18	32	36%	64%	15	35	30%	70%
IQP Enc.	COBYLA	1	50	27	23	54%	46%	22	28	44%	56%
	L_BGFS_B	1	50	21	29	42%	58%	18	32	36%	64%
	SLSQP	1	50	25	25	50%	50%	22	28	44%	56%
	COBYLA	2	50	10	40	20%	80%	10	40	20%	80%
	L_BGFS_B	2	50	14	36	28%	72%	13	37	26%	74%
	SLSQP	2	50	20	30	40%	60%	13	37	26%	74%

Table 4.2: For 6 and 12 qubit VRP circuits using SVM on with 2 layers, the table above shows the Accuracy and Error with reference to classical minimum (over 50 iterations) for VQE simulations on variable hamiltonians utilizing Amplitude, Angle, Higher-Order, and IQP encoding schemes, Over the use of COBYLA, SLSQP and L_BGFS_B optimizers.

4.4.2 Inferences from Simulation

As we scan through the results of SVM VRP simulations across the encoding schemes we observe some clear and distinct trends regarding the experiment. The tables 4.1 and 4.2 summarize the results obtained from the plots of all the encoding schemes used in this experiment. We list these trends as our outcomes of this experiment in the below points,

- The approach to solving VRP using machine learning is successful and is

capable of accomplishing the same or a superior result than the conventional approach using VQE and QAOA.

- The use of encoding/decoding schemes can serve the purpose of creating superposition and entanglement and eliminate the additional effort required to construct the mixer hamiltonian when solving the VRP using the standard approach of QAOA and VQE.
- While the standard approach to solving VRP or any combinatorial optimization problem requires a few layers of circuit depth (2 in most cases), we are able to achieve the same on the first layer itself with this approach, proving that it is more efficient than the standard approach.
- We also observe a distinct trend that as the number of layers increases, the accuracy decreases, which can be used to determine where to limit the optimization depth.
- Encoding/decoding schemes reduce the number of optimization layers but increase the circuit's complexity by introducing more gates. Therefore, when selecting an encoding scheme, we must take into account the complexity of the generated circuit and the number of required gates, as well as the number of classical resources (memory, CPU) it will require. There must be a trade-off between circuit complexity and the desired problem accuracy.
- Despite the fact that amplitude encoding provided the greatest accuracy, it could not be used to simulate a 12-qubit VRP scenario due to the large number of gates required. Angel encoding, on the other hand, was found to be much simpler due to a significantly smaller number of gates, as well as providing excellent accuracy (96% for COBYLA, and 92% for SLSQP and L_BGFS_B in variable hamiltonian simulation) across all the available optimizers. This again demonstrates that the complexity of circuits and the number of gates used are the most important considerations when choosing an encoding/decoding scheme.
- It can be noticed that AgE performs the best in terms of circuit complexity and accuracy rates due to the formation of a single layer of superposition. In

other encodings (HO, IqpE), we observe multi-layered complex superposition structures, which is the reason for fluctuations or error rates. Also in the fact that increasing layers also increases the superposition structures and therefore decreases the accuracy.

- Using COBYLA as an optimizer, HO encoding yielded intriguing results with reduced accuracy in circuits with fewer qubits (6 qubits) and higher accuracy in circuits with more qubits (12 qubits) for both fixed and variable hamiltonian simulations. The trend is disregarded by SLSQP and L_BGFS_B. This demonstrates that the algorithm's performance is extremely dependent on the optimizer; therefore, when evaluating the algorithm's performance, the most efficient optimizer should be selected by comparing the available optimizers.
- The IQP encoding scheme performed the worst in this experiment, with the lowest accuracy and highest error rates among all other encodings used for 1-layer, 2-layer, fixed, and variable Hamiltonians simulations. Therefore, the IqpE method cannot be used to solve VRP using QSVM.
- All of the optimizers used in the experiments performed well across AE, AgE, and HO encodings; however, COBYLA outperformed the other two due to its consistently high level of accuracy, but SLSQP is more resistant to accuracy fluctuations caused by an increase in optimization depth or in the presence of multi-layered circuits.

4.4.3 Complexity and Cost Considerations

Along with the inferences and observation, we want to touch upon the complexity of the quantum circuits particularly the number of qubits, quantum cost and circuit depth in this section. Here, we compare the SVM VRP solution with the standard VQE solution that is described in our previous work [60].

In order to compare we have listed the quantum cost and circuit depth of standard VQE implementation with that of SVM VRP implementation each consisting of 6 and 12 qubits with 1 and 2 layers (Table 4.3). We can broadly derive the following observations from Table 4.3.

- Transitioning from the conventional VQE method for VRP to using a Quantum Support Vector Machine results in a rise in the number of gates, elevating quantum depth and cost.
- The rise in quantum cost and depth is not linear and varies depending upon the encoding scheme employed when comparing different encoding schemes with the conventional VQE implementation of VRP.
- Increasing the number of layers and qubits in various techniques consistently leads to a rise in quantum depth and quantum cost. The extent of this increase varies based on the encoding method used.
- For the VQE standard, amplitude encoding, and angle encoding, there is a proportionate rise in depth and cost, roughly doubling as we go through layers and qubits.
- Higher order encoding (HO) and IQP encoding exhibit a much smaller rise in cost and depth over layers and qubits compared to Amplitude and Angle encoding.
- As the number of qubits increases from 6 to 12, the depth and cost rise by almost four times compared to the 6-qubit scenario; however, the increase is somewhat less than four times for HO and IQP encodings.
- Based on the depth and cost analysis, it is evident that HO and IQP provide lower costs and circuit depths compared to Amplitude and angle encoding, but their findings are less precise. Amplitude and angle encoding provide the highest level of precision. This helps in determining the balance between cost and accuracy when choosing encoding techniques for QSVM.

4.4.4 Experimental Setup, data gathering, and statistics

This experiment is conducted within the ambit of the QISKIT framework. While performing the experiment, we used a quantum instance object, and the ansatz runs inside the quantum instance object. A random seed is added to the quantum

Method	Layer	1		2	
	Qubit	depth	cost	depth	cost
VQE Standard	6	48	63	95	120
	12	201	234	401	456
Amplitude Encoding	6	400	693	800	1386
	12	NA	NA	NA	NA
Angel Encoding	6	58	73	116	146
	12	223	256	446	512
HO Encoding	6	73	113	95	145
	12	256	410	302	478
IQP Encoding	6	65	85	79	119
	12	236	280	262	350

Table 4.3: Table consisting of Quantum Depth and Quantum Cost of Various Encoding Schemes compared with Standard VQE implementation of VRP.

instance to stabilize VQE results. All the experiments have been run 50 + 50 times, one with a fixed Hamiltonian matrix and the other by varying the Hamiltonian matrix. The objective of the experiments is to ensure that the results of experiments are just not dependent on a single Hamiltonian. This is also to ensure that the used circuits achieve classical minimum or near classical minimum regardless of the hamiltonian used. Thus apart from the plots, the tables 4.1, 4.2 become the figure of merit. In addition to the many hours of testing and debugging, it is to be noted that the results reported here amounted to 150 hours of CPU time on a 24-core AMD workstation using Qiskit’s built-in simulators [50].

4.5 Conclusion

In this paper, we presented a novel technique for solving VRP through the use of a 6 and 12-qubit circuit-based quantum support vector machine (QSVM) with a variational quantum eigensolver for both fixed and variable Hamiltonians. In the experiment, multiple encoding strategies were used to convert the VRP formulation into a QSVM and solve it. In addition, we utilized multiple classical optimizers

available within the QISKIT framework to measure the output variation and accuracy rates. Consequently, our machine learning-based approach to resolving VRP has proven fruitful thus far. Using a QSVM to implement a gate-based simulation of a 3-city or 4-city VRP on a 6-qubit or 12-qubit system accomplishes the goal. The method not only resolves VRP, but also outperforms the conventional method of resolving VRP via multiple Optimization phases involving only VQE and QAOA. In addition, selecting appropriate encoding methods establishes the optimal balance between circuit complexity and optimization depth, thereby enabling multiple approaches to solve CO problems using machine learning techniques.

A QUANTUM APPROACH TO SYNTHETIC MINORITY OVERSAMPLING TECHNIQUE (SMOTE)

Abstract

The paper proposes the Quantum-SMOTE method, a novel solution that uses quantum computing techniques to solve the prevalent problem of class imbalance in machine learning datasets. Quantum-SMOTE, inspired by the Synthetic Minority Oversampling Technique (SMOTE), generates synthetic data points using quantum processes such as swap tests and quantum rotation. The process varies from the conventional SMOTE algorithm's usage of K-Nearest Neighbors (KNN) and Euclidean distances, enabling synthetic instances to be generated from minority class data points without relying on neighbor proximity. The algorithm asserts greater control over the synthetic data generation process by introducing hyperparameters such as rotation angle, minority percentage, and splitting factor, which allow for customization to specific dataset requirements. The approach is tested on a public dataset of TelecomChurn and evaluated alongside two prominent classification algorithms, Random Forest and Logistic Regression, to determine its impact along with varying proportions of synthetic data.

5.1 Introduction

5.1.1 Unbalanced Classification

Unbalanced classification is a prevalent problem in machine learning [70, 71], especially when the classes in a dataset are not represented evenly. Due to this imbalance, models may be biased towards the dominant class, frequently at the price of adequately forecasting the minority class. Such scenarios are common in real-world applications such as fraud detection in banking, insurance, and retail industries, detecting spam in email content, and predicting customer churn in Telecom, where the class of interest is usually underrepresented. To mitigate the problem of unbalanced classes, multiple techniques are used across industries, out of which Synthetic Minority Oversampling Techniques (SMOTE) [72, 73] are quite popular.

5.1.2 Overview of SMOTE

SMOTE is a statistical method used to augment the number of instances in a dataset in a balanced manner. The technique was first presented by Chawla et al. [73], whose main objective is to tackle the issue of imbalanced datasets, namely in the realm of classification. Imbalanced datasets are common in many real-world circumstances, where the frequency of instances belonging to a certain class is much lower than the others. The disparity may result in unsatisfactory performance of classification models since they have a tendency to exhibit bias towards the dominant class. SMOTE resolves this problem by generating artificial samples from the underrepresented class.

5.1.3 Existing works on SMOTE

During our study and implementation of the SMOTE technique and its modifications, we have come across academic papers authored by other researchers that explore the progress and real-world uses of this algorithm [74–76]. Research on the incorporation of SMOTE into ensemble learning approaches has been a substantial focus. The combination seeks to use the advantages of both techniques in order

to enhance the classification performance on datasets with uneven distribution. The use of SMOTEBoost [77], and RusBoost highlights the significance of SMOTE in ensemble learning techniques. Moreover, current research is underway in the domain of image classification with a specific emphasis on the use of SMOTE [78].

5.1.4 Purpose and Scope

Since SMOTE is a widely used technique in machine learning to address unbalanced classification, we believe that a quantum computing approach will greatly enhance its efficiency in quantum machine learning applications. Since quantum computing is greatly useful in problems related to high dimensional datasets, A SMOTE algorithm in quantum machine Learning will be of significant value. In this paper, we propose a novel method of generating synthetic data points by using the quantum swap test and quantum rotations, which can be used to increase the number of minority class representatives in a large dataset and help reduce bias in classification models. We have also applied the method to a publicly available dataset named Telco Customer Churn [79] used for telecom churn classification and recorded the results.

5.1.5 Organization

The paper is structured in the following manner. Section 5.2 explores the core mathematical principles, including the Basic Concept, several versions of the SMOTE algorithm, and the K-Means Clustering technique. Section 5.3 presents an examination of the development of SMOTE utilizing quantum techniques, namely the use of the swap test and rotation principles. This is followed by analyzing the outcomes obtained by applying these concepts to actual data. Section 5.4 involves the application of the quantum SMOTE algorithm to a real-world dataset. This process comprises data preparation, clustering, and the production of synthetic data using the SMOTE method. We utilize the SMOTE technique on the telecom data, varying the proportions of the minority class to 30%, 40%, and 50%, respectively. In Section 5.4.2, we provide a summary of the results and model parameters of the classification Models, which elucidate the effects of Quantum SMOTE.

5.2 Background

5.2.1 Basic Concept of SMOTE

SMOTE was proposed way back in 2002 by Chawla et al. [73] as a way to address issues with unbalanced classification. The primary objective of the SMOTE algorithm is to generate Synthetic data points from minority classes using K Nearest neighbors and Euclidean distances. The synthetic data points, in turn, increase the population of the minority class in the population, which counters the bias towards the majority class in a classification scenario. SMOTE is widely used and accepted, and since then, multiple variants of SMOTE have been proposed by various researchers. In the below subsections, we will cover the working of the SMOTE algorithm and its Variants.

5.2.2 How SMOTE Works

SMOTE [73] is an over-sampling technique that addresses imbalanced datasets by generating synthetic instances for the minority class instead of just duplicating existing examples. To address the imbalance in class distribution, the minority class is augmented by generating synthetic samples along the line segments connecting the K nearest neighbors of each minority class sample. Neighbors are randomly selected from the K nearest neighbors, based on the desired level of over-sampling. The initial approach used a set of five closest neighbors. For example, when the required over-sampling quantity is 300%, only three neighbors are selected from the five nearest neighbors, and one sample is created in the direction of each selected neighbor.

Synthetic samples are produced as follows:

1. Find the feature vector's closest neighbor and compute the difference between the two.
2. Pick a uniformly random number between 0 and 1 and multiply it by this difference.
3. Add the resulting number to the original feature vector.

The result is the random creation of a synthetic point along the line segment between two feature vectors. This method broadens the minority group's density and resolves the decision boundary.

Algorithm 2 SMOTE(N, A, m)

```

1: Input:
2:  $N$  = number of samples in the minority class.
3:  $A$  = the percentage of SMOTE to be applied.
4:  $m$  = number of nearest neighbors to be considered.
5: Output:
6: Generate  $(N/100) \times A$  artificial samples for the minority class.
7: procedure SMOTE( $N, A, m$ )
8:   if Proportion of class  $A < 100\%$  then
9:     Randomly choose a percentage of the minority class samples to be
     SMOTEd.
10:   end if
11:   if  $A < 100$  then
12:      $N \leftarrow (A/100) \times N$ 
13:      $A \leftarrow 100$ 
14:   end if
15:    $A \leftarrow \text{int}(A/100)$ 
16:    $\text{numattrs} \leftarrow$  total count of attributes
17:    $\text{Sample}[][] \leftarrow$  array containing the original minority class samples
18:    $\text{newindex} \leftarrow 0$ 
19:    $\text{Synthetic}[][] \leftarrow$  array for creating artificial samples
20:   for  $i = 1$  to  $N$  do
21:     Compute  $m$  closest neighbours for  $i$  and save indices in  $\text{nnarray}$ 
22:     Fill array  $A$  with values from  $\text{nnarray}$  starting at index  $i$ 
23:   end for
24:   POPULATE( $A, i, \text{nnarray}$ )
25:   while  $A \neq 0$  do
26:     Select a random integer from 1 to  $m$  as  $nn$ 
27:     for  $\text{attr} = 1$  to  $\text{numattrs}$  do
28:        $\text{diff} \leftarrow \text{Sample}[\text{nnarray}[\text{nn}]][\text{attr}] - \text{Sample}[i][\text{attr}]$ 
29:        $\text{gap} \leftarrow$  random number between 0 and 1
30:        $\text{Synthetic}[\text{newindex}][\text{attr}] \leftarrow \text{Sample}[i][\text{attr}] + \text{gap} \times \text{diff}$ 
31:     end for
32:      $\text{newindex} \leftarrow \text{newindex} + 1$ 
33:      $A \leftarrow A - 1$ 
34:   end while
35:   return "End of Populate"
36: end procedure

```

5.2.3 Variants of SMOTE

As the SMOTE algorithm became popular, multiple variations have been proposed. For the sake of reference, we mention some of them in this section.

Borderline-SMOTE:

Borderline-SMOTE specifically targets the minority class samples that are in close proximity to the boundary with the majority class. The objective is to produce artificial samples at close proximity to the boundary rather than over the whole of the distribution of the minority class [80].

ADASYN (Adaptive Synthetic Sampling):

ADASYN specifically aims to generate synthetic samples for the minority class. However, unlike SMOTE, ADASYN adjusts its approach based on the unique properties of the dataset. It produces additional synthetic data for minority class samples that are more challenging to learn (i.e., those that are incorrectly categorized using the K-Nearest Neighbor method) in contrast to those that are less difficult. The number of artificial samples to be generated for each underrepresented sample is contingent upon the complexity of learning that particular sample [81].

SMOTE-ENN (SMOTE with Edited Nearest Neighbors):

SMOTE-ENN [82] is a hybrid technique that integrates the concepts of oversampling and under-sampling to tackle the problem of class imbalance in machine learning. The SMOTE method is used to oversample the minority class, whereas the ENN rule is used for undersampling. SMOTE algorithm creates new samples in the minority class by selecting the K-nearest neighbors from the same class and creating interpolations between the original sample and its neighbors.

Each instance in the dataset undergoes testing by comparing it with its three closest neighbors. If the majority of the neighbors do not have the same class as the instance, the instance is removed. This mostly pertains to the dominant class within skewed datasets.

The implementation of SMOTE-ENN involves the following steps:

Initial Step: Utilise SMOTE technique to oversample the minority class and generate synthetic instances, hence achieving a balanced distribution of classes.

Next, implement the ENN rule on the dataset that has been oversampled. ENN

will exclude instances from both the majority and minority classes that are deemed to be noisy or are located on the boundary between the two classes.

Result: This integrated method not only resolves the disparity by augmenting the number of instances in the underrepresented category but also enhances the dataset's quality, resulting in a more distinct and less susceptible decision border between the classes, reducing overfitting. This helps in cleaning the space between the majority and minority classes.

SMOTE-Tomek Links:

SMOTE-Tomek Links is a hybrid method that combines the Synthetic Minority Over-sampling Technique (SMOTE) with Tomek Links, an under-sampling technique. This combination is used to mitigate class imbalance in machine learning datasets. A pair of examples belonging to contrasting classes are classified as a Tomek Link if they are the closest neighbors of each other. Essentially, they are closely related points, but belong to separate classes. The objective is to eliminate these Tomek Links in order to enhance the clarity of the class boundaries. Usually, the instance belonging to the majority class in each pair of Tomek Links is eliminated, which helps in minimising the overlap between classes [73, 83].

SVMSMOTE: SVMSMOTE (Support Vector Machine Synthetic Minority Over-sampling Technique) [84] integrates ideas from Support Vector Machines (SVMs) into SMOTE. SVMSMOTE uses SVMs to detect support vectors among the samples of the minority class. Support vectors are often defined as the data points that are in close proximity to the decision border separating different classes. Within the framework of class imbalance, these minority class samples are often the most crucial ones to prioritize for over-sampling. SVMSMOTE creates synthetic samples in the proximity of the detected support vectors rather than distributing them randomly throughout the whole space of the minority class. The objective of this strategy is to enhance the decision border region where the classifier is prone to uncertainty.

5.2.4 K-Means Clustering

K-means clustering [?] is a widely used unsupervised machine learning approach that divides a dataset into K separate and non-overlapping groups. The main

goal of the K-means algorithm is to categorize data points into clusters, where each point is assigned to the cluster with the closest average value, which acts as the center or centroid of the cluster. The technique sequentially allocates data points to the centroid that is closest to them and updates the locations of the centroids by calculating the mean of the points in each cluster. This procedure iterates until convergence, which is achieved when the locations of the centroids no longer exhibit substantial changes or when a predetermined number of iterations is reached. The k-means algorithm is very susceptible to the starting position of centroids, which might result in convergence to local optima. Therefore, it is crucial to do numerous iterations of the algorithm with various initializations to ensure accurate results. Although K-means is computationally fast and easy to implement, its main strengths lie in its ability to uncover patterns in data, cluster comparable observations, and assist in exploratory data analysis in many domains, such as picture segmentation, customer segmentation, and pattern identification.

5.2.5 ROC Curve

The Receiver Operating Characteristic (ROC) is a commonly used graphical plot to assess the effectiveness of a binary classifier system while the discrimination threshold is adjusted. It is especially advantageous in scenarios where there is a requirement to strike a balance between a true positive rate and a false positive rate.

The True Positive Rate (TPR), often referred to as Sensitivity, Recall, or Probability of Detection, is determined by the formula $TPR = TP/(TP + FN)$, where TP represents the count of true positives and FN represents the count of false negatives. The False Positive Rate (FPR), often referred to as the Probability of False Alarm, is determined by the formula $FPR = FP/(FP + TN)$, where FP represents the count of false positives and TN represents the count of true negatives.

An ROC curve illustrates the relationship between the true positive rate (TPR) and the false positive rate (FPR) across different threshold values. The x -axis corresponds to the False Positive Rate, while the y -axis corresponds to the True Positive Rate.

The AUC, or Area Under the Curve, is a metric that quantifies a classifier's

capacity to differentiate between classes. It serves as a concise representation of the ROC curve. A model with a higher AUC value indicates superior performance.

5.3 Emulating SMOTE Using Quantum

Upon examining the SMOTE algorithm and its modifications as presented by [73], we have adopted a distinct method for oversampling the minority class by using quantum approaches. It is often seen in real-world datasets that the minority class is unevenly distributed in the population. Therefore, producing synthetic data uniformly throughout all distribution zones may not effectively address the issue of bias. Our method entails dynamically segmenting the whole population using clustering methods and thereafter creating synthetic data inside each cluster to achieve the desired minority proportion. The target minority percentage is the overall percentage of minorities in the population following the introduction of synthetic data.

Synthetic data creation requires using quantum rotation to manipulate individual data points from the minority class. This is done by representing each data point as a multidimensional vector and rotating it along the X (or Y or Z) direction. In the next part, we will get into the specifics of selecting X rotations. The rotation angle is computed as the angle formed between the vector of the minority data point and the centroid vector of the cluster it belongs to. It is important to mention that while determining the angle slice, a relatively tiny angle is used to reduce sudden departures from the initial minority class data point. If there are many synthetic data points to be created, the remaining synthetic data points are obtained by incrementing the angle from the starting value.

