SPATIAL QUANTUM COMPUTATION IN GRAPH OPTIMIZATION PROBLEMS IN TRANSPORTATION APPLICATIONS

AMIRHOSSEIN NOURBAKHSHREZAEI

A DISSERTATION SUBMITTED TO THE FACULTY OF GRADUATE STUDIES IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

GRADUATE PROGRAM IN EARTH AND SPACE SCIENCE

YORK UNIVERSITY
TORONTO, ONTARIO

August 8, 2024

Abstract

Efficient transportation management is an important aspect of urban sustainability, impacting economic growth, environmental sustainability, and quality of life. This research explores the potential of Quantum Computing (QC) to address spatial optimization problems in transportation systems. By leveraging the principles of quantum mechanics, this research aims to enhance the efficiency and effectiveness of transportation networks through QC-based solutions to challenges such as reducing the spread of viruses on road networks, dynamic rebalancing of Bike Sharing Systems (BSS), clustering of BSS stations, and Feature Selection (FS) for predictive models. The study begins by examining the potential of QC in solving combinatorial optimization problems, specifically focusing on minimizing exposure to COVID-19 during city journeys. A novel QC-based approach is developed for the dynamic rebalancing of BSS, which is a critical component of BSS management. The research further explores the clustering of BSS stations using Quantum Machine Learning (QML) techniques to enhance system management and improve user satisfaction. Additionally, this research introduces a QC-based FS method to improve the accuracy of predictive models, utilizing spatial data to optimize station placement and service availability. The proposed methodologies are validated through different experiments and real-world data, demonstrating significant improvements in computational efficiency and solution quality compared to traditional methods. Overall, this research advances the application of QC in transportation systems, providing a QC-based framework for future studies and practical implementations in urban transportation management. It highlights the transformative potential of QC in addressing pressing urban mobility challenges, paving the way for more sustainable and efficient transportation networks.

Acknowledgments

I would like to express my deepest gratitude to my supervisor, Prof. Mojgan Jadidi, for her support, guidance, and encouragement throughout my PhD journey. Her insightful advice and invaluable feedback have been instrumental in shaping this dissertation.

I am also thankful to my committee members, Prof. Gunho Sohn and Prof. Manos Papagelis, for their constructive feedback and guidance during the Research Evaluation Course (REC) meetings. Their expertise and suggestions have greatly contributed to the refinement and success of this research.

In addition, I am also thankful to Dr. Kyarash Shahriari (Intelligent Data Analytics Inc.) and Prof. Seyed Soheil Mansouri (Denmark Technical University) for their support in opening the opportunity to learn Quantum Computing and adopting it in my research.

I thank Geospatial Visual Analytics (GeoVA) Lab members: Dr. Hamid Kiavarz, Mr. Aman Usmani, Mr. Xuyang Han, Ms. Sepideh Dibadin, Dr. Sarah Keykhosravi, and Mr. Damith Tennakoon for their endless encouragement and constructive feedback during our group meetings. I express my sincere acknowledgement to my co-author Mr. Soroush Sheikh Gargar for his expertise in machine learning.

I extend my heartfelt thanks to my friends and family for their continuous support and encouragement. Special thanks go to my wife, Sahar, whose love, patience, and understanding have been my source of strength throughout this journey. I also want to express my gratitude to my mother and sisters, whose encouragement has been invaluable.

I honor the memory of my father, whose wisdom and guidance continue to inspire me.

Thank you all for being part of this journey and for your contributions to my success.