The objective of this strategy is to ensure that the created synthetic data points remain within the statistical bounds of their respective cluster, while also boosting the density of the minority class. In the following sections, we will provide a comprehensive analysis of the algorithm, rotation, and data creation process.

The figures 5.1 illustrate fundamental differences in Classical and Quantum SMOTE procedures

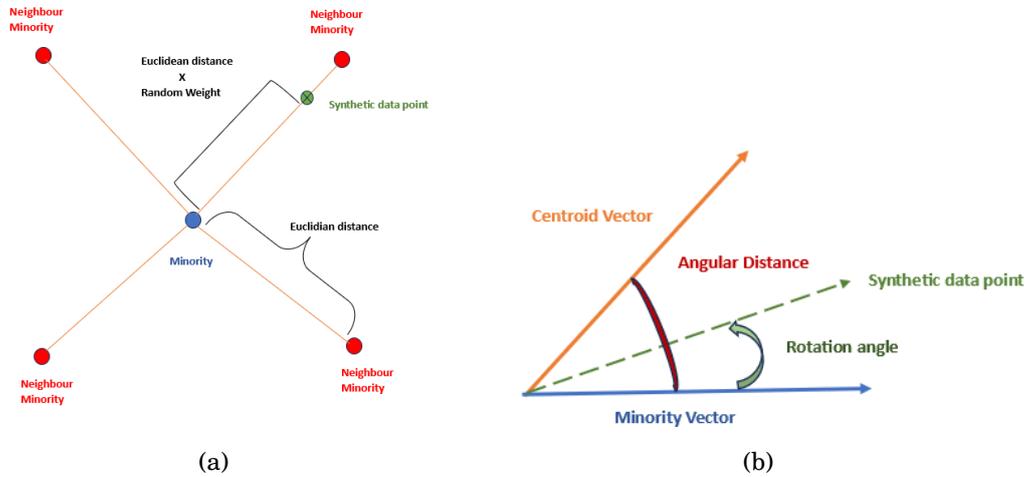


Figure 5.1: Plot illustrating different SMOTE mechanisms. (a) Classical SMOTE, (b) Quantum SMOTE.

5.3.1 Swap Test

The quantum swap test is a quantum procedure used to ascertain the degree of similarity between two quantum states, ψ and ϕ . The test result quantifies the degree of overlap between the two states, which is directly linked to their inner product $\langle\psi|\phi\rangle$. Usually, we tackle the swap test in the following manner.

Setup: Commence by using a control qubit, normally in the state $|0\rangle$, together with two quantum registers that are in the respective states ψ and ϕ .

Hadamard Transformation: Perform a Hadamard gate operation on the control qubit. This results in the creation of a superposition state, where the control qubit is in a state that is proportional to the sum of $|0\rangle$ and $|1\rangle$.

Controlled Swap: Execute a regulated exchange (or Fredkin gate) using the control qubit. When the control qubit is in the state $|1\rangle$, it performs a swap operation on the two quantum registers. Alternatively, it does not alter them.

Second Hadamard: Apply a second Hadamard gate to the control qubit.

Measurement: Conduct a measurement on the control qubit. If the two quantum states $|\psi\rangle$ and $|\phi\rangle$ are indistinguishable, the control qubit will consistently be seen in the state $|0\rangle$. The likelihood of seeing the state $|0\rangle$ diminishes as the states grow more different.

Outcome: The chance of seeing the control qubit in the state $|0\rangle$ after the swap test provides information on the similarity of the two quantum states. More precisely, the likelihood is proportional to the square of the magnitude of their inner product. The mathematical expression for the probability $P(0)$ of measuring the state $|0\rangle$ is,

$$(5.1) \quad P_0 = \frac{1}{2}(1 + |\langle\psi|\phi\rangle|^2).$$

From this above expression, $\langle\psi|\phi\rangle$ can be determined as,

$$(5.2) \quad \langle\psi|\phi\rangle = \sqrt{2P_0 - 1}$$

Fig. 5.2 circuit illustrates the basic swap test.

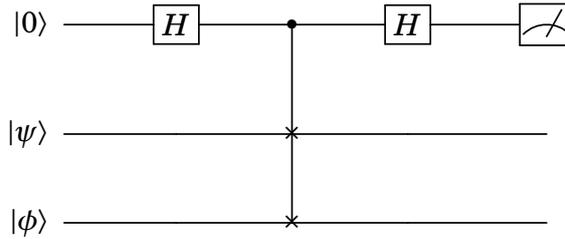


Figure 5.2: Swap test circuit.

The swap test probability can be defined as,

$$(5.3) \quad \text{swap_test_probability} = 1 - 2p_0 + p_1$$

where p_0 and p_1 are probabilities of the states $|0\rangle$ and $|1\rangle$ respectively.

5.3.1.1 Compact Swaptest

For the purpose of this paper, we have adopted a modified version of the swap test to find the inner product of our two vectors, namely the centroid and an arbitrary minority data point within the cluster. The procedure is already discussed in the articles [85, 86]. Though the article describes the procedure as a dissimilarity measure and uses it to calculate Euclidian distance, we have used it to calculate the

inner product of quantum states and thereby the angular distance. The advantage of this procedure is that it requires less number of qubits

$$n = \log_2(M) + 1$$

where n is the number of qubits and M is the classical data encoded by amplitude embedding. The procedure is as follows,

We amplitude encode two vectors C (Centroid) and M (Minority) by

$$(5.4) \quad C \longrightarrow |C\rangle = \frac{1}{|C|} \sum_i C_i |q_i\rangle$$

$$(5.5) \quad M \longrightarrow |M\rangle = \frac{1}{|M|} \sum_i M_i |q_i\rangle$$

We define the quantum states $|\psi\rangle$ and $|\phi\rangle$ as:

$$(5.6) \quad \begin{aligned} |\psi\rangle &= \frac{|0\rangle \otimes |C\rangle + |1\rangle \otimes |M\rangle}{\sqrt{2}} \\ |\phi\rangle &= \frac{|C||0\rangle - |M||1\rangle}{\sqrt{Z}} \\ Z &= |C|^2 + |M|^2 \end{aligned}$$

Let's divulge into the details of this circuit

$$(5.7) \quad \begin{aligned} &|0\rangle|\phi\rangle|\psi\rangle \\ &= |+\rangle \left(\frac{(C|0\rangle - M|1\rangle)}{\sqrt{Z}} \right) \left(\frac{|0\rangle|C\rangle + |1\rangle|M\rangle}{\sqrt{2}} \right) \\ &= \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}} \right) \left(\frac{C|0\rangle - M|1\rangle}{\sqrt{Z}} \right) \left(\frac{|0\rangle|C\rangle + |1\rangle|M\rangle}{\sqrt{2}} \right) \\ &= \frac{1}{2\sqrt{Z}} [|0\rangle(C|0\rangle - M|1\rangle)(|0\rangle|C\rangle + |1\rangle|M\rangle) \\ &\quad + |1\rangle(C|0\rangle - M|1\rangle)(|0\rangle|C\rangle + |1\rangle|M\rangle)] \\ &= \frac{1}{2\sqrt{Z}} [|0\rangle(C|0\rangle|0\rangle|C\rangle + C|0\rangle|1\rangle|M\rangle - M|1\rangle|0\rangle|C\rangle - M|1\rangle|1\rangle|M\rangle) \\ &\quad + |1\rangle(C|0\rangle|0\rangle|C\rangle + C|0\rangle|1\rangle|M\rangle - M|1\rangle|0\rangle|C\rangle - M|1\rangle|1\rangle|M\rangle)] \end{aligned}$$

Applying a controlled swap operation

$$\begin{aligned}
 &= \frac{1}{2\sqrt{Z}}[|0\rangle(C|0\rangle|0\rangle|C\rangle + C|0\rangle|1\rangle|M\rangle - M|1\rangle|0\rangle|C\rangle - M|1\rangle|1\rangle|M\rangle) \\
 (5.8) \quad &+ |1\rangle(C|0\rangle|0\rangle|C\rangle + C|1\rangle|0\rangle|M\rangle - M|0\rangle|1\rangle|C\rangle - M|1\rangle|1\rangle|M\rangle)]
 \end{aligned}$$

Applying Hadamard

$$\begin{aligned}
 &= \frac{1}{2\sqrt{Z}}[|+\rangle(C|0\rangle|0\rangle|C\rangle + C|0\rangle|1\rangle|M\rangle - M|1\rangle|0\rangle|C\rangle - M|1\rangle|1\rangle|M\rangle) \\
 &+ |-\rangle(C|0\rangle|0\rangle|C\rangle + C|1\rangle|0\rangle|M\rangle - M|0\rangle|1\rangle|C\rangle - M|1\rangle|1\rangle|M\rangle)] \\
 &= \frac{1}{2\sqrt{2Z}}[(|0\rangle + |1\rangle)(C|0\rangle|0\rangle|C\rangle + C|0\rangle|1\rangle|M\rangle - M|1\rangle|0\rangle|C\rangle - M|1\rangle|1\rangle|M\rangle) \\
 &+ (|0\rangle - |1\rangle)(C|0\rangle|0\rangle|C\rangle + C|1\rangle|0\rangle|M\rangle - M|0\rangle|1\rangle|C\rangle - M|1\rangle|1\rangle|M\rangle)] \\
 &= \frac{1}{2\sqrt{2Z}}[|0\rangle(2C|0\rangle|0\rangle|C\rangle + (C|0\rangle|1\rangle|M\rangle - M|0\rangle|1\rangle|C\rangle) + (C|1\rangle|0\rangle|M\rangle \\
 &- M|1\rangle|0\rangle|C\rangle) - 2M|1\rangle|1\rangle|M\rangle) \\
 (5.9) \quad &+ |1\rangle(C|0\rangle|1\rangle|M\rangle + M|0\rangle|1\rangle|C\rangle - M|1\rangle|0\rangle|C\rangle - C|1\rangle|0\rangle|M\rangle)]
 \end{aligned}$$

The probability of 0 can be calculated as,

$$\begin{aligned}
 P_0 &= \frac{1}{8Z} |(2C|0\rangle|0\rangle|C\rangle + (C|0\rangle|1\rangle|M\rangle - M|0\rangle|1\rangle|C\rangle) \\
 &\quad + (C|1\rangle|0\rangle|M\rangle - M|1\rangle|0\rangle|C\rangle) - 2M|1\rangle|1\rangle|M\rangle|^2 \\
 &= \frac{1}{8Z} |(2C|0\rangle|0\rangle|C\rangle + |0\rangle|1\rangle(C|M\rangle - M|C\rangle) \\
 &\quad + |1\rangle|0\rangle(C|M\rangle - M|C\rangle) - 2M|1\rangle|1\rangle|M\rangle|^2 \\
 &= \frac{1}{8Z} |(2C|0\rangle|0\rangle|C\rangle + (|0\rangle|1\rangle + |1\rangle|0\rangle)(C|M\rangle - M|C\rangle) - 2M|1\rangle|1\rangle|M\rangle|^2 \\
 &= \frac{1}{8Z} |(2C|0\rangle|0\rangle|C\rangle - 2M|1\rangle|1\rangle|M\rangle) + (|0\rangle|1\rangle + |1\rangle|0\rangle)(C|M\rangle - M|C\rangle)|^2 \\
 &= \frac{1}{8Z} (|2C|0\rangle|0\rangle|C\rangle - 2M|1\rangle|1\rangle|M\rangle|^2 + ||0\rangle|1\rangle + |1\rangle|0\rangle|^2 |C|M\rangle - M|C\rangle|^2) \\
 &= \frac{1}{8Z} (4C^2 + 4M^2 + ||0\rangle|1\rangle + |1\rangle|0\rangle|^2 |C|M\rangle - M|C\rangle|^2) \\
 &= \frac{1}{8Z} (4Z + 2(C^2 + M^2 - 2CM \langle M|C\rangle)) \\
 &= \frac{1}{8Z} (4Z + 2(Z - 2CM \langle M|C\rangle)) \\
 &= \frac{1}{4Z} (2Z + (Z - 2CM \langle M|C\rangle)) \\
 &= \frac{1}{4Z} (3Z - 2CM \langle M|C\rangle) \\
 \Rightarrow \langle M|C\rangle &= \frac{(3 - 4P_0)Z}{2CM}
 \end{aligned}
 \tag{5.10}$$

The above equation 5.10 states that after measurement, from the probability of 0, we obtain the inner product between the centroid and minority. In a slightly different perspective, let us calculate the inner product of ψ and ϕ ,

$$\langle \phi | \psi \rangle = \left(\frac{\langle C| \otimes \langle 0| - \langle M| \otimes \langle 1|}{\sqrt{Z}} \right) \left(\frac{|0\rangle \otimes |C\rangle + |1\rangle \otimes |M\rangle}{\sqrt{2}} \right)
 \tag{5.11}$$

Expanding the inner product:

$$\begin{aligned}
 \langle \phi | \psi \rangle &= \frac{1}{\sqrt{Z}} \frac{1}{\sqrt{2}} (\langle C| \otimes \langle 0| (|0\rangle \otimes |C\rangle) + \langle C| \otimes \langle 0| (|1\rangle \otimes |M\rangle)) \\
 &\quad - \langle M| \otimes \langle 1| (|0\rangle \otimes |C\rangle) - \langle M| \otimes \langle 1| (|1\rangle \otimes |M\rangle))
 \end{aligned}
 \tag{5.12}$$

Simplifying each term:

$$\begin{aligned}
 \langle C | \otimes \langle 0 | (|0\rangle \otimes |C\rangle) &= \langle C | C \rangle \otimes \langle 0 | 0 \rangle = |C|^2 \\
 \langle C | \otimes \langle 0 | (|1\rangle \otimes |M\rangle) &= 0 \\
 \langle M | \otimes \langle 1 | (|0\rangle \otimes |C\rangle) &= 0 \\
 (5.13) \quad \langle M | \otimes \langle 1 | (|1\rangle \otimes |M\rangle) &= \langle M | M \rangle \otimes \langle 1 | 1 \rangle = |M|^2
 \end{aligned}$$

So, the inner product simplifies to:

$$\begin{aligned}
 \langle \phi | \psi \rangle &= \frac{1}{\sqrt{Z}} \frac{1}{\sqrt{2}} (|C|^2 - |M|^2) \\
 (5.14) \quad \langle \phi | \psi \rangle &= \frac{|C|^2 - |M|^2}{\sqrt{2Z}}
 \end{aligned}$$

Calculating $|\langle \phi | \psi \rangle|^2$:

$$(5.15) \quad |\langle \phi | \psi \rangle|^2 = \left(\frac{|C|^2 - |M|^2}{\sqrt{2Z}} \right)^2 = \frac{(|C|^2 - |M|^2)^2}{2Z}$$

$$(5.16) \quad 2Z|\langle \phi | \psi \rangle|^2 = 2Z \left(\frac{(|C|^2 - |M|^2)^2}{2Z} \right)$$

simplifying:

$$(5.17) \quad 2Z|\langle \phi | \psi \rangle|^2 = (|C|^2 - |M|^2)^2$$

Assuming

$$\begin{aligned}
 2Z|\langle \phi | \psi \rangle|^2 &= D^2 \\
 (5.18) \quad \implies D^2 &= 2Z|\langle \phi | \psi \rangle|^2
 \end{aligned}$$

The term D refers to the Euclidean distance [86], and the inner product of $\langle \phi | \psi \rangle$ represents the swapest probability.

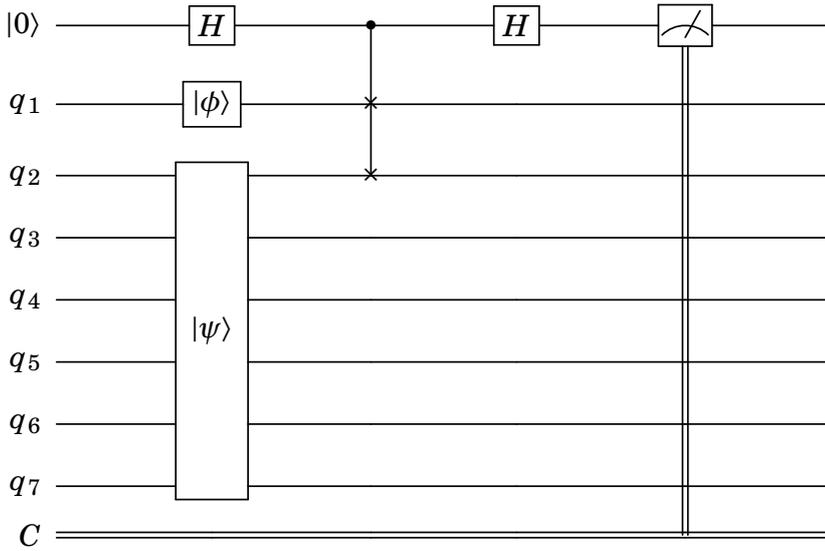


Figure 5.3: Compact Swap test circuit.

Based on this, we define the angle between two vectors or angular distance as

$$(5.19) \quad \text{angular_distance} = 2 \cos^{-1}(\sqrt{\text{swap_test_probability}})$$

The above angular distance or the angle between two vectors will be used to rotate the minority class data point, which we will describe subsequently.

5.3.2 Applying Rotation to data point

After calculating the angle (angular distance) between two vectors, we rotate the actual minority data point by an angle less than the calculated angle to create a synthetic data point. We choose to minimize the angle of rotation to prevent abrupt fluctuation of values in the minority data point. We perform rotations along the X, Y, and Z axes to analyze their impact on the minority data points.

As the angle of rotation is minimal for the minority data point vector, we have derived the angle of rotation with the below logic.

split_factor is the factor by which we want to divide the generated angle, we have experimented with 2, 5, 10 and 100 for various rotations mentioned above and will outline the result of rotation for a sample containing 10 data points.

Algorithm 3 Angle of rotation calculation logic

```

sf: split_factor
if angular_distance >  $\frac{\pi}{2}$  then
    angle  $\leftarrow \left| \frac{\pi}{2} - \text{angular\_distance} \right| / \text{sf}$ 
else if angular_distance < 0 then
    angle  $\leftarrow \left| \left( \frac{\pi}{2} - \text{angular\_distance} \right) \times \text{random}(0.5, 1) \right| / \text{sf}$ 
else
    angle  $\leftarrow \text{random}(0, \text{angular\_distance}) / \text{sf}$ 
end if

```

The aforementioned figures illustrate the influence of rotation on data points. Initially, we selected a subset of data points from the minority class and visually represented them in Figure 5.4. This figure displays a scatter plot of the data points. Subsequently, we have performed X, Y, and Z rotations on these data points, using the *split_factor* as a basis. We conducted experiments by incrementally increasing the split factor and evaluating the resulting effects on rotations.

X Rotation: X Rotation refers to the rotation of a data point in relation to the X axis. We conducted experiments with split factors of 2, 5, and 10. Upon increasing the split factor from 2 to 10, we see that the synthetic data points created by each split factor exhibit a greater proximity to the original data points. When the split factor 2 is used for X rotation, the resulting data points are located at a certain distance from the original location. As we go from 5 to 10, the freshly created data points get increasingly closer together. At 10, the synthetic data point is the closest among the three dividing factors.

Y Rotation: Y Rotation refers to the rotational movement of data points around the Y axis. From the analysis of figure 5.6, it is evident that the newly created data points exhibit a high sensitivity to Y rotations. Additionally, these data points need the generation of extremely tiny angles in order to be positioned in close proximity to the Source(the minority sample). It is evident that as the splitting factor (100) increases, resulting in extremely tiny angles, the created data point is closest to the source. Conversely, small splitting factors (5) yield data points that deviate significantly from the nature of the data point sample.

Z Rotation: Z rotation refers to the rotation of data points around the Z axis. Based on the evidence shown in Figure 5.9, we can confidently infer that the

behavior of Z rotation is similar to that of X rotation. Additionally, it is evident that using splitting factors of 5 and 10 results in the generation of additional data points that are in close proximity to the source.

In general, it can be confidently said that all rotations have the ability to generate synthetic data points. However, the Y rotation is more sensitive, but the X and Z rotations provide similar outcomes.

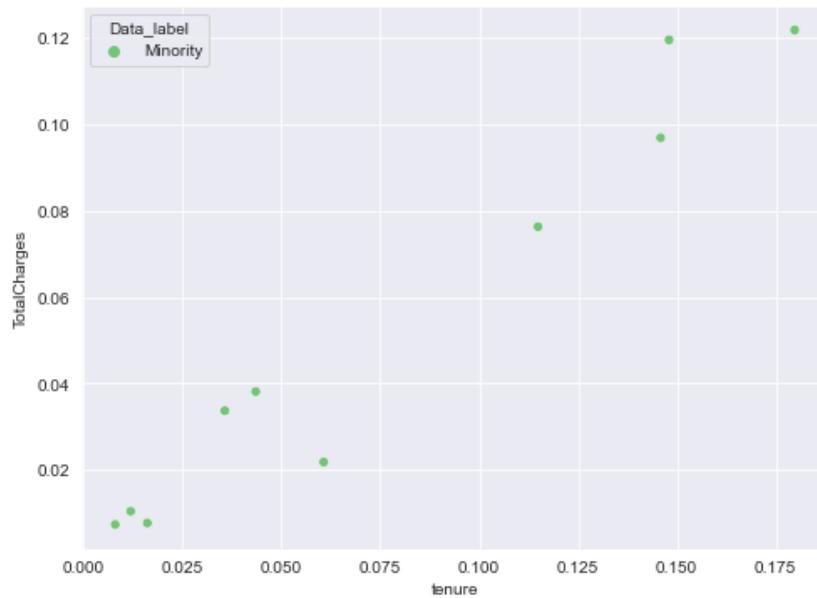
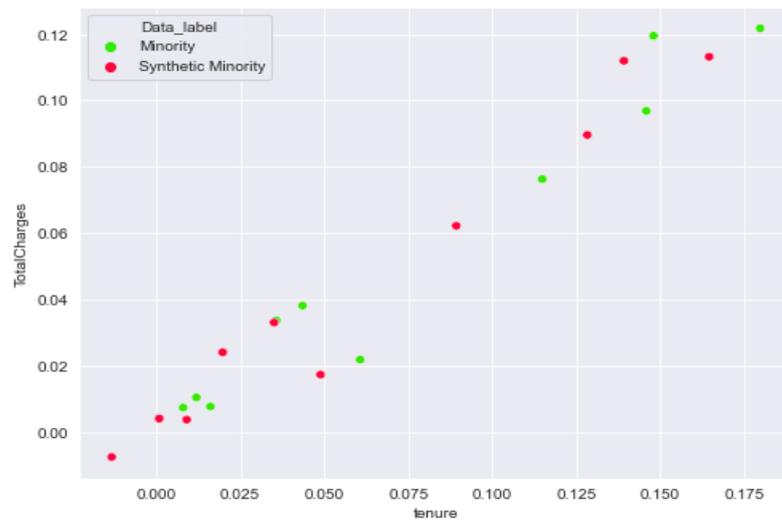
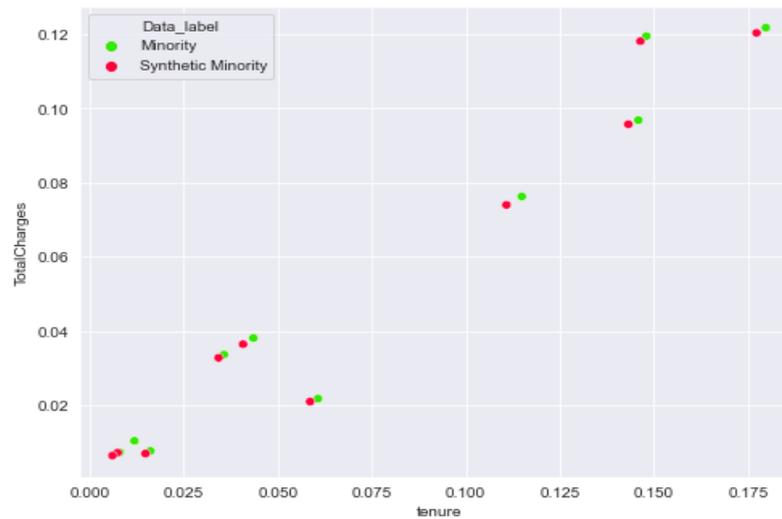


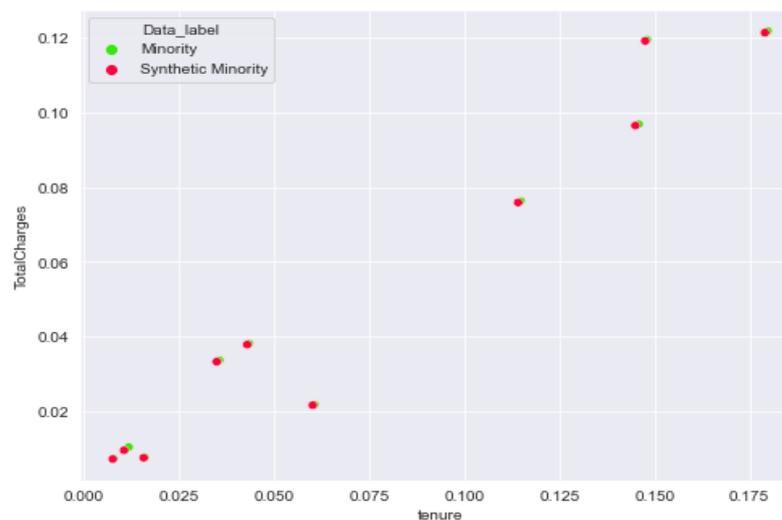
Figure 5.4: Plot representing Sample data points of Minority class from population without any rotation



(a)

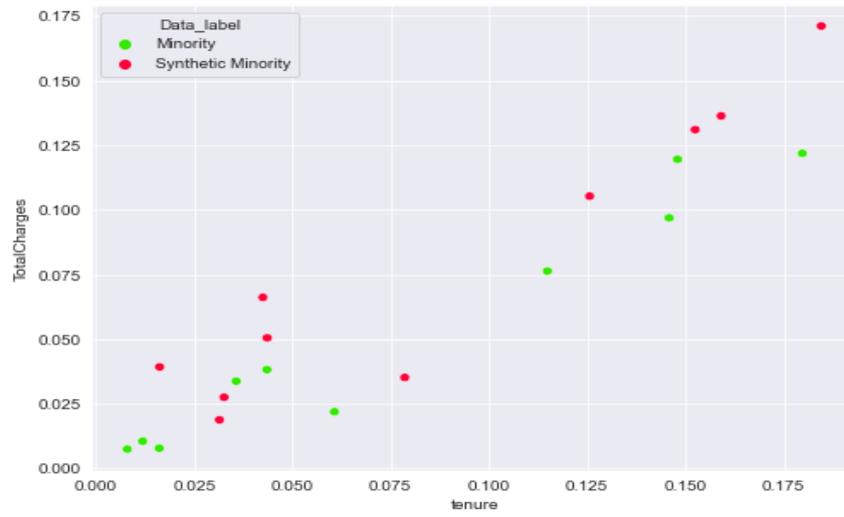


(b)

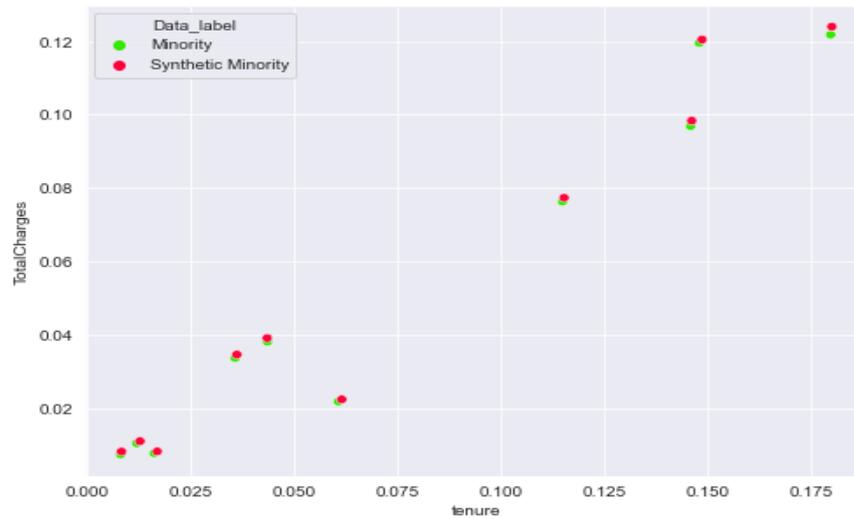


(c)

Figure 5.5: Plot illustrating impact of X Rotation on Sample data points of Minority class. (a) X Rotation with $split_factor = 2$, (b) X Rotation with $split_factor = 5$, (c) X Rotation with $split_factor = 10$.

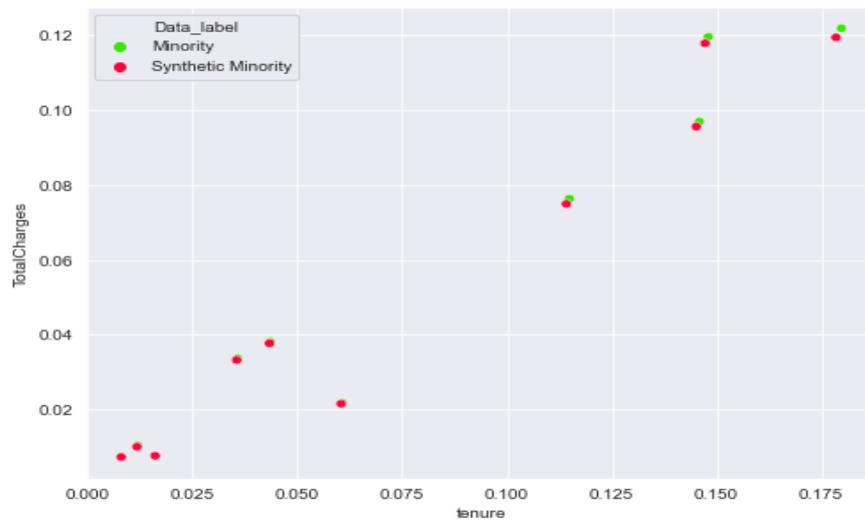


(a)

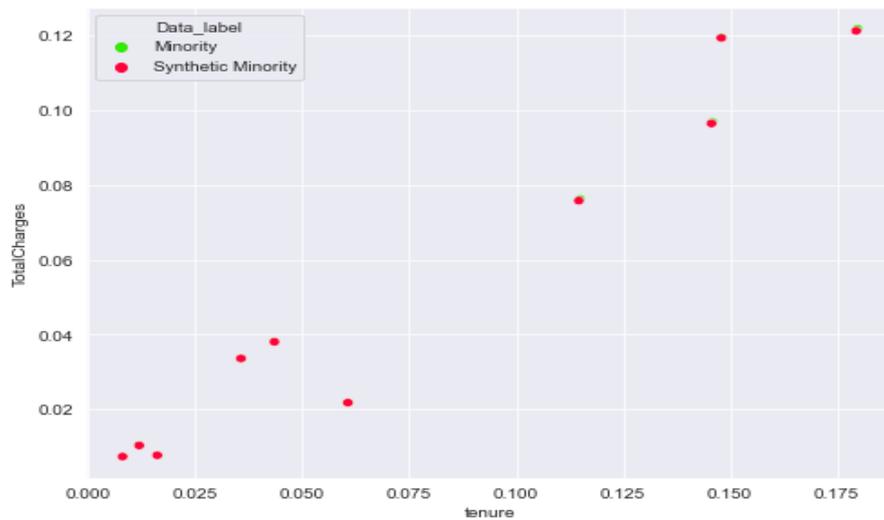


(b)

Figure 5.6: Plot illustrating impact of Y Rotation on Sample data points of Minority class. (a) Y Rotation with $split_factor = 5$, (b) Y Rotation with $split_factor = 100$



(a)



(b)

Figure 5.7: Plot illustrating impact of Z Rotation on Sample data points of Minority class. (a) Z Rotation with $split_factor = 5$, (b) Z Rotation with $split_factor = 10$

5.3.3 Quantum SMOTE Algorithm

We now introduce QuantumSMOTE. Broadly, our algorithm proceeds in two steps: clustering of the population and generating synthetic data points by the swap test and rotation of minority class data points. We believe clustering is an essential

pre-step to synthetic data generation. Though we can use any clustering method that produces clusters in data, we have used K-Means Clustering in our research with a minimum of 3 clusters, and we recommend the same for further research on this topic.

Post clustering, we proceed with synthetic data generation, and for the purpose of simplicity, we name this part the QuantumSMOTE function. The pseudocode of this is described in the section below. Generally, it comprises four distinct parts: Data preparation for the swap test, application of the swap test, rotation of synthetic data points, and generation of synthetic data points for each cluster based on the target.

Algorithm 4 Preparation for Swap Test

```

1: function PREPSWAP TEST(data_point1, data_point2)
2:   norm_data_point1  $\leftarrow$  0
3:   norm_data_point2  $\leftarrow$  0
4:   Dist  $\leftarrow$  0
5:   for i  $\leftarrow$  0 to length(data_point1) - 1 do
6:     norm_data_point1  $\leftarrow$  norm_data_point1 + data_point1[i]2
7:     norm_data_point2  $\leftarrow$  norm_data_point2 + data_point2[i]2
8:     Dist  $\leftarrow$  Dist + (data_point1[i] + data_point2[i])2
9:   end for
10:  Dist  $\leftarrow$   $\sqrt{Dist}$ 
11:  data_point1_norm  $\leftarrow$   $\sqrt{norm\_data\_point1}$ 
12:  data_point2_norm  $\leftarrow$   $\sqrt{norm\_data\_point2}$ 
13:  Z  $\leftarrow$  round(data_point1_norm2 + data_point2_norm2)
14:   $\phi$   $\leftarrow$  [round(data_point1_norm/ $\sqrt{Z}$ , 3), -round(data_point2_norm/ $\sqrt{Z}$ , 3)]
15:  Initialize array  $\psi$ 
16:  for i  $\leftarrow$  0 to length(data_point1) - 1 do
17:     $\psi$ .append(round(data_point1[i]/(data_point1_norm  $\times$   $\sqrt{2}$ ), 3))
18:     $\psi$ .append(round(data_point2[i]/(data_point2_norm  $\times$   $\sqrt{2}$ ), 3))
19:  end for
20:  return  $\phi$ ,  $\psi$ 
21: end function

```

Algorithm 5 Swap Test

```

1: function SWAP TESTV1( $\psi, \phi$ )
2:   Initialize Quantum Register  $q1$  with 1 qubit
3:   Initialize Quantum Register  $q2$  with  $n+2$  qubits
4:   Initialize Classical Register  $c$  with 1 bit
5:   Create Quantum Circuit with  $q1, q2,$  and  $c$ 
   States initialization
6:   Initialize  $q2[0]$  with state  $\phi$ 
7:   Initialize  $q2[1 : n + 2]$  with state  $\psi$ 
   The swap test operator
8:   Apply Pauli-X Gate to  $q2[1]$ 
   Swap Test
9:   Apply Hadamard Gate to  $q1[0]$ 
10:  Apply Controlled SWAP Gate on  $q1[0], q2[0],$  and  $q2[1]$ 
11:  Apply Hadamard Gate to  $q1[0]$ 
12:  Measure  $q1$  into classical register  $c$ 
   Simulation and result collection
13:  Set up quantum simulator
14:  Execute the quantum circuit on the simulator
15:  Collect the result into a variable  $result$ 
16:  Extract measurement counts from  $result$ 
   Calculate the Swap Test probability
17:   $p0 \leftarrow \frac{\text{counts.get('0', 0)}}{\text{total\_shots}}$ 
18:   $p1 \leftarrow \frac{\text{counts.get('1', 0)}}{\text{total\_shots}}$ 
19:   $swap\_test\_probability \leftarrow 1 - 2 \times p0 + p1$ 
20:  Print  $swap\_test\_probability$ 
   Calculate the angular distance
21:   $angular\_distance \leftarrow 2 \times \arccos(\sqrt{swap\_test\_probability})$ 
22:  Print  $angular\_distance$ 
23:  return  $swap\_test\_probability, angular\_distance$ 
24: end function

```

Algorithm 6 Normalize Array

```
1: function NORMALIZEARRAY(arr) Calculate the sum of squares of the  
   elements in the array  
2:   sum_of_squares  $\leftarrow$  SUMOFSQUARES(arr)  
   Check if the sum of squares is already very close to 1  
3:   if ISCLOSE(sum_of_squares, 1.0, rtol =  $1e - 6$ ) then  
4:     return arr  
5:   end if  
   Calculate the scaling factor to make the sum of squares equal to 1  
6:   scaling_factor  $\leftarrow 1.0/\sqrt{\text{sum\_of\_squares}}$   
   Normalize the array by multiplying each element by the scaling  
   factor  
7:   normalized_arr  $\leftarrow arr \times scaling\_factor$   
8:   return normalized_arr  
9: end function
```

Algorithm 7 Create Synthetic Data

```
1:  $ad$  : angular_distance
2:  $sf$  : split_factor
3: function CREATESYNDATA( $n$ ,  $loop\_ctr$ ,  $angle\_increment$ ,  $ad, sf$ ,
    $data\_point1$ ,  $data\_point2$ )
4:    $data\_point1 \leftarrow$  NORMALIZEARRAY( $data\_point1$ )
5:    $data\_point2 \leftarrow$  NORMALIZEARRAY( $data\_point2$ )
6:   Initialize Quantum Circuit  $circuit$  with  $n$  qubits
7:    $circuit.INITIALIZE(data\_point1)$ 
8:   if  $ad > \frac{\pi}{2}$  then
9:      $angle \leftarrow \left| \frac{\pi}{2} - angular\_distance \right| / sf$ 
10:  else if  $ad < 0$  then
11:     $angle \leftarrow \left| \frac{\pi}{2} - ad \right| \times \text{RANDOMUNIFORM}(0.5, 1) / sf$ 
12:  else
13:     $angle \leftarrow \text{RANDOMUNIFORM}(0, ad) / sf$ 
14:  end if
15:  Print "rotation angle",  $angle$ 
16:   $angle \leftarrow angle + angle\_increment$ 
17:  for  $l \leftarrow 0$  to  $n - 1$  do
18:    Apply RX gate to  $circuit$  at qubit  $l$  with angle  $angle$ 
19:  end for
   Simulate the quantum circuit
20:  Set up quantum simulator
21:  Execute  $circuit$  on the simulator and store result in  $job$ 
22:   $result \leftarrow job.result()$ 
23:   $statevector \leftarrow result.get\_statevector()$ 
   Extract the final data point from the statevector
24:   $new\_data\_point \leftarrow \text{REAL}(statevector)$ 
25:  return  $new\_data\_point$ 
26: end function
```

Algorithm 8 Quantum Synthetic Minority Over-sampling Technique

```

1: function QUANTUMSMOTE(Data, Target_pct, cluster_centroids)
2:   Initialize an empty DataFrame syn_dataframe
3:   Set target_synthetic_percent to 30
4:   for each cluster clus_idx in centroid_df do
5:     Calculate minority_count_in_cluster and total_count_in_cluster
6:     minority_percent ← Calculate minority percentage
7:     Print minority percentage for cluster clus_idx
8:     Calculate required synthetic iterations: synthetic_loop_itr
9:     Print "Needed synthetic datapoint iterations =", synthetic_loop_itr
10:    if synthetic_loop_itr in (0,1) then
11:      Set synthetic_loop_itr1 ← 1
12:    else if synthetic_loop_itr > 1 then
13:      synthetic_loop_itr1 ← CEIL(synthetic_loop_itr)
14:      fraction_part ← synthetic_loop_itr1 -
      FLOOR(synthetic_loop_itr)
15:    else
16:      synthetic_loop_itr1 ← -1
17:    end if
18:    if synthetic_loop_itr1 ≥ 0 then
19:      for syn_loop ← 0 to synthetic_loop_itr1 - 1 do
20:        Choose centroid_temp and minority_temp appropriately
21:        Flatten centroid_temp to centroid_dp_tmp
22:        for each minority_dp_temp in minority_temp do
23:          Prepare phi and psi for swap test
24:          Calculate probabilities and angles for swap test
25:          Calculate number of iterations and angle increment
26:          Generate synthetic data and create DataFrame syn_df_temp
27:          Append syn_df_temp to syn_dataframe
28:        end for
29:      end for
30:    else if synthetic_loop_itr1 < 0 then
31:      Print "Cluster clus_idx has sufficient minority percentage."
32:    else
33:      Print "No processing required."
34:    end if
35:  end for
36:  return syn_dataframe
37: end function

```

5.4 Case Study and Results

To test the QuantumSMOTE algorithm, we analyse the publicly available dataset of telecom churn [79]. This dataset is widely used to experiment and test various models for customer retention and is quite useful in comparing classical models with the models post-induction of synthetic data by the quantum SMOTE algorithm. In the following subsections, we will describe data behavior, data preparation for modeling, and applying QuantumSMOTE to the data.

5.4.1 Improving Telecom Churn Prediction Using SMOTE

The telecom churn dataset is purposefully developed to predict customer behavior and help in generating customer retention programs. Each row in the dataset represents an individual consumer, with each column representing different attributes of these customers. Notably, the dataset has such characteristics as:

Churn Indicator: This column identifies customers who have terminated their service during the previous month.

Subscribed Services: A detailed list of all services that each customer has signed up for, such as phone service, multiple lines, internet, online security, online backup, device protection, tech support, and streaming TV and movies.

Account Information: comprises of how long they have been a client for, the terms of the contract they entered into with their company, and which method they would prefer to use when making payments so as to keep track of their spending habits effectively through electronic means like electronic mail that may save on transaction costs like envelope usage, monthly expenditure and cumulative costs incurred so far.

Demographic Information: It provides information about the customer's gender, age group, whether or not they are married, and whether they have dependent children.

5.4.1.1 Preparing Data For Quantum SMOTE

The Telco churn dataset is amenable to a usual data preparation process, which broadly includes the following phases.

Missing Value Tearment: Inspect the telco churn dataset for null values and adapt a strategy to handle them. Since we found a very small percentage of records (11, to be precise) that have missing values across multiple columns, we proceeded with dropping them.

Removing Irrelevant Data: Identify and remove any columns that are not relevant to churn prediction, such as customer IDs that are unique and not predictive of churn.

Data Type Conversion: To ensure that each column is of the appropriate data type, we have converted multiple columns with text data as to category. These included columns such as PhoneService, MultipleLines, InternetService, OnlineSecurity, OnlineBackup, DeviceProtection, TechSupport, StreamingTV, StreamingMovies, Contract, PaperlessBilling, PaymentMethod, gender, SeniorCitizen, Partner and Dependents. We also converted the numerical columns such as TotalCharges, tenure and MonthlyCharges to float to avoid any of them being treated as text due to import issues.

Exploratory Data Analysis (EDA): EDA was performed to understand the distribution and relationship of variables. We applied various univariate, and bivariate analyses to understand the behavior of data, particularly numeric variables, which are essential for creating models. The variables TotalCharges, tenure, and MonthlyCharges are particularly important since the distribution of these variables later will be used to verify the effect of the SMOTE procedure.

Label Encoding: For the sake of better visualization and correlation analysis with the target variable, we performed label encoding of multiple categorical variables.

Correlation Analysis: We conducted a correlation analysis of numerical variables to eliminate multicollinearity. Also, we conducted a correlation analysis of all the variables with the target to select the best-fit variables for modeling. Post correlation analysis, we are able to drop multiple variables that are not relevant for the purpose of modeling.

Onehot Encoding: Post selection of features, we converted all the categorical variables to Onehot encoding, thereby creating multiple numerical columns for each categorical value.

Feature Scaling: Since Onehot encoding created multiple numerical columns

with values 0 and 1, the continuous variables such as TotalCharges, tenure and MonthlyCharges are scaled by minmax scaling to lie between 0 and 1.

5.4.1.2 Clustering

As we have indicated earlier, the Quantum SMOTE algorithm relies on unique customer segments to calculate the angle between the segment centroid (mean) and minority data point; we have used the K-Means clustering method approach to derive segments. The approach for identifying inherent groupings among customers is based on their attributes, which can further assist in understanding customer behavior and improving retention strategies. For the sake of our experiment, we have identified 3 clusters using the K-Means approach to generate new data and highlight the achievements. The outcome of the clustering approach is at least 3 clusters (datasets that are dynamically segmented) with different majority-minority populations. These are useful when deriving angles based on which minority population across clusters will be most valuable for the SMOTE algorithm.

5.4.1.3 Quantum SMOTE and Synthetic Data

After applying the Clustering algorithm to the Telecom Churn dataset and processing the data, we proceeded to apply the Quantum SMOTE Algorithm (8) to each cluster. The goal was to enhance the representation of the minority population to a certain percentage of the overall dataset. The procedure used two primary approaches previously mentioned, namely the swap test (Algo. 5) and rotation (Algo. 7).

Swap Test: The fundamental operation of the swap test has been previously explained in the preceding sections. We use the swap test in a modified manner to compute the angular distance between the vector representing the minority data point and the vector representing the centroid. The procedure is effectively executed in Ref. [85]. The swap test requires two inputs, denoted as ϕ and ψ . The state ϕ is determined by computing the norms of the inputs, which consist of the centroid and minority data points. On the other hand, the state ψ is obtained by concatenating the normalized components of the inputs. The execution of this

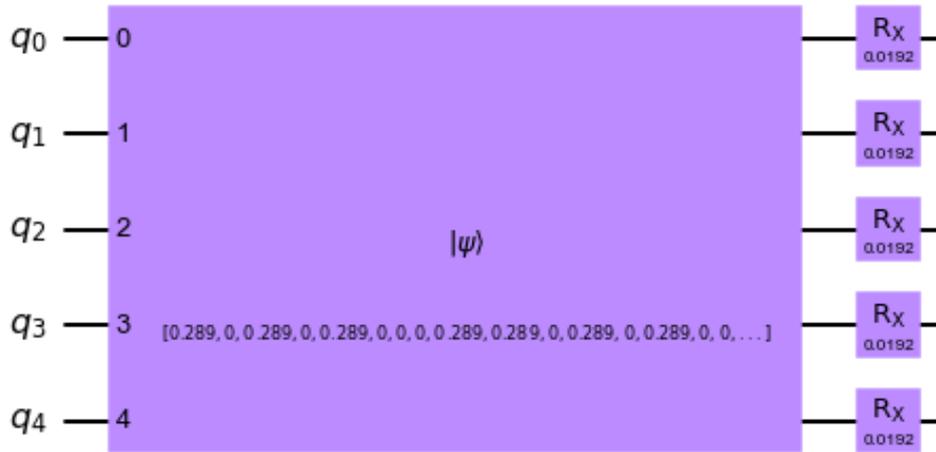


Figure 5.8: Data point rotation circuit.

preparation is shown in the auxiliary function 4. The circuit that is obtained is rendered in Fig. 5.3.

The main purpose of using this technique to swap test is to minimize the required number of qubits in constructing the swap test circuit, which becomes particularly advantageous as the dataset dimension expands. After performing feature selection and Onehot encoding, we obtained a final count of 32 columns. Consequently, our swap test circuit necessitates the use of 8 qubits and a classical register. Nevertheless, using a traditional methodology may have resulted in the use of 65 qubits. The swap test circuit facilitates the calculation of the angular distance between the cluster centroid and the minority data point.

Rotation: After performing the swap test, it is necessary to rotate the minority data point by an angle that represents a minute fraction of the total angular distance. The rotation circuit executes the rotation of the normalized minority data point vector. In the preceding section 5.3.2, we have provided a detailed explanation of the different rotations of X, Y, and Z. In this experiment, we applied X rotations to all of the minority data points. To account for numerous interactions or repeated rotations of a single minority data point, we have adjusted the rotation angle by 0.0174, which corresponds to the conversion from radians to degrees. We are attempting to adjust the angle of the minority data point using angular degrees,

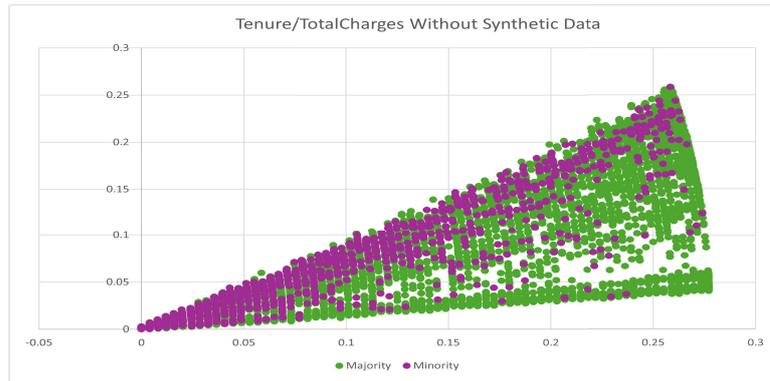
even though the angular distance generated by the swap test is in radians. The rotation circuit comprises the state vector of the normalized data point and rotation gates (Fig. 5.8). By rotating minority data points, synthetic data points that closely resemble the original data points are created, thanks to the use of modest rotation angles. When the synthetic data points are included in the original dataset, it leads to an increase in the total density of the minority class. The scatter distribution of synthetic data points in the population is shown in Fig. 5.9. The data illustrates the distribution of classes (majority, minority, and synthetic minority) as the proportion of the minority class increases from 30% to 50%.

5.4.1.4 Observation from generated data

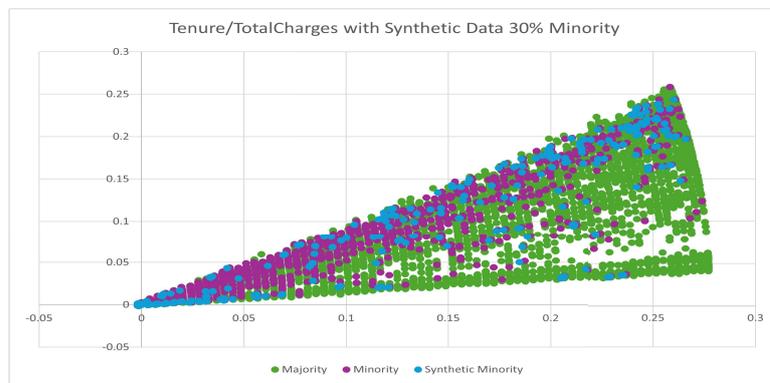
Following the generation of synthetic data points by rotation, our next step is to examine the general distribution of important variables throughout the whole population. The objective is to assess if the introduction of artificial data points has caused any significant statistical deviation in the distribution of the variable. The figures 5.10, 5.11, 5.12, and 5.13 illustrate the distribution of three important variables in the dataset: Tenure, MonthlyCharges, and Total charges. The distribution before the induction of synthetic data points is shown in Fig. 5.10. The distributions following the induction of synthetic data points, resulting in total minority percentages of 30, 40, and 50, are shown in figures 5.11, 5.12, and 5.13 accordingly. After applying SMOTE, we can confidently state that there is a little distortion to the distribution of variables, but the bins have increased in size. The use of relatively modest angles during rotation prevents any significant deformation to the geometry of the distribution. By comparing the charts depicting the variables after using the SMOTE technique, we see a progressive rise in the values within each category, ranging from 30% to 50%. This confirms the successful use of the SMOTE method.

5.4.1.5 Applying Classification Models

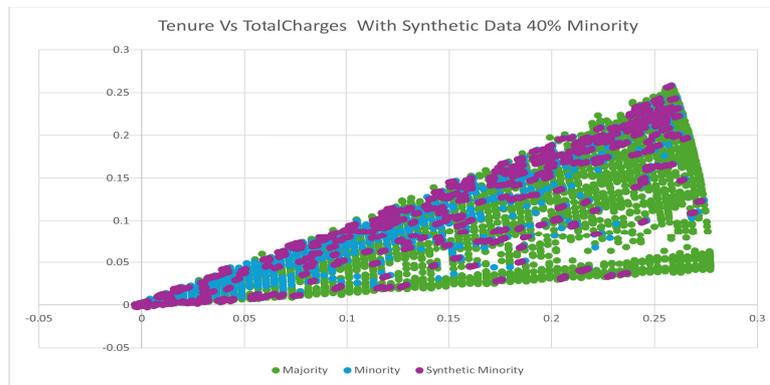
In order to comprehensively evaluate the effectiveness of the Synthetic Minority Over-sampling Technique (SMOTE) in addressing class imbalances, our research used two classification models, namely Random Forest and Logistic Regression,



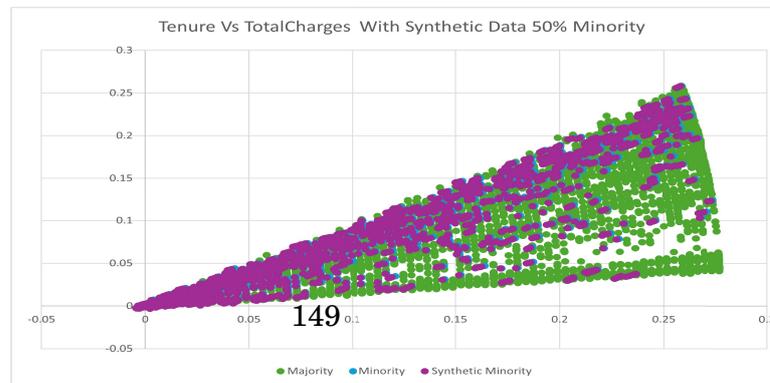
(a) data points with no synthetic



(b) 30% synthetic



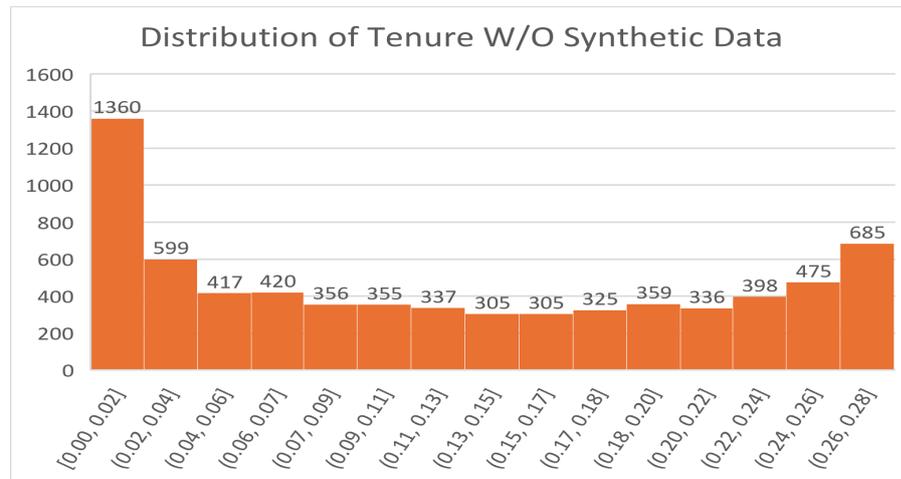
(c) 40% synthetic



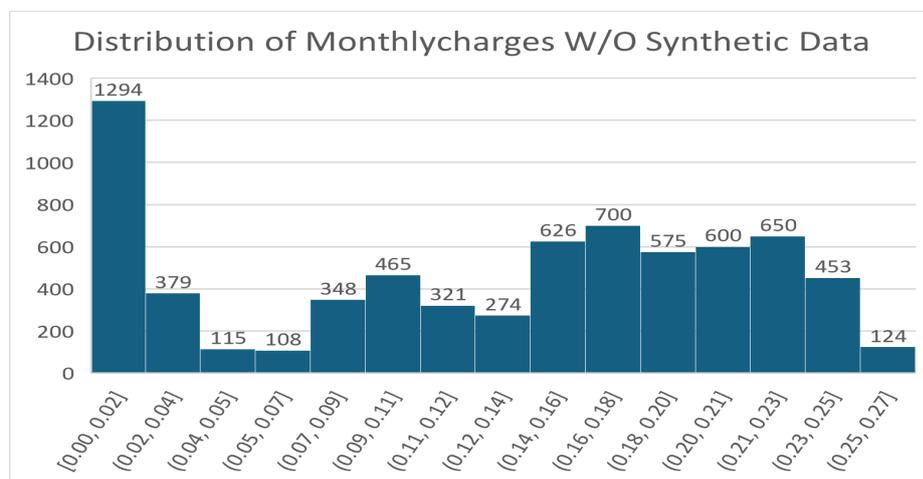
(d) 50% synthetic

Figure 5.9: Plot illustrating impact of synthetic data generation on Sample data points of Minority class. (a) data points with no synthetic, (b) 30% synthetic, (c) 40% synthetic, (d) 50% synthetic.

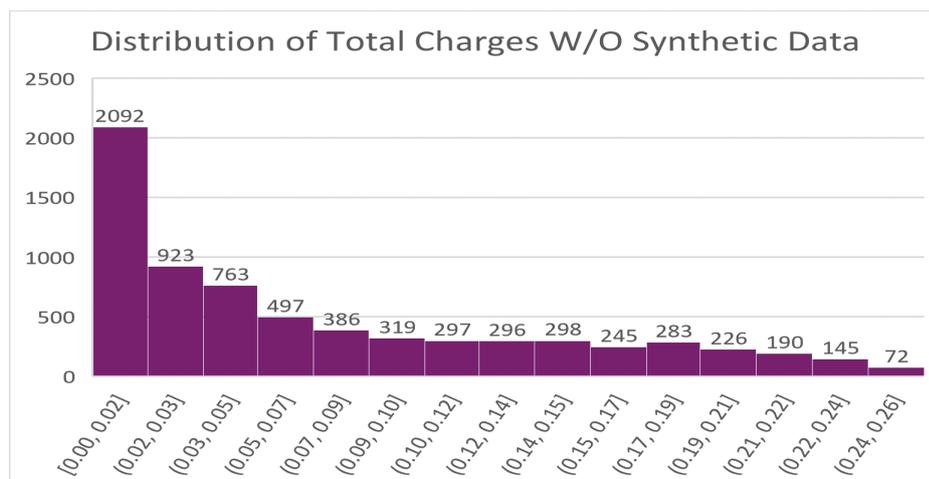
CHAPTER 5. A QUANTUM APPROACH TO SYNTHETIC MINORITY
OVERSAMPLING TECHNIQUE (SMOTE)



(a)

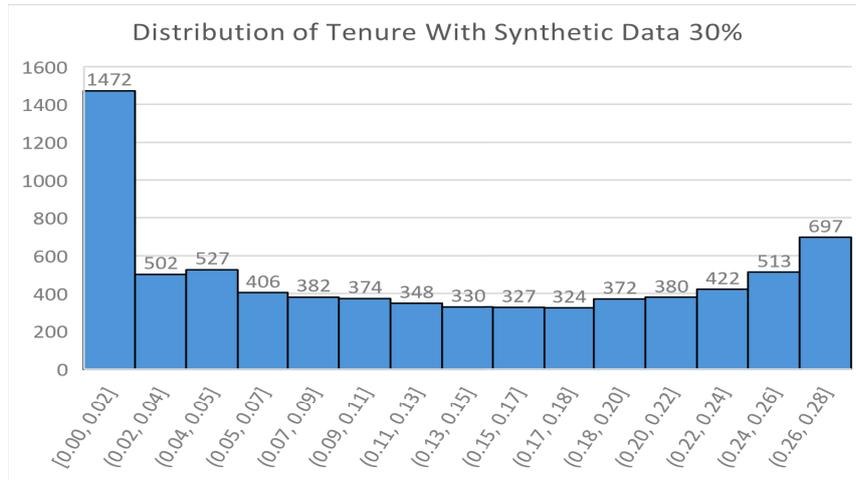


(b)

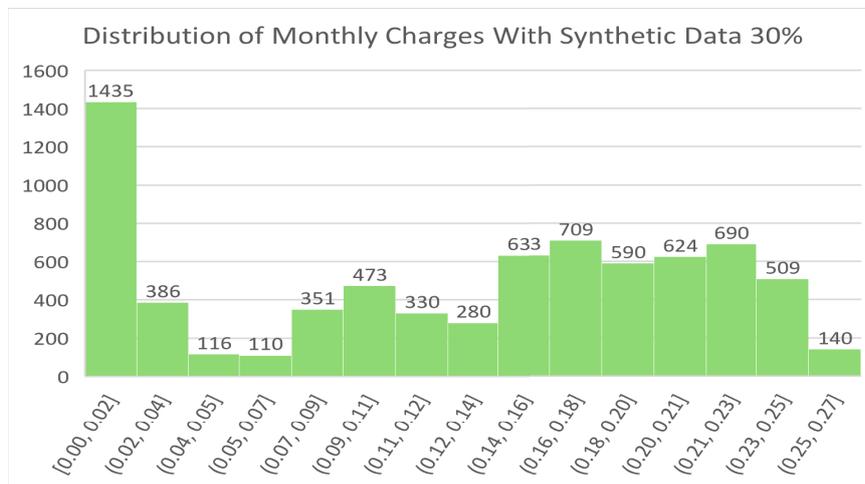


(c)

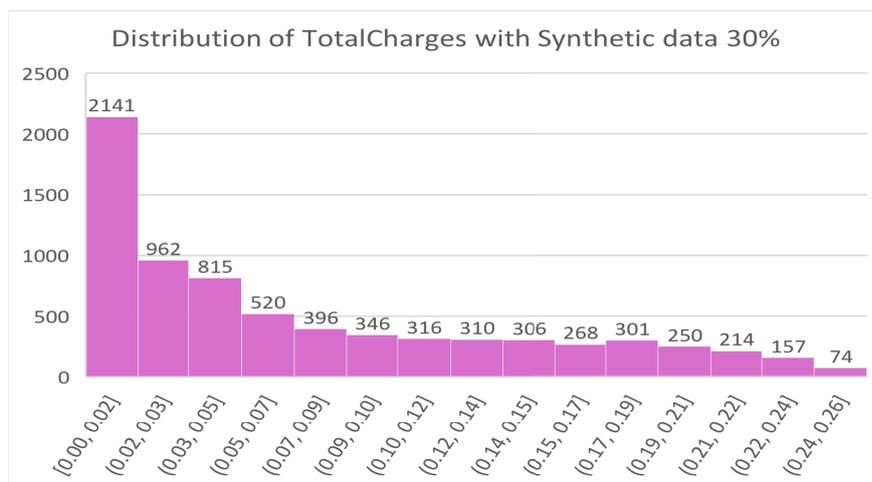
Figure 5.10: Plot illustrating distribution of 3 columns: (a) Tenure, (b) Monthly-Charges, and (c) TotalCharges.



(a)



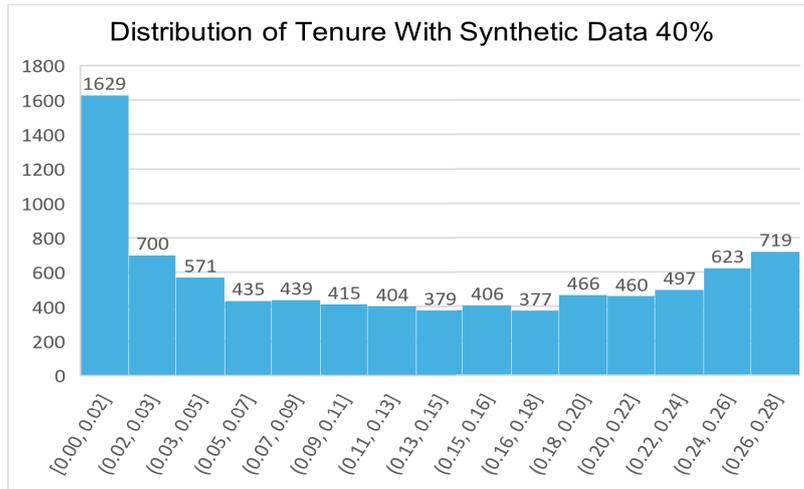
(b)



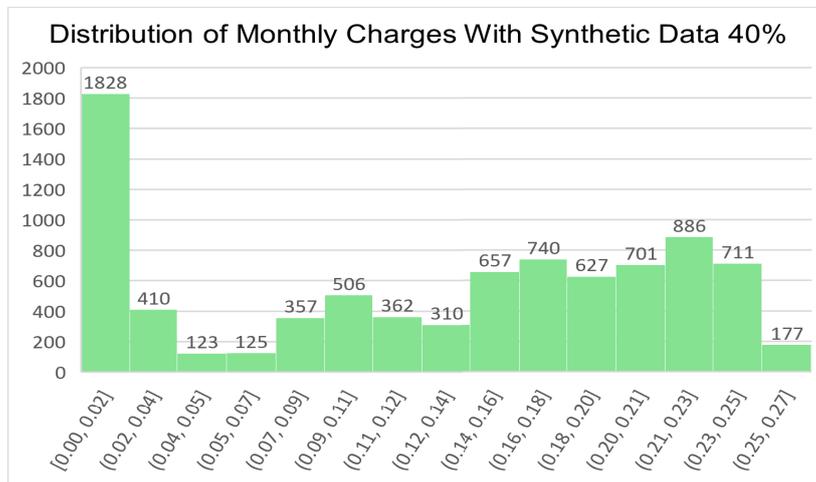
(c)

Figure 5.11: Plot illustrating distribution of 3 columns with induction of synthetic datapoints with overall 30% minority : (a) Tenure, (b) MonthlyCharges, and (c) TotalCharges.

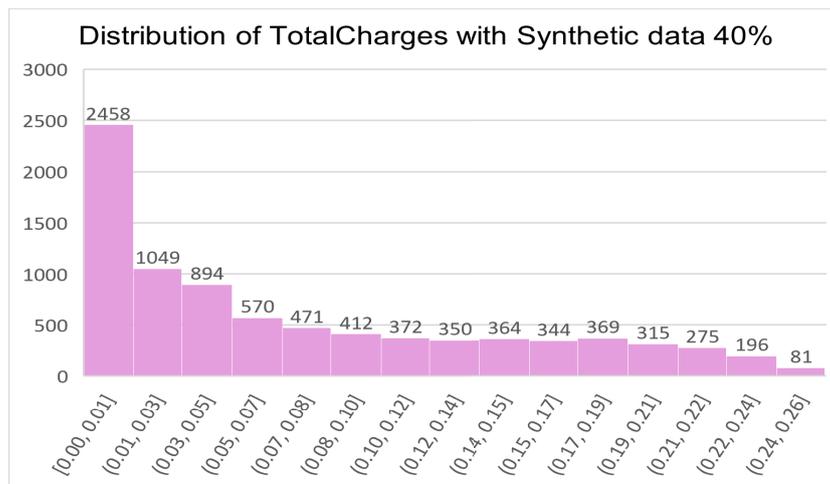
CHAPTER 5. A QUANTUM APPROACH TO SYNTHETIC MINORITY
OVERSAMPLING TECHNIQUE (SMOTE)



(a)

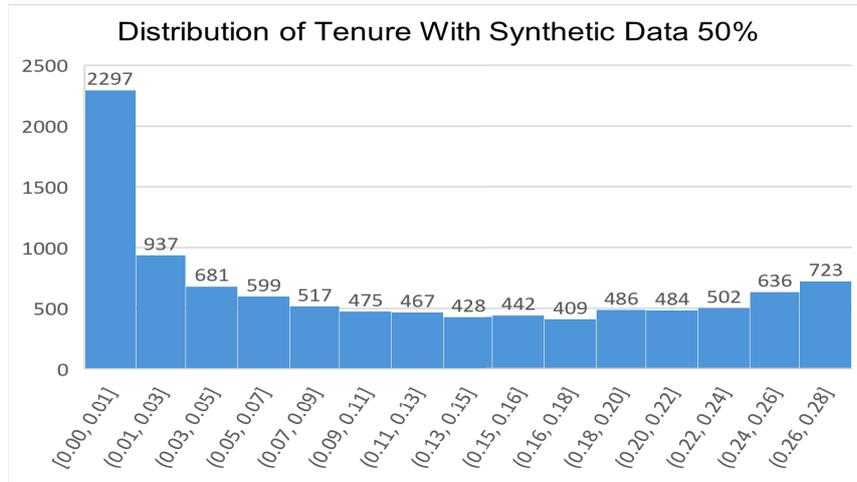


(b)

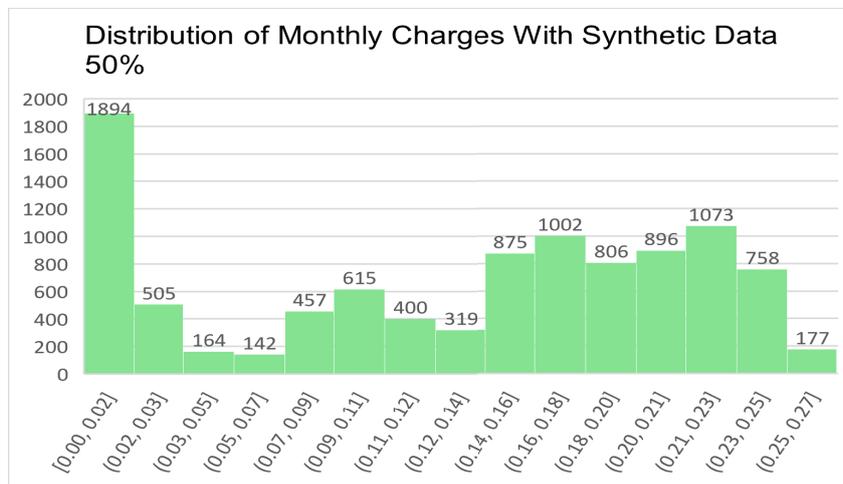


(c)

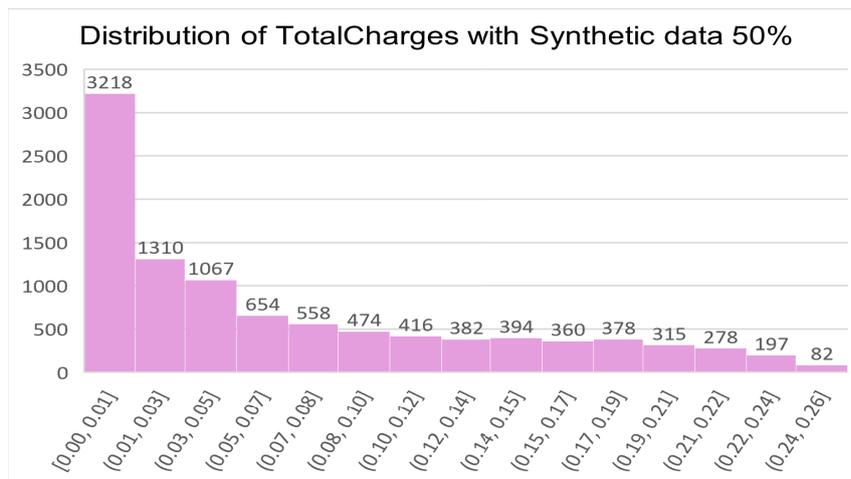
Figure 5.12: Plot illustrating distribution of 3 columns with induction of synthetic data points with overall 40% minority: (a) Tenure, (b) MonthlyCharges, and (c) TotalCharges.



(a)



(b)



(c)

Figure 5.13: Plot illustrating distribution of 3 columns with induction of synthetic data points with overall 50% minority : (a) Tenure, (b) MonthlyCharges, and (c) TotalCharges.

on the Telecom Churn Dataset. The selection of these models was made to assess the influence of using SMOTE on the performance of the models, particularly in situations characterized by an imbalance in class distribution. The Random Forest algorithm is well recognized for its ability to efficiently handle skewed datasets. This model utilizes ensemble learning by creating multiple decision trees and aggregating their predictions to mitigate overfitting. The algorithm natively addresses class imbalances by using techniques such as bootstrap sampling and adjusting its class weights parameter to enhance sensitivity towards the minority class. This eliminates the requirement for external interventions like SMOTE [87]. On the other hand, Logistic Regression, a model well regarded for its simplicity and effectiveness in situations where binary classification is needed, was selected to provide a contrasting analytical viewpoint. The classification strategy of Logistic Regression, which entails estimating the likelihood that a certain data point belongs to a specific class, does not inherently tackle the issue of class imbalance [88]. This attribute makes it a perfect contender for evaluating the immediate impacts of SMOTE on model efficacy, providing valuable observations on how SMOTE might augment a model's capacity to identify the underrepresented class in unbalanced datasets. The research seeks to evaluate the efficiency of the SMOTE method across various modeling techniques by comparing the performances of these models before and after their deployment. An investigation of SMOTE's adaptability in enhancing classification results is crucial, especially for models such as Logistic Regression that lack inherent methods for addressing data imbalances [73]. To evaluate the model, we have used the Confusion Matrix, Accuracy, Precision, Recall, F1-Score, and the Area Under the Receiver Operating Characteristic Curve (AUC-ROC). Below are the model evaluation charts for the Random Forest Model followed by the Logistic Regression Model.

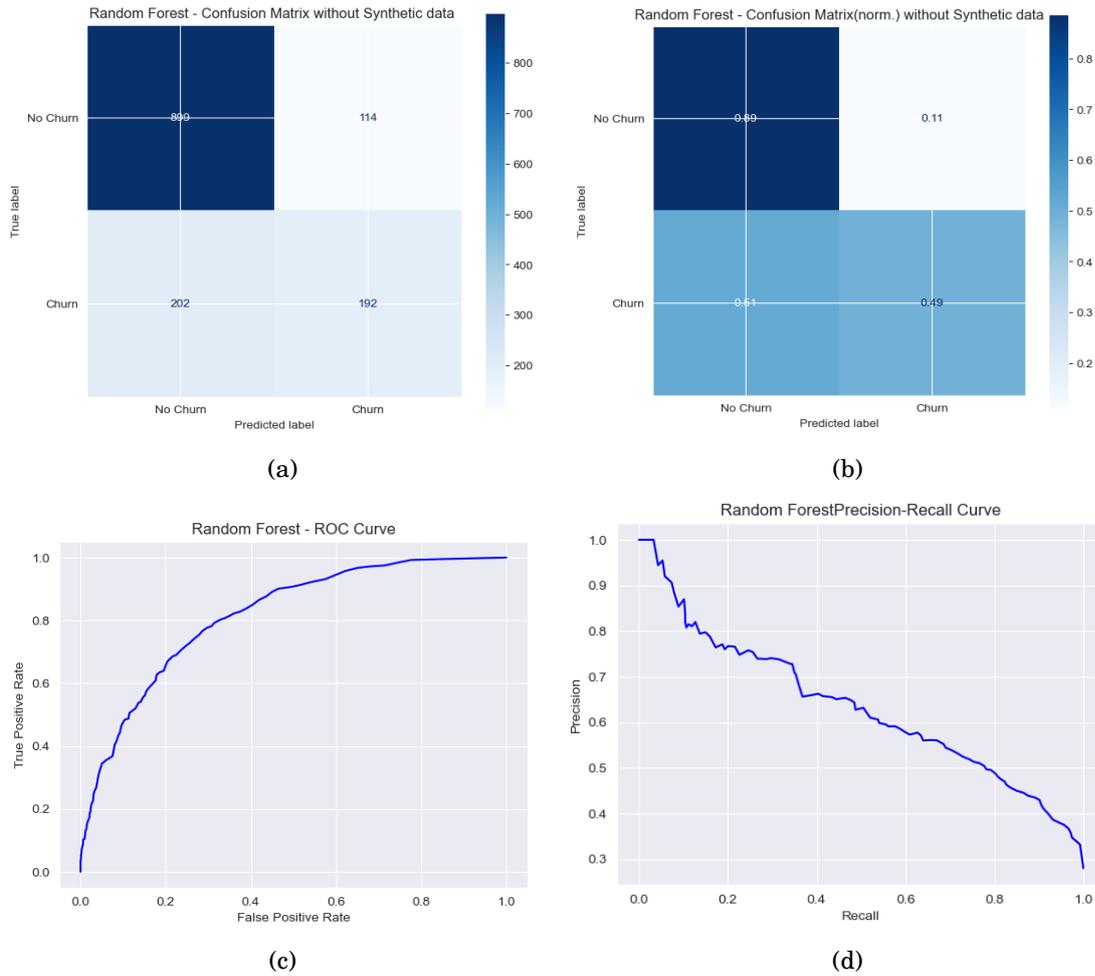


Figure 5.14: Plot illustrating Model Charts for random forest model with out SMOTE. (a) Confusion Matrix Random Forest Model, (b) Normalized Confusion Matrix Random Forest Model, (c) AUC-ROC Random Foerest Model, (d) Precision Recall Curve Random Forest Model.

CHAPTER 5. A QUANTUM APPROACH TO SYNTHETIC MINORITY OVERSAMPLING TECHNIQUE (SMOTE)

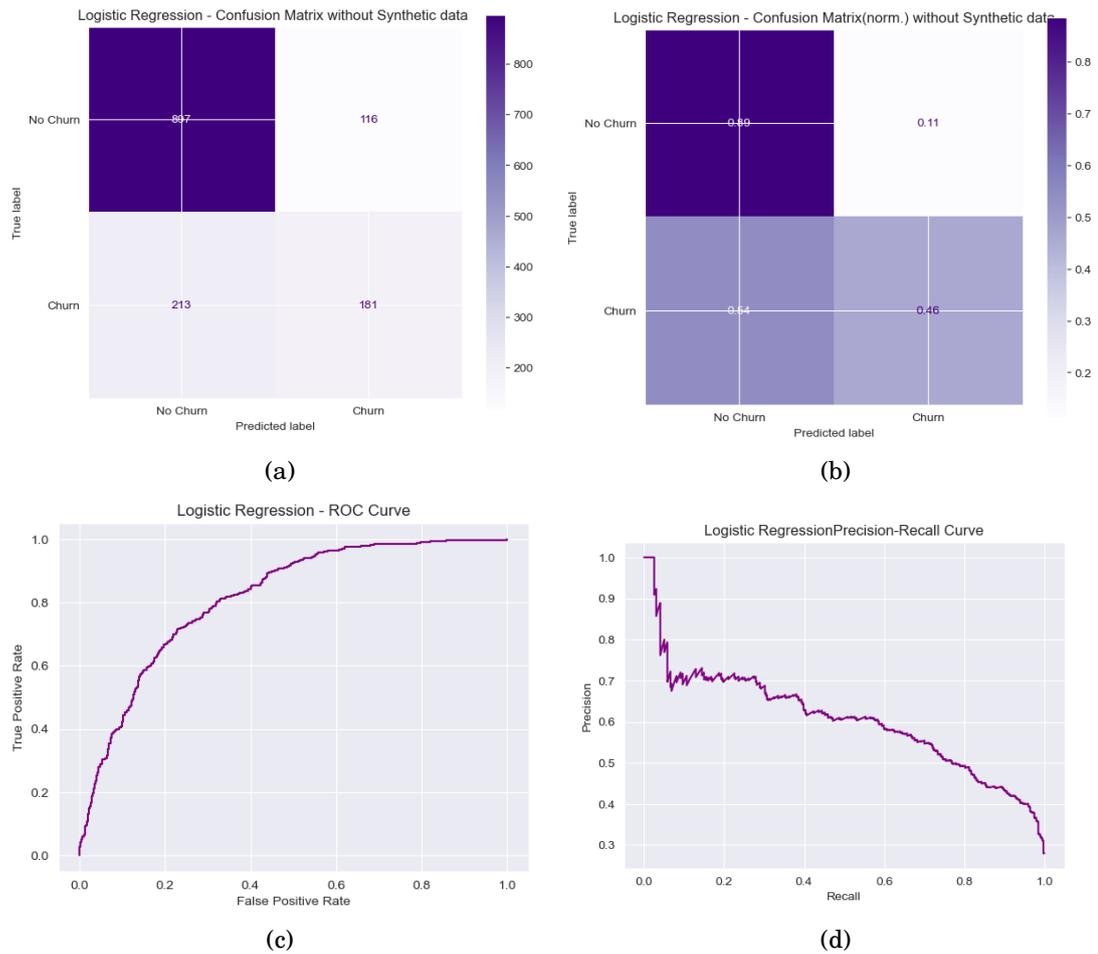


Figure 5.15: Plot illustrating Model Charts for logistic regression model without SMOTE. (a) Confusion Matrix Logistic Regression Model, (b) Normalized Confusion Matrix Logistic Regression Model, (c) AUC-ROC Logistic Regression Model, (d) Precision Recall Curve Logistic Regression Model.

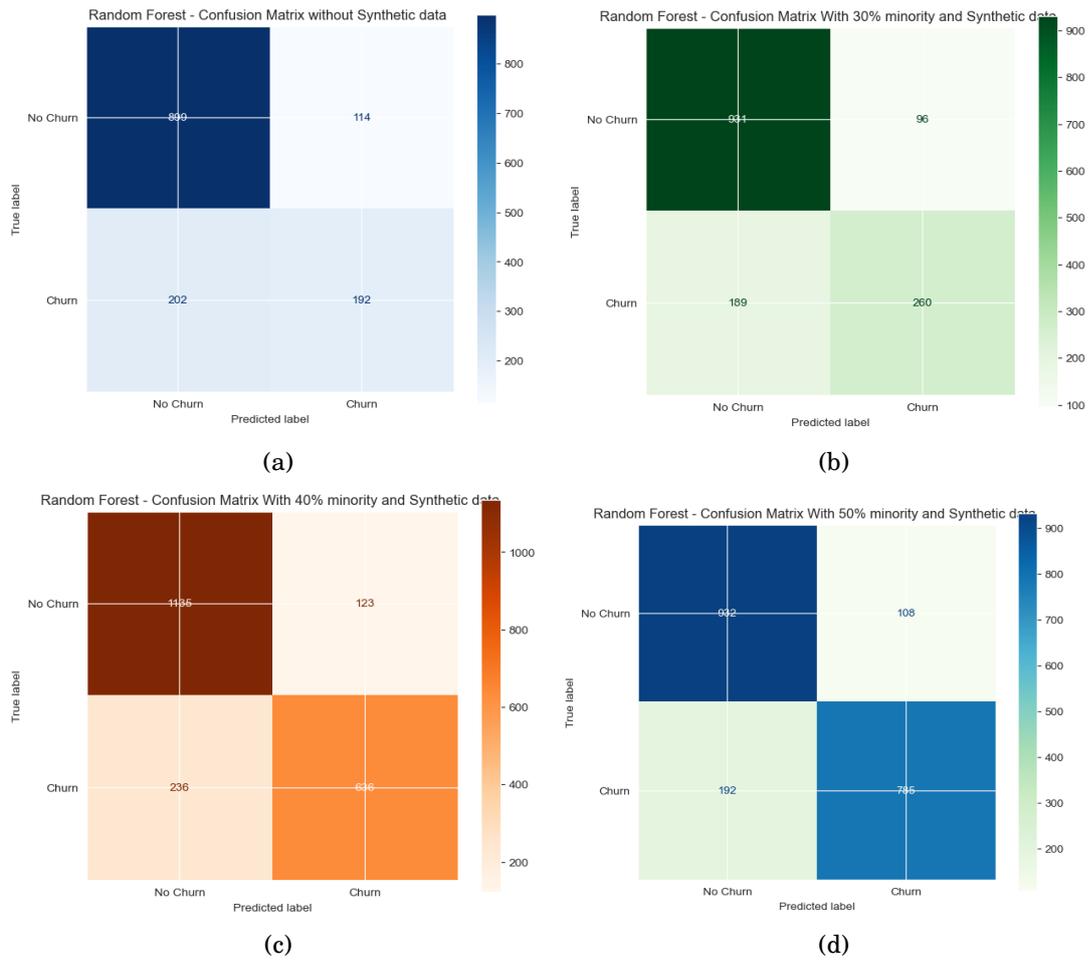


Figure 5.16: Plot illustrating Confusion Matrix for random forest model with and without smote for comparison. (a) Confusion Matrix Random Forest Model without smote, (b) Confusion Matrix Random Forest Model with smote and 30% Minority, (c) Confusion Matrix Random Forest Model with smote and 40% Minority, (d) Confusion Matrix Random Forest Model with smote and 50% Minority.

CHAPTER 5. A QUANTUM APPROACH TO SYNTHETIC MINORITY OVERSAMPLING TECHNIQUE (SMOTE)

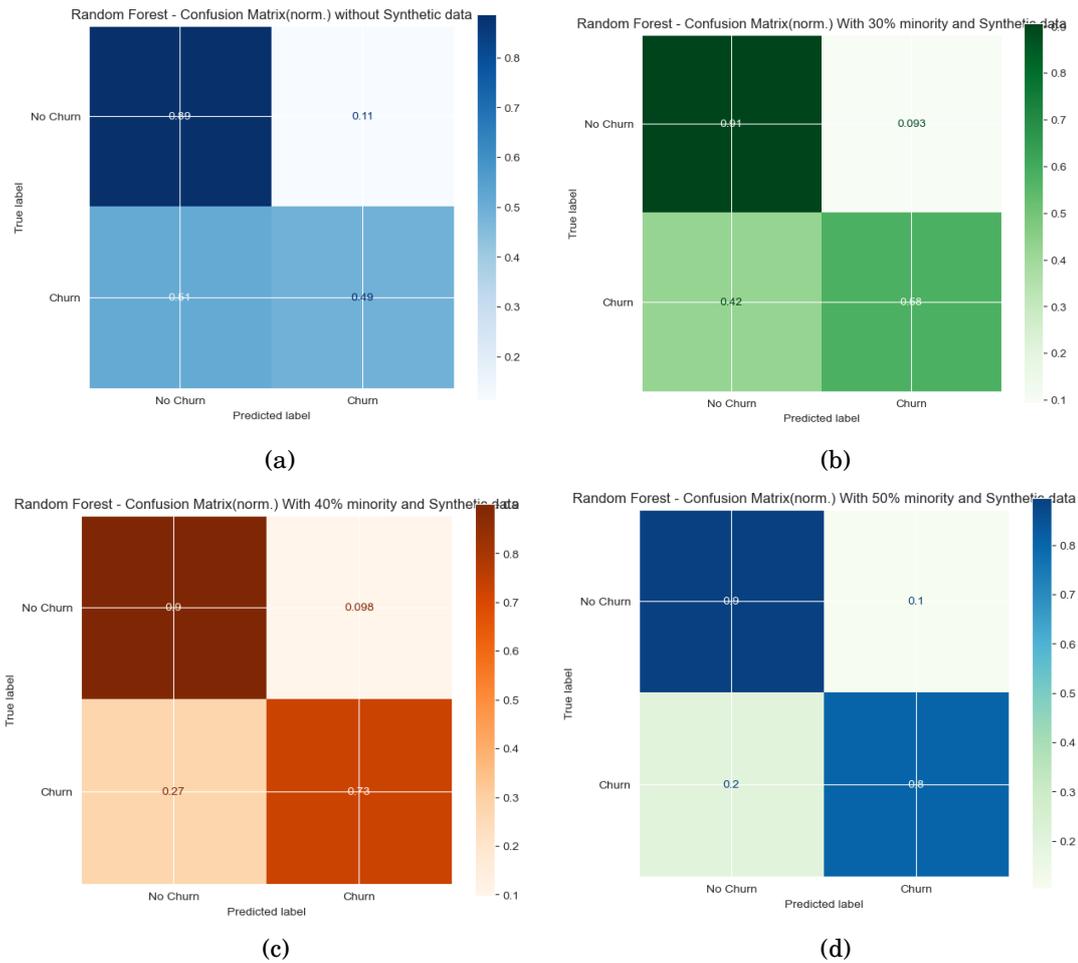


Figure 5.17: Plot illustrating Normalized Confusion Matrix for random forest model with and without smote for comparison. (a) Normalized Confusion Matrix Random Forest Model without smote, (b) Normalized Confusion Matrix Random Forest Model with smote and 30% Minority, (c) Normalized Confusion Matrix Random Forest Model with smote and 40% Minority, (d) Normalized Confusion Matrix Random Forest Model with smote and 50% Minority.

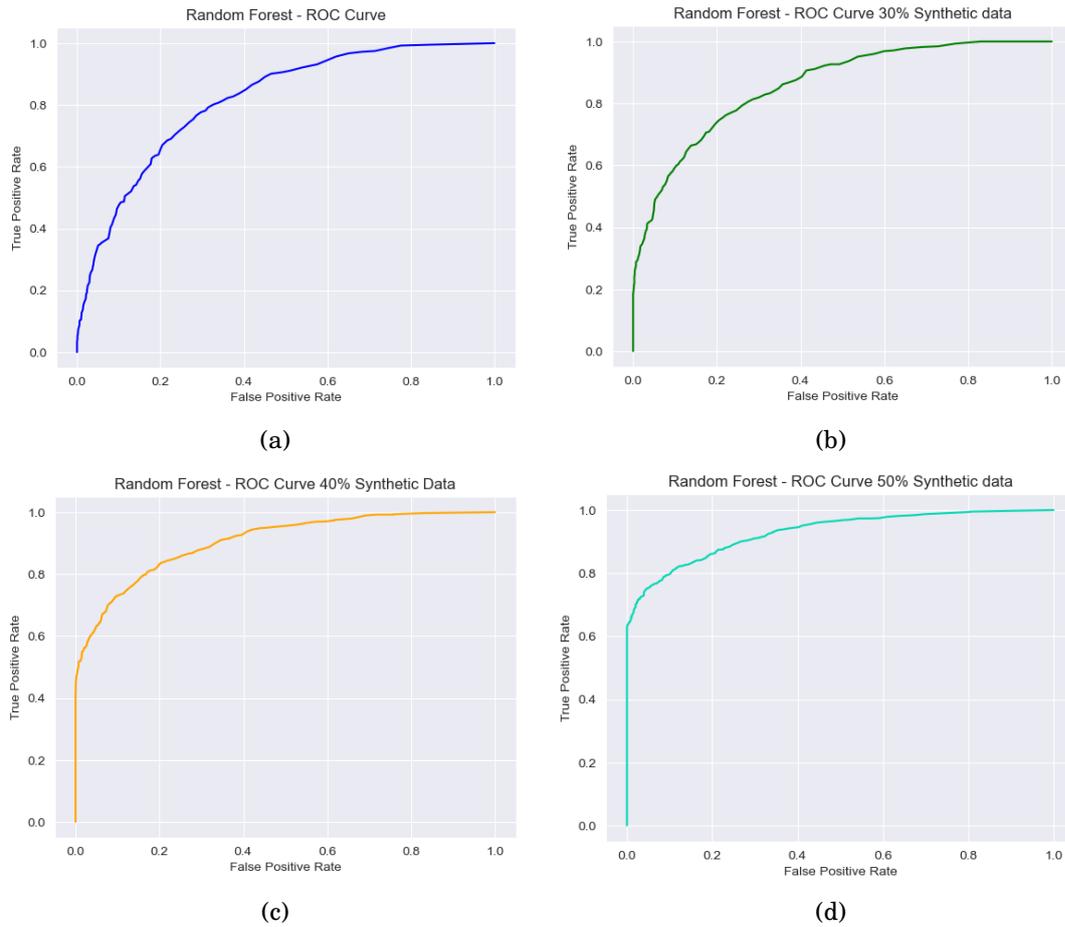


Figure 5.18: Plot illustrating Area Under Receiver Operating Characteristic Curve (AUC-ROC) for random forest model with and without smote for comparison. (a) AUC-ROC Random Forest Model without smote, (b) AUC-ROC Random Forest Model with smote and 30% Minority, (c) AUC-ROC Random Forest Model with smote and 40% Minority, (d) AUC-ROC Random Forest Model with smote and 50% Minority.

CHAPTER 5. A QUANTUM APPROACH TO SYNTHETIC MINORITY OVERSAMPLING TECHNIQUE (SMOTE)

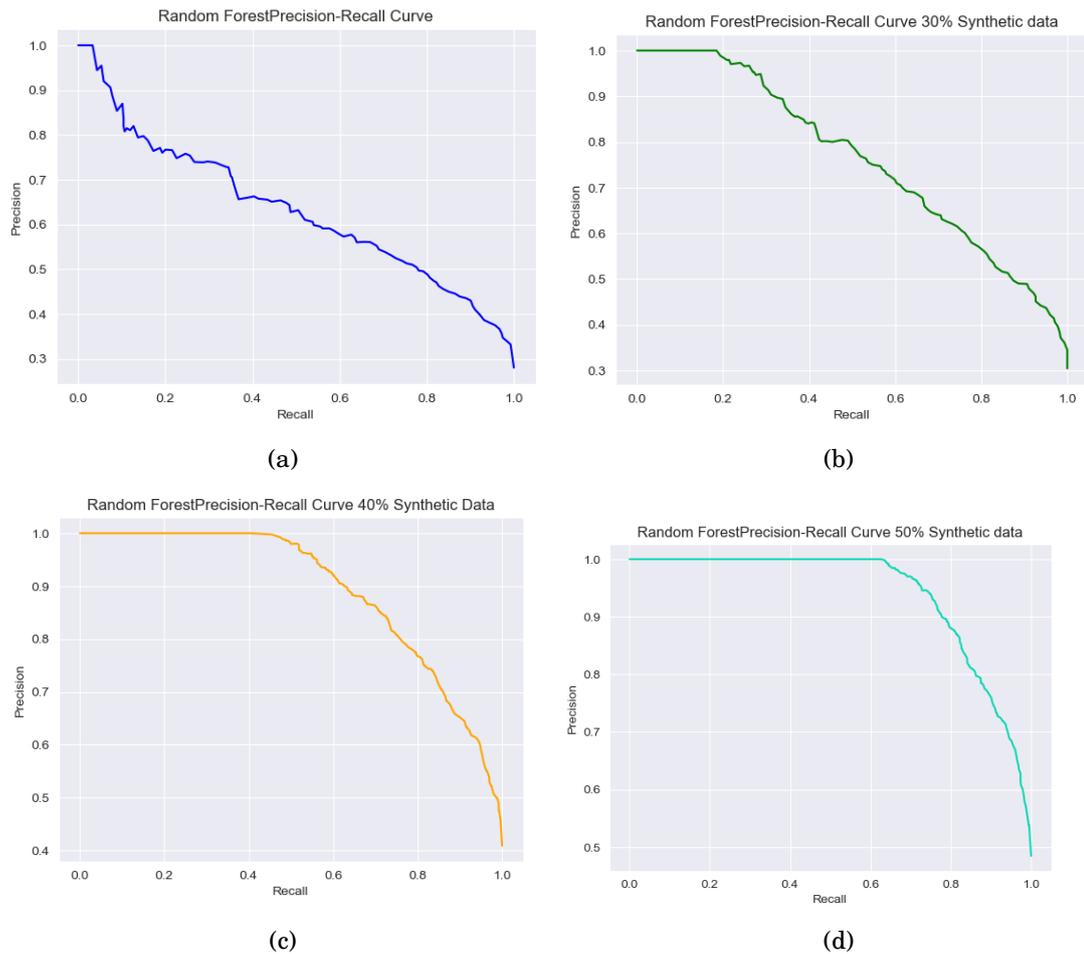


Figure 5.19: Plot illustrating Precision-Recall Curve (AUC) for random forest model with and without smote for comparison. (a) Precision-Recall Curve (AUC) Random Forest Model without smote, (b) Precision-Recall Curve (AUC) Random Forest Model with smote and 30% Minority, (c) Precision-Recall Curve (AUC) Random Forest Model with smote and 40% Minority, (d) Precision-Recall Curve (AUC) Random Forest Model with smote and 50% Minority.

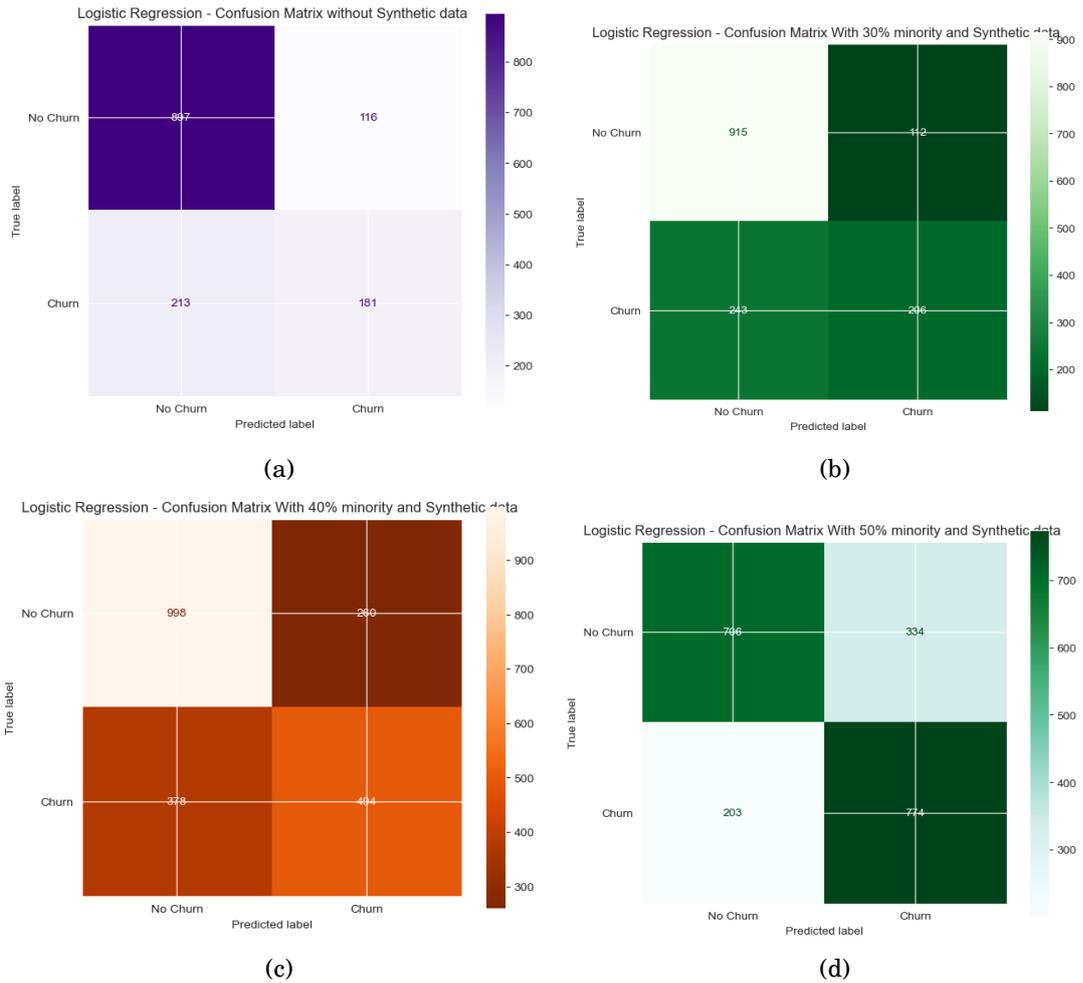


Figure 5.20: Plot illustrating Confusion Matrix for Logistic Regression model with and without smote for comparison. (a) Confusion Matrix Logistic Regression Model without smote, (b) Confusion Matrix Logistic Regression Model with smote and 30% Minority, (c) Confusion Matrix Logistic Regression Model with smote and 40% Minority, (d) Confusion Matrix Logistic Regression Model with smote and 50% Minority.

CHAPTER 5. A QUANTUM APPROACH TO SYNTHETIC MINORITY OVERSAMPLING TECHNIQUE (SMOTE)

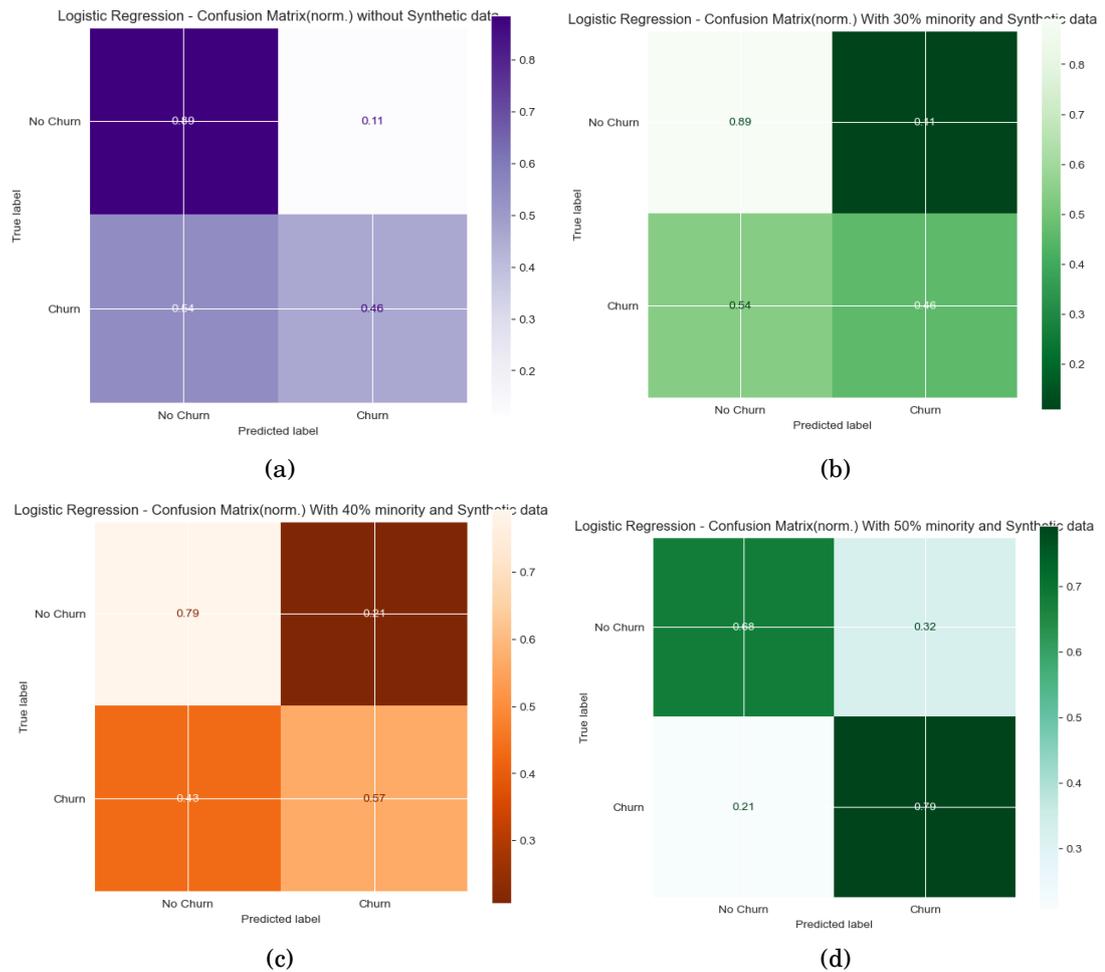


Figure 5.21: Plot illustrating Normalized Confusion Matrix for Logistic Regression model with and without smote for comparison. (a) Normalized Confusion Matrix Logistic Regression Model without smote, (b) Normalized Confusion Matrix Logistic Regression Model with smote and 30% Minority, (c) Normalized Confusion Matrix Logistic Regression Model with smote and 40% Minority, (d) Normalized Confusion Matrix Logistic Regression Model with smote and 50% Minority.

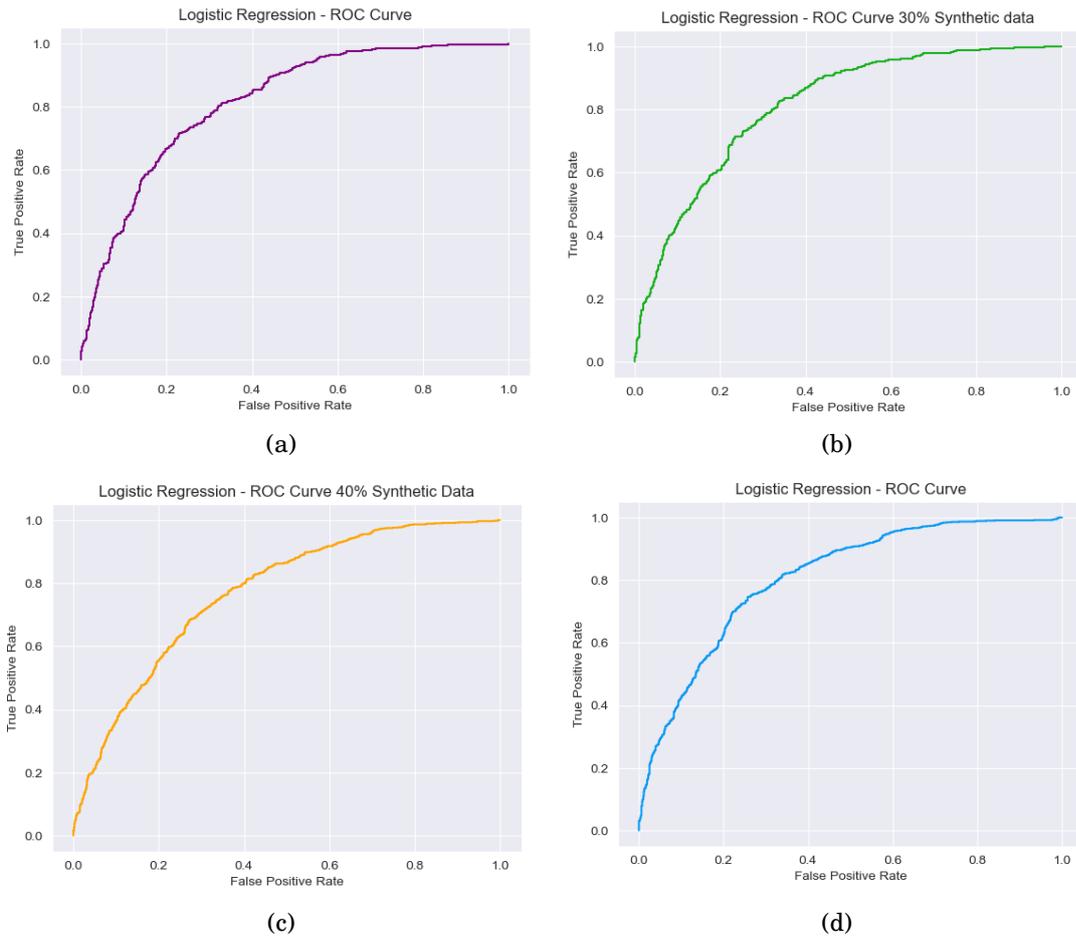


Figure 5.22: Plot illustrating Area Under Receiver Operating Characteristic Curve (AUC-ROC) for Logistic Regression model with and without smote for comparison. (a) AUC-ROC Logistic Regression Model without smote, (b) AUC-ROC Logistic Regression Model with smote and 30% Minority, (c) AUC-ROC Logistic Regression Model with smote and 40% Minority, (d) AUC-ROC Logistic Regression Model with smote and 50% Minority.

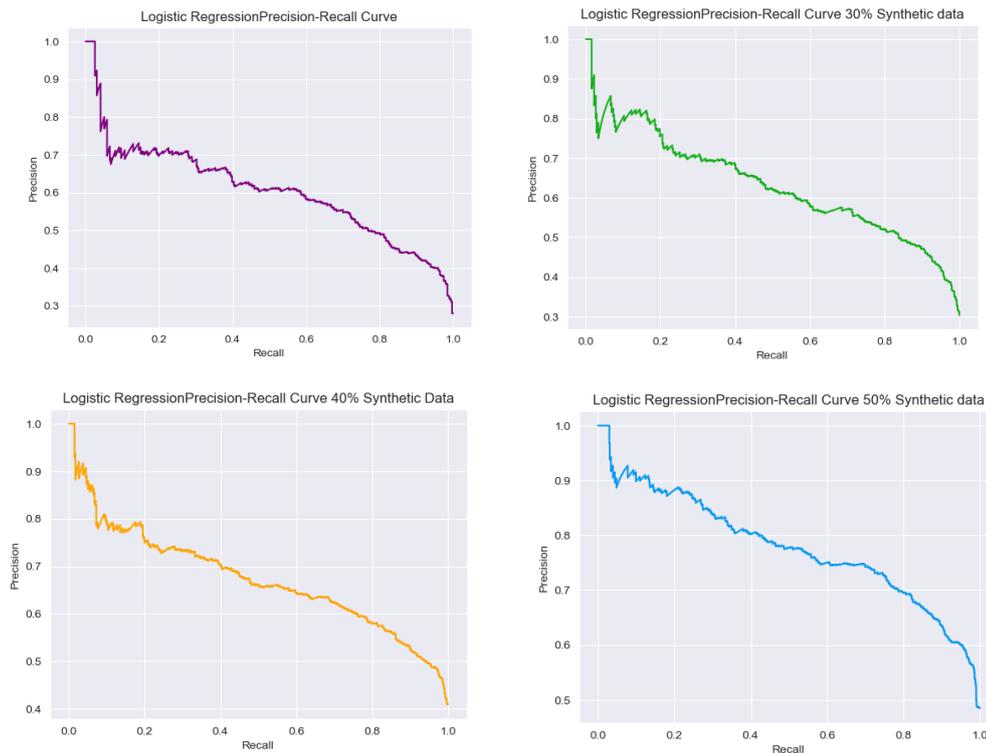


Figure 5.23: Plot illustrating Precision-Recall Curve (AUC) for Logistic Regression model with and without smote for comparison. (a) Precision-Recall Curve (AUC) Logistic Regression Model without smote, (b) Precision-Recall Curve (AUC) Logistic Regression Model with smote and 30% Minority, (c) Precision-Recall Curve (AUC) Logistic Regression Model with smote and 40% Minority, (d) Precision-Recall Curve (AUC) Logistic Regression Model with smote and 50% Minority.

In the next section, we will describe the impact of SMOTE on the above evaluation charts.

5.4.1.6 Impact of Quantum SMOTE on Model Statistics

As we applied SMOTE on our two chosen models, we observed different behaviors of the models post-application of QuantumSMOTE.

Random Forest:

The Random Forest model excels in effectively addressing the Telecom Churn Dataset, particularly when dealing with imbalances in class distribution. The model's intrinsic advantages, together with its performance improvements using

the SMOTE, provide a detailed analysis of its impact in tricky classification scenarios. As we walk through the model's performance parameters of Confusion matrices (Figs. 5.16 and 5.17), Receiver Operating Characteristic Curve (AUC-ROC) (Fig. 5.18), Precision Recall Curve (AUC) (Fig. 5.19) we can see gradual improvements with induction of synthetic samples using SMOTE. We discuss the overall improvements in the points below.

- *Performance Without Synthetic Data*

The introduction of SMOTE to the dataset led to observable improvements across various performance measures. Notably, as the percentage of synthetic minority increased, both test accuracy and F1 scores saw visible improvements. These improvements highlight the synergy between Random Forest's ensemble methodology and the balanced class distribution achieved through SMOTE. The model's adaptability to handle more balanced datasets and improve in predictive accuracy and precision recall underscores its versatility and effectiveness in handling imbalanced data scenarios.

- *Effects of Varying Degrees of Synthetic Data Augmentation on Performance 30% Minority with Synthetic Data:*

Test accuracy and F1 scores started to rise at this augmentation level, signaling the start of performance gains. With no change to the training data, the model achieved a test accuracy of 0.800813 and an F1 score improvement of 0.6343. Both the PR and ROC AUCs increased, reaching 0.757604 and 0.854414, respectively.

40% and 50% Minority with Synthetic Data:

The test accuracy (0.822183) and F1 score (0.764202) were significantly improved by 40% Minority using Synthetic Data. PR had an AUC of 0.888143 and ROC had an AUC of 0.905165. The test accuracy increased to 0.846306 and the F1 score to 0.834755 with 50% Minority using Synthetic Data. At their peak, PR and ROC AUC values were 0.940063 and 0.928649, respectively. Both the 40% and 50% SMOTE augmentation levels improved the model more, but the 50% augmentation level was when it really shone. Results showing significant improvements in test accuracy, F1 scores, and AUC

scores for PR and ROC show that the model is better at identifying the minority class and can generalize more effectively.

Logistic Regression: Performance in the analysis of the Logistic Regression model depicts its ability to handle class imbalance, especially when augmented with the SMOTE. We describe the behavior of Logistic Regression and its outcomes across different scenarios in following sections based on Confusion matrices (Figs. 5.20 and 5.21), Receiver Operating Characteristic Curve (AUC-ROC) (Fig. 5.22), Precision-Recall Curve (AUC) (Fig. 5.23).

- *Performance Without Synthetic Data:* Initially, the Logistic Regression model showed decent performance with a test accuracy of 0.796622, indicating its ability to accurately predict outcomes in over 80% of cases.

Nevertheless, the F1 score, which is calculated as the harmonic mean of accuracy and recall, had a relatively low value of 0.523878. This suggests that while the model was usually reliable, it had challenges in achieving a trade-off between accuracy and recall, especially in correctly identifying the minority class. The Precision-Recall (PR) and Receiver Operating Characteristic (ROC) obtained Area Under the Curve (AUC) scores of 0.60415 and 0.814921, respectively. These scores indicate a reasonable potential to differentiate between classes, while there is potential for improvement in managing unbalanced data.

- *30% Minority with Synthetic:* By inducing synthetic data to constitute 30% of the minority class, the test accuracy saw a slight decline to 0.759485. This reduction implies that while the synthetic data was intended to balance the distribution of classes, it could have contributed to the complexity of class distribution that somewhat affected the general accuracy of predictions. However, the F1 score saw a small rise to 0.537158, suggesting that the model's capacity to maintain a balance between accuracy and recall improved under somewhat more equitable class settings. The AUC scores for PR (Precision-Recall) and ROC (Receiver Operating Characteristic) saw marginal enhancements to 0.632638 and 0.81238, respectively. These gains

indicate a minor boost in the model's ability to differentiate between the classes when synthetic data is employed.

- *40% Minority with Synthetic:* With the percentage of synthetic data was increased to 40%, the test accuracy decreased to 0.700469. However, the F1 score increased to 0.607626. This implies that while the model's overall prediction accuracy declined, its capacity to detect the minority class improved, as shown by the higher F1 score. The area under the curve (AUC) scores for precision-recall (PR) and receiver operating characteristic (ROC) were 0.673914 and 0.769356, respectively. These values suggest that the model's accuracy and recall balance improved, but there was a minor decline in its overall discriminating power.
- *50% Minority with Synthetic:* By using synthetic data to achieve a 50% minority representation, the model demonstrated a notable improvement in test accuracy, reaching 0.733763. Yet, the F1 score increased substantially to 0.742446. The substantial rise in the F1 score demonstrates the improved ability of the model to properly detect the minority class due to a more evenly balanced dataset. The area under the curve (AUC) scores for precision-recall (PR) and receiver operating characteristic (ROC) increased to 0.778797 and 0.807275, respectively, indicating the enhanced ability of the model to distinguish between classes in a more balanced setup.

5.4.1.7 Final thoughts on SMOTE Performance

The comparison of Logistic Regression and Random Forest models, enhanced with SMOTE, demonstrates the intricate nature of resolving class imbalance in machine learning. The performance enhancements of the Logistic Regression model, particularly in achieving a balanced precision-recall trade-off with the use of SMOTE, are consistent with the research conducted by Chawla et al. (2002). In their study, SMOTE was presented as a method to increase classifier performance by mitigating the issue of class imbalance via the generation of synthetic samples.

The Random Forest model demonstrates good performance, regardless of SMOTE. This underscores the model's intrinsic abilities in effectively dealing

Random Forest					
Scores	Accuracy Score			AUC Score	
Data Set Type	Train	Test	F1 Score	PR	ROC
Without Synthetic	1.000	0.784	0.575	0.627	0.811
30% Minority with Synthetic	1.000	0.801	0.634	0.758	0.854
40% Minority with Synthetic	0.996	0.822	0.764	0.888	0.905
50% Minority with Synthetic	0.996	0.846	0.835	0.940	0.929
Logistic Regression					
Without Synthetic	0.797	0.766	0.524	0.604	0.815
30% Minority with Synthetic	0.753	0.759	0.537	0.633	0.812
40% Minority with Synthetic	0.724	0.700	0.608	0.674	0.769
50% Minority with Synthetic	0.732	0.734	0.742	0.779	0.807

Table 5.1: Table comparing Accuracy, F1, and AUC score of Random Forest Model for telecom churn dataset without SMOTE, and post SMOTE with minority % as 30%, 40%, and 50%.

with class imbalances [87]. The ensemble strategy of the model, which combines predictions from numerous decision trees, inevitably offers a degree of resilience to imbalance, which is further strengthened by the use of SMOTE. Fernandez et al. [74] provide evidence supporting the effectiveness of ensemble approaches in handling unbalanced data. They propose that combining techniques such as Random Forest with SMOTE may lead to substantial improvements in model performance. All of the findings described in the assessment of Model performances are summarized in the table 5.1 and the Confusion Matrix comparison table 5.2.

Confusion Matrix Comparison								
Random Forest	W/O Synthetic		30% SMOTE		40% SMOTE		50% SMOTE	
	TP	FP	TP	FP	TP	FP	TP	FP
	899	114	931	96	1135	123	932	108
	FN	TN	FN	TN	FN	TN	FN	TN
	202	192	189	260	236	636	192	785
Logistic Regression	W/O Synthetic		30% SMOTE		40% SMOTE		50% SMOTE	
	TP	FP	TP	FP	TP	FP	TP	FP
	807	116	915	112	998	260	706	334
	FN	TN	FN	TN	FN	TN	FN	TN
	213	181	243	206	387	494	203	774

Table 5.2: Table comparing Confusion matrix of Random Forest and Logistic regression without SMOTE, and post SMOTE with minority% as 30%, 40%, and 50%.

5.4.2 Inferences from Simulation

In the process of creating the Quantum-SMOTE algorithm, we have come across several conclusions that we want to outline in the points below.

- The QuantumSMOTE algorithm functions similarly to the traditional SMOTE method but has the benefit of quantum phenomena.
- The QuantumSMOTE technique utilizes the swap test and quantum rotation, distinguishing it from the standard SMOTE algorithm that relies on K Nearest Neighbors (KNN) [89, 90] and Euclidean distances [73, 80, 82, 91].
- The QuantumSMOTE technique utilizes quantum rotation to eliminate neighbor dependencies and create several synthetic data points from a single data point in the minority class.
- The technique includes hyperparameters that enable users to manage various elements of synthetic data creation, such as rotation angle, minority percentage, and splitting factor.

- The QuantumSMOTE procedure generates synthetic data points to ensure that the distribution of variables closely resembles the original data distribution.
- By selecting a smaller angle of rotation, the synthetic data points are positioned near the original minority data point, increasing the density of minority data points in a sparsely populated area.
- The rotation circuit for minority data points does not encourage the use of any entanglement process or similar gates such as CNOT, ZZ, etc. since they will generate undesired effects on rotation and result in unexpected outcomes.
- By using the compact swap test approach, more columns may be stored in fewer qubits. We used 5 qubits to handle 32 variables, and by scaling, we can handle 1024 variables with just 10 qubits.
- The algorithm's use of low-depth circuits makes it less susceptible to issues associated with lengthy circuits like noise and decoherence. It effectively showcases how quantum techniques may enhance traditional machine-learning methods.
- Similar to classical SMOTE, QuantumSMOTE generates synthetic data that enhances the Precision-Recall score of machine learning algorithms such as Logistic Regression [88] and significantly benefits ensemble algorithms like Random Forest [87]. This suggests its alignment with contemporary machine learning environments and confirms its applicability in current unbalanced classification scenarios.

5.5 Conclusion

The QuantumSMOTE technique improves conventional class imbalance correction by employing quantum computing, particularly swap tests and quantum rotation, as opposed to classical approaches that rely on K Nearest Neighbors (KNN) and Euclidean distances. This quantum approach allows for the direct production of synthetic data points from minority class instances using quantum

rotations, preventing the need for neighbor-based interpolation. QuantumSMOTE has customisable hyperparameters such as rotation angle, minority percentage, and splitting factor, allowing for personalized synthetic data synthesis to accurately solve dataset imbalances.

One notable feature of QuantumSMOTE is its capacity to generate synthetic instances that closely resemble the original data distribution, along with enhancing the balance of minority classes in datasets. The algorithm's use of compact swap tests enables efficient data representation, needing fewer qubits to manage a high number of variables, hence improving scalability and lowering quantum computing resource needs. Furthermore, its use of low-depth circuits reduces sensitivity to quantum noise and decoherence, making it a reliable option for quantum-enhanced data augmentation.

QuantumSMOTE's success is proven by its favorable influence on the Precision-Recall scores of machine learning algorithms such as Logistic Regression and Random Forest, highlighting its compatibility and utility in modern machine learning procedures. This technique is a forward-thinking integration of quantum computing with data science, providing an innovative and efficient solution to the problem of class imbalance in machine learning datasets.

Part III

Part III

CONCLUSION

6.1 Research Findings

In this thesis, we developed quantum machine learning based approach for solving critical problems that include Optimization and data classification scenarios. In this context, we have studied the impact of quantum noise on quantum algorithms and presented an approach to quantify the impact of quantum noise. We discuss the overall findings of the research below.

6.1.1 Chapter 3

In chapter 3 "[Analysis of The Vehicle Routing Problem Solved via Hybrid Quantum Algorithms in the Presence of Noisy Channels](#)" the paper sheds light on the critical influence of noise in quantum computing simulations and outlines a path for future research that focuses on optimizing algorithms to handle real-world quantum environments effectively. We outline the outcomes of the research for this noise analysis under the following heads.

Chapter 3: Impact of Noise on VRP Solution

This research provides comprehensive simulation results for different noise parameters, comparing settings without noise to scenarios with multiple noise channels.

It has been shown that when the noise parameters or the number of circuit layers rise, departures from the classical minimum do not exhibit a linear trend. Notably, there were noticeable variations in performance when comparing outcomes from distinct circuit layers in the presence of specific noise conditions, such as the bit-flip noise channel. The investigation verifies that various noise channels have discrete effects on quantum circuits, while comparable patterns arise throughout a wide range of noise values (0.05 to 0.5). This implies a consistent behavior that applies to all forms of noise, which affects the approach to selecting noise parameters in simulations. The COBYLA optimizer has been shown to be very successful for noisy quantum circuits used in the VRP, surpassing other optimizers such as SPSA as the noise levels increase. This underscores the need of selecting the appropriate optimizer to alleviate the impact of noise in quantum calculations.

Chapter 3: Future Directions

The paper proposes an additional investigation into various optimization techniques in order to understand their viability under different levels of noise. This has the potential to result in more resilient quantum algorithms that are less sensitive to interference. There is a proposal to do experiments on larger Vehicle Routing Problem (VRP) instances utilizing tangible quantum devices. This advancement beyond traditional simulations will help in verifying the results in real-life scenarios and expand the limits of what can be simulated using conventional methods. Future research endeavors seek to construct more extensive noise models using empirical data obtained from actual noisy intermediate-scale quantum (NISQ) devices. This will augment the comprehension of noise impacts and direct the enhancement of quantum algorithms for pragmatic applications such as VRP.

6.1.2 Chapter 4

In chapter 4, "**Solving The Vehicle Routing Problem via Quantum Support Vector Machines,**" we have devised a novel approach to solving optimization problems using quantum machine learning, particularly Quantum Support Vector Machines (QSVM). We have also shown that the use of quantum encoding techniques can effectively emulate the effects of superposition and entanglement and help reduce circuit depths. We can summarise these under the following heads.

Chapter 4:Efficiency of QSVMs

QSVMs in solving VRP have been shown to be very successful, often achieving results that are on par with or superior to traditional quantum computing approaches like VQE (Variational Quantum Eigensolver) and QAOA (Quantum Approximate Optimization Algorithm). This highlights the ability of machine learning models in quantum computing to manage complex optimization challenges such as VRP effectively. Various encoding/decoding systems are essential for the generation of quantum superposition and entanglement, which are vital for quantum computing. These approaches may streamline the quantum circuit by minimizing the need to create mixer Hamiltonians, which are often used in QAOA. Unlike traditional quantum approaches, which often need numerous layers to reach high precision, QSVMs may achieve comparable or superior accuracy even with only the first layer of quantum circuits. Nevertheless, the act of increasing the number of layers has a propensity to decrease accuracy, indicating the existence of an optimum threshold for circuit depth in order to sustain superior performance. The selection of an optimizer, such as COBYLA, SLSQP, and L-BGFS-B, has a significant impact on the precision of QSVM solutions. Specifically, COBYLA demonstrates exceptional performance in terms of consistently achieving high levels of accuracy across different encoding methods and the complexity of quantum circuits.

Chapter 4:Future Directions for Research

We can propose that future research in the use of QSVMs should focus on investigating new or enhanced encoding and decoding methods that might effectively reduce the complexity of quantum circuits while simultaneously optimizing problem-solving accuracy. It is necessary to find a balance between the complexity of circuits, which refers to the number of gates, and the computational resources needed, such as classical memory and CPU. Conducting a thorough analysis and comparison of various quantum optimizers is crucial in order to identify the most efficient ones for certain quantum machine learning tasks. The diverse performance of various optimizers highlights the significance of their involvement in the efficacy of QSVMs. It would be essential to test the scalability of QSVM techniques by applying them to bigger VRP instances and more intricate routing difficulties. This involves deploying these models on physical quantum hardware to evaluate their practical viability and effectiveness in real-life situations. An in-depth analysis

of QSVMs in contrast to classic quantum computing systems such as VQE and QAOA, focusing on quantum cost, circuit depth, and overall efficiency, might provide valuable insights into their respective strengths and limits. Further empirical research is required to comprehensively comprehend the complete influence of various encoding strategies on the efficiency of QSVMs, particularly in the presence of diverse quantum circuit conditions and degrees of noise. This could possibly result in more resilient, efficient, and practical quantum computing systems.

6.1.3 Chapter 5

In Chapter 5, the article titled "[A Quantum Approach to Synthetic Minority Oversampling Technique \(SMOTE\)](#)" presents the Quantum SMOTE algorithm, which is a quantum computing adaptation of the traditional SMOTE approach. The system is designed to tackle class imbalances in datasets, and the study discusses its development and consequences. We will provide a concise overview of the main findings and implications of the study, as well as suggest possible avenues for further research.

Chapter 5: Overall Findings

Quantum SMOTE uses quantum rotation and swap tests, which sets it apart from regular SMOTE, which depends on KNN and Euclidean distances to generate synthetic data points. This quantum methodology enables the direct generation of synthetic data points from instances belonging to the minority class, thereby removing the need for interpolation based on neighbouring data points. The technique has customizable hyperparameters, such as the rotation angle, minority percentage, and splitting factor, which enable users to fine-tune the production of synthetic data in order to accurately replicate the distribution of the original data and efficiently mitigate imbalances in the dataset. Quantum SMOTE exhibits efficacy in data representation and scalability. The technique employs a compact swap test approach, which enables the management of a substantial number of variables using a relatively small number of qubits. For example, it can effectively handle 1024 variables with only 10 qubits, demonstrating a substantial decrease in quantum resource needs. Quantum SMOTE uses low-depth quantum circuits to mitigate the typical challenges of noise and decoherence in quantum computing.

This approach enhances the dependability of data augmentation using quantum technology.

The approach has shown a beneficial impact on the precision-recall scores of several machine learning algorithms, such as Logistic Regression and Random Forest. This compatibility is shown in modern machine-learning scenarios, including coping with imbalances in class distribution.

chapter 5: Future directions

Future research should aim to optimize the Quantum SMOTE algorithm by investigating more quantum phenomena or using more advanced quantum operations to improve the production of synthetic data points and increase its efficiency. Conducting experiments using Quantum SMOTE on a wider range of datasets, including bigger ones, would provide more evidence to support its efficacy and resilience. Additionally, it would be advantageous to assess its effectiveness using other machine learning models, especially ones that are not often linked to issues of class imbalance. By combining Quantum SMOTE with other quantum computing approaches or hybrid quantum-classical machine learning frameworks, it is possible to create even more potent tools for addressing intricate data science challenges. An in-depth examination of quantum resource allocation for the management of high-dimensional data might provide valuable insights for enhancing the accessibility and practicality of quantum machine learning in a wide range of applications.

Quantum SMOTE is a potential fusion of quantum computing with data science that offers a creative solution to the ongoing issue of class imbalance in machine learning. The development of this technology creates new opportunities to use quantum phenomena in practical applications of data science in the real world.

6.2 Concluding Remarks

It is important to note that this thesis makes a substantial contribution to the integration of quantum computing with data science, especially in terms of tackling optimization issues and class imbalances in datasets. The developed quantum machine learning techniques enhance the effectiveness of tackling such issues and pave the way for utilizing quantum phenomena in practical data science applications. The ongoing development of these quantum approaches is anticipated

to result in quantum computing systems that are more robust, efficient, and practical. These systems have the potential to outperform classical methods in a variety of areas of data processing and optimization.

APPENDIX



APPENDIX

BIBLIOGRAPHY

- [1] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*. Cambridge University Press, Dec 2010. [Online]. Available: [http://books.google.ie/books?id=-s4DEy7o-a0C&printsec=frontcover&dq=Nielsen,+M.+A.,+%26+Chuang,+I.+L.+\(2010\).+Quantum+Computation+and+Quantum+Information.+Cambridge+University+Press&hl=&cd=1&source=gbs_api](http://books.google.ie/books?id=-s4DEy7o-a0C&printsec=frontcover&dq=Nielsen,+M.+A.,+%26+Chuang,+I.+L.+(2010).+Quantum+Computation+and+Quantum+Information.+Cambridge+University+Press&hl=&cd=1&source=gbs_api)
- [2] J. Preskill, “Quantum Computing in the NISQ era and beyond,” *Quantum*, vol. 2, p. 79, Aug. 2018. [Online]. Available: <https://doi.org/10.22331/q-2018-08-06-79>
- [3] K. Bharti, A. Cervera-Lierta, T. H. Kyaw, T. Haug, S. Alperin-Lea, A. Anand, M. Degroote, H. Heimonen, J. S. Kottmann, T. Menke, W.-K. Mok, S. Sim, L.-C. Kwek, and A. Aspuru-Guzik, “Noisy intermediate-scale quantum algorithms,” *Rev. Mod. Phys.*, vol. 94, p. 015004, Feb 2022. [Online]. Available: <https://link.aps.org/doi/10.1103/RevModPhys.94.015004>
- [4] A. Peruzzo, J. R. McClean, P. Shadbolt, M. Yung, X. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. O’Brien, “A variational eigenvalue solver on a photonic quantum processor,” *Nature Communications*, vol. 5, no. 1, Jul 2014. [Online]. Available: <https://www.nature.com/articles/ncomms5213>
- [5] E. Farhi, J. Goldstone, S. Gutmann, and M. Sipser, “Quantum Computation by Adiabatic Evolution,” Jan. 2000. [Online]. Available: <https://arxiv.org/abs/quant-ph/0001106v1>

- [6] L. K. Grover, “A fast quantum mechanical algorithm for database search,” 1996.
- [7] P. W. Shor, “Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer,” *SIAM Journal on Computing*, vol. 26, no. 5, p. 1484–1509, Oct. 1997. [Online]. Available: <http://dx.doi.org/10.1137/S0097539795293172>
- [8] F. e. a. Arute, “Quantum supremacy using a programmable superconducting processor,” *Nature*, vol. 505, no. 7779, Oct 2019. [Online]. Available: <https://www.nature.com/articles/s41586-019-1666-5>
- [9] V. Havlicek, A. D. Corcoles, K. Temme, A. W. Harrow, A. Kandala, J. M. Chow, and J. M. Gambetta, “Supervised learning with quantum enhanced feature spaces,” *Nature*, vol. 567, pp. 209–212, 2019. [Online]. Available: <https://doi.org/10.1038/s41586-019-0980-2>
- [10] E. Farhi, J. Goldstone, and S. Gutmann, “A quantum approximate optimization algorithm,” *arXiv*, Nov. 2014. [Online]. Available: <https://arxiv.org/abs/1411.4028v1>
- [11] N. Guerrero, “Solving Combinatorial Optimization Problems using the Quantum Approximation Optimization Algorithm,” *Theses and Dissertations*, Mar. 2020. [Online]. Available: <https://scholar.ait.edu/etd/3263>
- [12] R. Hoque, “The quantum approximate optimization algorithm,” p. 14. [Online]. Available: <https://ryanhoque.github.io/data/QAOA.pdf>
- [13] J. R. McClean, M. E. Kimchi-Schwartz, J. Carter, and W. A. de Jong, “Hybrid quantum-classical hierarchy for mitigation of decoherence and determination of excited states,” *Phys. Rev. A*, vol. 95, p. 042308, Apr 2017. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevA.95.042308>
- [14] P. J. J. O’Malley, R. Babbush, I. D. Kivlichan, J. Romero, J. R. McClean, R. Barends, J. Kelly, P. Roushan, A. Tranter, N. Ding, B. Campbell, Y. Chen, Z. Chen, B. Chiaro, A. Dunsworth, A. G. Fowler, E. Jeffrey,

- E. Lucero, A. Megrant, J. Y. Mutus, M. Neeley, C. Neill, C. Quintana, D. Sank, A. Vainsencher, J. Wenner, T. C. White, P. V. Coveney, P. J. Love, H. Neven, A. Aspuru-Guzik, and J. M. Martinis, “Scalable quantum simulation of molecular energies,” *Phys. Rev. X*, vol. 6, p. 031007, Jul 2016. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevX.6.031007>
- [15] S. Endo, Z. Cai, S. C. Benjamin, and X. Yuan, “Hybrid quantum-classical algorithms and quantum error mitigation,” *Journal of the Physical Society of Japan*, vol. 90, no. 3, p. 032001, Mar. 2021. [Online]. Available: <http://dx.doi.org/10.7566/JPSJ.90.032001>
- [16] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio, and P. J. Coles, “Variational quantum algorithms,” *Nature Reviews Physics*, vol. 3, no. 9, p. 625–644, Aug. 2021. [Online]. Available: <http://dx.doi.org/10.1038/s42254-021-00348-9>
- [17] A. Kandala, A. Mezzacapo, K. Temme, M. Takita, M. Brink, J. M. Chow, and J. M. Gambetta, “Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets,” *Nature*, vol. 549, no. 7671, pp. 242–246, Sep. 2017, publisher: Nature Publishing Group. [Online]. Available: <https://www.nature.com/articles/nature23879>
- [18] A. Montanaro, “Quantum algorithms: an overview,” *npj Quantum Information*, vol. 2, no. 1, p. 15023, Jan. 2016. [Online]. Available: <https://doi.org/10.1038/npjqi.2015.23>
- [19] S. Jordan, “<https://quantumalgorithmzoo.org/>.” [Online]. Available: <https://quantumalgorithmzoo.org/#ONML>
- [20] L. K. Grover, *A fast quantum mechanical algorithm for database search*, May 1996. [Online]. Available: <https://arxiv.org/abs/quant-ph/9605043v3>
- [21] V. Dasari, M. S. Im, and L. Beshaj, *Solving machine learning optimization problems using quantum computers*, M. Blowers, R. D. Hall, and V. R. Dasari, Eds. Online Only,

- United States: SPIE, apr 2020. [Online]. Available: <https://www.spiedigitallibrary.org/conference-proceedings-of-spie/11419/2565038/Solving-machine-learning-optimization-problems-using-quantum-computers/10.1117/12.2565038.full>
- [22] E. National Academies of Sciences, “Chapter: 3 Quantum Algorithms and Applications,” in *Quantum Computing: Progress and Prospects (2019)*. [Online]. Available: <https://www.nap.edu/catalog/25196/quantum-computing-progress-and-prospects>
- [23] S. Harwood, C. Gambella, D. Trenev, A. Simonetto, D. Bernal, and D. Greenberg, “Formulating and Solving Routing Problems on Quantum Computers,” *IEEE Transactions on Quantum Engineering*, vol. 2, pp. 1–17, 2021. [Online]. Available: <https://ieeexplore.ieee.org/document/9314905>
- [24] K. Srinivasan, S. Satyajit, B. K. Behera, and P. K. Panigrahi, “Efficient quantum algorithm for solving travelling salesman problem: An ibm quantum experience,” May 2018. [Online]. Available: <https://arxiv.org/abs/1805.10928v1>
- [25] S. Feld, C. Roch, T. Gabor, C. Seidel, F. Neukart, I. Galter, W. Mauerer, and C. Linnhoff-Popien, *A Hybrid Solution Method for the Capacitated Vehicle Routing Problem Using a Quantum Annealer*, 2019, vol. 6. [Online]. Available: <https://www.frontiersin.org/article/10.3389/fict.2019.00013>
- [26] M. Nazari, A. Oroojlooy, L. Snyder, and M. Takac, *Reinforcement Learning for Solving the Vehicle Routing Problem*. Curran Associates, Inc., 2018, vol. 31.
- [27] U. Azad, B. K. Behera, E. A. Ahmed, P. K. Panigrahi, and A. Farouk, “Solving Vehicle Routing Problem Using Quantum Approximate Optimization Algorithm,” *IEEE Transactions on Intelligent Transportation Systems*, 2022. [Online]. Available: <https://ieeexplore.ieee.org/document/9774961>
- [28] C. Papalitsas, T. Andronikos, K. Giannakis, G. Theocharopoulou, and S. Fanarioti, “A QUBO Model for the Traveling Salesman Problem with

- Time Windows,” *Algorithms*, vol. 12, no. 11, 2019. [Online]. Available: <https://www.mdpi.com/1999-4893/12/11/224>
- [29] F. Glover, G. Kochenberger, M. Ma, and Y. Du, “Quantum Bridge Analytics II: QUBO-Plus, network optimization and combinatorial chaining for asset exchange,” *4OR*, vol. 18, no. 4, pp. 387–417, 2020, publisher: Springer. [Online]. Available: https://ideas.repec.org/a/spr/ajoor/v18y2020i4d10.1007_s10288-020-00464-9.html
- [30] G. Kochenberger, J.-K. Hao, F. Glover, M. Lewis, Z. Lu, H. Wang, and Y. Wang, “The unconstrained binary quadratic programming problem: A survey,” *Journal of Combinatorial Optimization*, vol. 28, Jul. 2014. [Online]. Available: <https://link.springer.com/article/10.1007/s10878-014-9734-0>
- [31] H. Irie, G. Wongpaisarnsin, M. Terabe, A. Miki, and S. Taguchi, “Quantum Annealing of Vehicle Routing Problem with Time, State and Capacity,” in *Quantum Technology and Optimization Problems*, ser. Lecture Notes in Computer Science, S. Feld and C. Linnhoff-Popien, Eds. Cham: Springer International Publishing, 2019, pp. 145–156. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-030-14082-3_13
- [32] A. Crispin and A. Syrichas, “Quantum Annealing Algorithm for Vehicle Scheduling,” in *2013 IEEE International Conference on Systems, Man, and Cybernetics*, Oct. 2013, pp. 3523–3528, iSSN: 1062-922X. [Online]. Available: <https://ieeexplore.ieee.org/document/6722354>
- [33] F. E. Office, “Application of Digital Annealer for Faster Combinatorial Optimization,” *FUJITSU Sci. Tech. J.*, vol. 55, no. 2, p. 7, 2019. [Online]. Available: <https://www.fujitsu.com/global/documents/about/resources/publications/fstj/archives/vol55-2/paper12.pdf>
- [34] E. Campos, D. Rabinovich, V. Akshay, and J. Biamonte, “Training saturation in layerwise quantum approximate optimization,” *Phys. Rev. A*, vol. 104, no. 3, p. L030401, Sep. 2021, publisher: American Physical Society. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevA.104.L030401>

- [35] S. Wang, E. Fontana, M. Cerezo, K. Sharma, A. Sone, L. Cincio, and P. J. Coles, “Noise-induced barren plateaus in variational quantum algorithms,” *Nat Commun*, vol. 12, no. 1, p. 6961, Nov. 2021, number: 1 Publisher: Nature Publishing Group. [Online]. Available: <https://www.nature.com/articles/s41467-021-27045-6>
- [36] J. Marshall, F. Wudarski, S. Hadfield, and T. Hogg, “Characterizing local noise in QAOA circuits,” *IOP SciNotes*, vol. 1, no. 2, p. 025208, aug 2020. [Online]. Available: <https://doi.org/10.1088/2633-1357/abb0d7>
- [37] W. Lavrijsen, A. Tudor, J. Muller, C. Iancu, and W. de Jong, “Classical Optimizers for Noisy Intermediate-Scale Quantum Devices,” in *2020 IEEE International Conference on Quantum Computing and Engineering (QCE)*, Oct. 2020, pp. 267–277. [Online]. Available: <https://arxiv.org/abs/2004.03004>
- [38] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio, and P. J. Coles, “Variational Quantum Algorithms,” *Nat Rev Phys*, vol. 3, no. 9, pp. 625–644, Sep. 2021, arXiv: 2012.09265. [Online]. Available: <http://arxiv.org/abs/2012.09265>
- [39] T. Albash and D. A. Lidar, “Adiabatic quantum computation,” *Rev. Mod. Phys.*, vol. 90, no. 1, p. 015002, Jan. 2018, publisher: American Physical Society. [Online]. Available: <https://link.aps.org/doi/10.1103/RevModPhys.90.015002>
- [40] E. K. Grant and T. S. Humble, “Adiabatic Quantum Computing and Quantum Annealing,” jul 2020, iSBN: 9780190871994. [Online]. Available: <https://oxfordre.com/physics/view/10.1093/acrefore/9780190871994.001.\0001/acrefore-9780190871994-e-32>
- [41] Y. Sun, J.-Y. Zhang, M. S. Byrd, and L.-A. Wu, “Adiabatic Quantum Simulation Using Trotterization,” *arXiv:1805.11568 [quant-ph]*, Jun. 2018, arXiv: 1805.11568. [Online]. Available: <http://arxiv.org/abs/1805.11568>
- [42] L. Zhou, S.-T. Wang, S. Choi, H. Pichler, and M. D. Lukin, “Quantum Approximate Optimization Algorithm: Performance, Mechanism, and

- Implementation on Near-Term Devices,” *Phys. Rev. X*, vol. 10, no. 2, p. 021067, Jun. 2020, publisher: American Physical Society. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevX.10.021067>
- [43] S. P. Singh, *The Ising Model: Brief Introduction and Its Application*. IntechOpen, Feb. 2020, publication Title: Solid State Physics - Metastable, Spintronics Materials and Mechanics of Deformable Bodies - Recent Progress. [Online]. Available: <https://www.intechopen.com/chapters/71210>
- [44] S. G. BRUSH, “History of the lenz-ising model,” *Rev. Mod. Phys.*, vol. 39, pp. 883–893, Oct 1967. [Online]. Available: <https://link.aps.org/doi/10.1103/RevModPhys.39.883>
- [45] A. Lucas, “Ising formulations of many NP problems,” *Frontiers in Physics*, vol. 2, 2014. [Online]. Available: <https://www.frontiersin.org/article/10.3389/fphy.2014.00005>
- [46] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. O’Brien, “A variational eigenvalue solver on a photonic quantum processor,” *Nature Communications*, vol. 5, no. 1, p. 4213, Jul. 2014. [Online]. Available: <https://doi.org/10.1038/ncomms5213>
- [47] “Qiskit: Vehicle routing.” [Online]. Available: https://qiskit.org/ecosystem/optimization/tutorials/07_examples_vehicle_routing.html
- [48] P. Date, R. Patton, C. Schuman, and T. Potok, “Efficiently embedding QUBO problems on adiabatic quantum computers,” *Quantum Information Processing*, vol. 18, no. 4, p. 117, Mar. 2019. [Online]. Available: <https://doi.org/10.1007/s11128-019-2236-3>
- [49] G. G. Guerreschi, “Solving Quadratic Unconstrained Binary Optimization with divide-and-conquer and quantum algorithms,” *arXiv:2101.07813 [quant-ph]*, Jan. 2021, arXiv: 2101.07813. [Online]. Available: <http://arxiv.org/abs/2101.07813>
- [50] M. S. A. *et al.*, “Qiskit: An open-source framework for quantum computing,” 2021. [Online]. Available: <https://github.com/Qiskit/qiskit/tree/0.25.0>

- [51] S. Ahadpour, F. Mirmasoudi, S. Ahadpour, and F. Mirmasoudi, "The role of noisy channels in quantum teleportation," *Revista mexicana de fisica*, vol. 66, no. 3, pp. 378–387, Jun. 2020. [Online]. Available: http://www.scielo.org.mx/scielo.php?script=sci_abstract&pid=S0035-001X2020000300378&lng=es&nrm=iso&tlng=en
- [52] R.-G. Zhou, Y.-N. Zhang, R. Xu, C. Qian, and I. Hou, "Asymmetric Bidirectional Controlled Teleportation by Using Nine-Qubit Entangled State in Noisy Environment," *IEEE Access*, vol. 7, pp. 75 247–75 264, 2019. [Online]. Available: <https://ieeexplore.ieee.org/document/8726306>
- [53] X. Bonet-Monroig, H. Wang, D. Vermetten, B. Senjean, C. Moussa, T. B. \check{S} ck, V. Dunjko, and T. E. O. \check{A}
- D. C. Liu and J. Nocedal, "On the limited memory BFGS method for large scale optimization," *Mathematical Programming*, vol. 45, no. 1-3, pp. 503–528, 1989.
- D. Kraft, "A software package for sequential quadratic programming," DFVLR Oberpfaffenhofen, Germany, DFVLR-FB 88-28, 1988.
- J. C. Spall, "Multivariate stochastic approximation using a simultaneous perturbation gradient approximation," *IEEE Transactions on Automatic Control*, vol. 37, no. 3, pp. 332–341, 1992.
- K. Vu, R. U. Pedersen, and T. S. Steingrimsson, "Comparative benchmarking of optimization algorithms including cobyla for variational quantum algorithms," *arXiv preprint arXiv:2201.02316*, 2022.
- R. Horodecki, P. Horodecki, M. Horodecki, and K. Horodecki, "Quantum entanglement," *Rev. Mod. Phys.*, vol. 81, pp. 865–942, Jun 2009. [Online]. Available: <https://link.aps.org/doi/10.1103/RevModPhys.81.865>
- S. N. Kumar and R. Panneerselvam, "A survey on the vehicle routing problem and its variants," *Intelligent Information Management*, vol. 4, no. 3, p. 9, 2012. [Online]. Available: https://www.scirp.org/html/4-8701154_19355.htm
- N. Mohanty, B. K. Behera, and C. Ferrie, "Analysis of the vehicle routing problem solved via hybrid quantum algorithms in the presence of noisy channels," *IEEE*

Transactions on Quantum Engineering, vol. 4, p. 1, 2023. [Online]. Available: <http://dx.doi.org/10.1109/TQE.2023.3303989>

P. Rebentrost, M. Mohseni, and S. Lloyd, “Quantum support vector machine for big data classification,” *Physical Review Letters*, vol. 113, no. 13, sep 2014. [Online]. Available: <https://doi.org/10.1103/PhysRevLett.113.130503>

A. Kariya and B. K. Behera, “Investigation of quantum support vector machine for classification in nisq era,” 2021.

R. Leporini and D. Pastorello, “Support vector machines with quantum state discrimination,” *Quantum Rep.*, vol. 3, no. 3, pp. 482–499, 2021. [Online]. Available: <https://doi.org/10.3390/quantum3030032>

G. Gentinetta, A. Thomsen, D. Sutter, and S. Woerner, “The complexity of quantum support vector machines,” 2022.

J. R. Glick, T. P. Gujarati, A. D. Corcoles, Y. Kim, A. Kandala, J. M. Gambetta, and K. Temme, “Covariant quantum kernels for data with group structure,” 2022.

I. F. Araujo, D. K. Park, F. Petruccione, and A. J. da Silva, “A divide-and-conquer algorithm for quantum state preparation,” *Scientific Reports*, vol. 11, no. 1, p. 6329, Mar. 2021. [Online]. Available: <https://doi.org/10.1038/s41598-021-85474-1>

R. LaRose and B. Coyle, “Robust data encodings for quantum classifiers,” *Phys. Rev. A*, vol. 102, p. 032420, Sep 2020. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevA.102.032420>

Qiskit, “Lecture 5.1 - Building a Quantum Classifier,” Oct. 2021. [Online]. Available: <https://www.youtube.com/watch?v=-sxlXNz7ZxU>

“Paddle Quantum.” [Online]. Available: <https://qml.baidu.com/tutorials/machine-learning/encoding-classical-data-into-quantum-states.html>

G. Haixiang, L. Yijing, J. Shang, G. Mingyun, H. Yuanyue, and G. Bing, “Learning from class-imbalanced data: Review of methods and applications,” *Expert Systems with Applications*, vol. 73, pp. 220–239, 2017. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0957417416307175>

- M. Blaszczyk and J. Jedrzejowicz, "Framework for imbalanced data classification," *Procedia Computer Science*, vol. 192, pp. 3477–3486, 2021. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1877050921018603>
- S. Wang, Y. Dai, J. Shen, and J. Xuan, "enResearch on expansion and classification of imbalanced data based on SMOTE algorithm," *enScientific Reports*, vol. 11, no. 1, p. 24039, Dec. 2021, number: 1 Publisher: Nature Publishing Group. [Online]. Available: <https://www.nature.com/articles/s41598-021-03430-5>
- N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "enSMOTE: Synthetic Minority Over-sampling Technique," *enJournal of Artificial Intelligence Research*, vol. 16, pp. 321–357, Jun. 2002. [Online]. Available: <https://doi.org/10.1613/jair.953>
- A. Fernandez, S. Garcia, F. Herrera, and N. V. Chawla, "enSMOTE for Learning from Imbalanced Data: Progress and Challenges, Marking the 15-year Anniversary," *enJournal of Artificial Intelligence Research*, vol. 61, pp. 863–905, Apr. 2018. [Online]. Available: <https://jair.org/index.php/jair/article/view/11192>
- M. Mukherjee and M. Khushi, "Smote-enc: A novel smote-based method to generate synthetic data for nominal and continuous features," *Applied System Innovation*, vol. 4, no. 1, 2021. [Online]. Available: <https://www.mdpi.com/2571-5577/4/1/18>
- C. Seiffert, T. M. Khoshgoftaar, J. V. Hulse, and A. Napolitano, "Rusboost: A hybrid approach to alleviating class imbalance," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 40, pp. 185–197, 2010. [Online]. Available: <https://ieeexplore.ieee.org/document/5299216>
- N. Chawla, A. Lazarevic, L. Hall, and K. Bowyer, "Smoteboost: Improving prediction of the minority class in boosting," vol. 2838, 01 2003, pp. 107–119. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-540-39804-2_12
- J. H. Joloudari, A. Marefat, M. A. Nematollahi, S. S. Oyelere, and S. Hussain, "Effective class-imbalance learning based on smote and convolutional neural networks," *Applied Sciences*, vol. 13, no. 6, 2023. [Online]. Available: <https://www.mdpi.com/2076-3417/13/6/4006>

“enTelco Customer Churn.” [Online]. Available: <https://www.kaggle.com/datasets/blastchar/telco-customer-churn>

H. Han, W.-Y. Wang, and B.-H. Mao, “enBorderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning,” in *enAdvances in Intelligent Computing*, ser. Lecture Notes in Computer Science, D.-S. Huang, X.-P. Zhang, and G.-B. Huang, Eds. Berlin, Heidelberg: Springer, 2005, pp. 878–887. [Online]. Available: https://link.springer.com/chapter/10.1007/11538059_91

H. He, Y. Bai, E. A. Garcia, and S. Li, “Adasyn: Adaptive synthetic sampling approach for imbalanced learning,” in *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, 2008, pp. 1322–1328. [Online]. Available: <https://ieeexplore.ieee.org/document/4633969>

G. E. A. P. A. Batista, R. C. Prati, and M. C. Monard, “A study of the behavior of several methods for balancing machine learning training data,” *ACM SIGKDD Explorations Newsletter*, vol. 6, no. 1, pp. 20–29, Jun. 2004. [Online]. Available: <https://doi.org/10.1145/1007730.1007735>

“Two modifications of cnn,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-6, no. 11, pp. 769–772, 1976. [Online]. Available: <https://ieeexplore.ieee.org/document/4309452>

L. Demidova and I. Klyueva, “Svm classification: Optimization with the smote algorithm for the class imbalance problem,” in *2017 6th Mediterranean Conference on Embedded Computing (MECO)*, 2017, pp. 1–4. [Online]. Available: <https://ieeexplore.ieee.org/document/7977136>

Calculate quantum euclidean distance with qiskit. Medium. [Online]. Available: <https://medium.com/qiskit/calculate-quantum-euclidean-distance-with-qiskit-df85525ab4>

M. Martínez-Felipe, J. Montiel-Pérez, V. Onofre, A. Maldonado-Romo, and R. Young, “Quantum block-matching algorithm using dissimilarity measure,” in *Service-Oriented Computing – ICSOC 2023 Workshops*, F. Monti, P. Plebani, N. Moha, H.-y. Paik, J. Barzen, G. Ramachandran, D. Bianchini, D. A. Tamburri, and M. Mecella, Eds. Singapore: Springer Nature Singapore, 2024, pp. 185–196.

BIBLIOGRAPHY

L. Breiman, “enRandom Forests,” *enMachine Learning*, vol. 45, no. 1, pp. 5–32, Oct. 2001. [Online]. Available: <https://doi.org/10.1023/A:1010933404324>

D. W. H. Jr., S. Lemeshow, and R. X. Sturdivant, “en-usApplied Logistic Regression, 3rd Edition | Wiley.” [Online]. Available: <https://onlinelibrary.wiley.com/doi/book/10.1002/9781118548387>

T. Cover and P. Hart, “Nearest neighbor pattern classification,” *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967. [Online]. Available: <https://ieeexplore.ieee.org/document/1053964>

N. S. Altman, “An introduction to kernel and nearest-neighbor nonparametric regression,” *The American Statistician*, vol. 46, no. 3, pp. 175–185, 1992. [Online]. Available: <https://www.tandfonline.com/doi/abs/10.1080/00031305.1992.10475879>

R. Blagus and L. Lusa, “SMOTE for high-dimensional class-imbalanced data,” *BMC Bioinformatics*, vol. 14, no. 1, p. 106, Mar. 2013. [Online]. Available: <https://doi.org/10.1186/1471-2105-14-106>