Contents

List of Tables

List of Figures

Abbreviations

AMC Automatic Modulation Classification

BSS Bike Sharing Systems

CFS Correlation-based Feature Selection

FS Feature Selection

LR Linear Regression

MBQC Measurement-Based Quantum Computing

MI Mutual Information

ML Machine Learning

QA Quantum Annealing

QAOA Quantum Approximate Optimization Algorithm

QC Quantum Computing

QFS Quantum Feature Selection

QIGA Quantum-Inspired Genetic Algorithm

QML Quantum Machine Learning

QUBO Quadratic Unconstrained Binary Optimization

RF Random Forest

RFE Recursive Feature Elimination

RMSE Root Mean Squared Error

RNN Recurrent Neural Networks

VQE Variational Quantum Eigensolver

XGBoost Extreme Gradient Boosting

Chapter 1

Introduction

1.1 Background and Motivation

Efficient transportation management is crucial for the sustainability and functionality of urban areas, impacting economic growth, environmental sustainability, and quality of life [2]. The complexity and dynamic nature of transportation systems necessitate advanced management strategies to optimize operations and address various challenges [3]. Efficient transportation management directly influences the economic vitality of a region [4]. Congestion and delays can lead to significant economic losses due to increased travel times, higher fuel consumption, and reduced productivity. Effective transportation management can mitigate these environmental impacts by promoting multi-modal transportation modes leading to the use of public transportation, carpooling, and non-motorized transport modes such as cycling and walking [5]. Implementing strategies to improve the quality of transportation and encourage increased use of multi-modal transportation can also lead to lower emissions.

Spatial data plays a critical role in transportation management by providing spatial context to various elements of the transportation network [6]. This data includes information about the location of entities such as trajectories, enabling detailed analysis and optimization of transportation services[7]. For instance, understanding the spatial distribution of transportation demand can help

in optimizing the placement of charging stations for electrical vehicles and improving service availability [8]. Spatial analysis also facilitates real-time monitoring and management of transportation systems, allowing for timely interventions in response to changing conditions [9].

Making decisions in transportation requires the consideration of both individual and public perspectives, each influencing the other[10]. On the individual level, travelers decide when and how to move from one location to another, choosing their routes based on personal schedules, preferences, and convenience. These decisions are influenced by factors such as traffic conditions, availability of different modes of transport, weather conditions, and the time of day. On the public level, transportation planning and policy-making play a crucial role. Authorities establish guidelines, regulations, and infrastructure development plans that directly impact individual choices [11]. This includes the design and maintenance of roads, the scheduling and routing of public transit, the implementation of traffic management systems, and the promotion of sustainable transport options like cycling and walking [12],[13]. The interplay between these individual and public decisions creates a complex environment where various aspects must be considered. For example, optimizing public transportation services can encourage individuals to use them more frequently, reducing congestion [14] and environmental impact [15]. Conversely, understanding individual travel behaviors and preferences can help public planners design more effective and user-friendly transportation systems [16]. Effective transportation management requires a holistic approach that integrates individual decisions with public policies and infrastructure planning by considering the needs and behaviors of individuals alongside broader public objectives. Traditionally, most transportation problems are effectively converted into graph models [17]. Graph optimization is a crucial approach to solving complex transportation problems. It involves modeling transportation networks as graphs, where nodes represent locations (e.g., bus stops, intersections) and edges represent connections (e.g., routes, roads) [18]. By leveraging the mathematical properties of graphs, complex transportation networks can be analyzed and optimized, leading to improved efficiency, reduced costs, and enhanced service quality [19]. For instance finding the shortest or most efficient path between points, considering constraints like traffic and capacity [20]. Another well-known problem is

about assigning vehicles to routes or schedules to maximize efficiency and minimize waiting times [21]. The multi-modal aspect of transportation has also another layer of complexity, as it involves coordinating different modes of transport, such as buses, trains, and bikes, to provide proper and efficient travel options. Each mode has its own set of variables and constraints. BSS, as a component of multi-modal transportation networks, present unique optimization challenges. These include the placement and redistribution of bikes, balancing supply and demand across various stations, analyzing the status of the stations, and predicting ridership. Addressing these challenges effectively requires advanced computational techniques capable of handling the intricate and dynamic nature of BSS within the broader transportation framework. Traditional computational methods can become infeasible in solving those challenges as the scale and complexity of the problem grow. Optimization problems in transportation are known for their significant computational complexity [22] which often fall into the category of Non-deterministic Polynomial-time hard (NP-hard) problems [23]. NP-hard problems are a class of problems in computational complexity theory that are at least as difficult as the hardest problems in Non-deterministic Polynomial-time (NP). NP-hard problems cannot be solved using deterministic approaches due to their inherent computational complexity [24]. This means that as the size of the input grows, the time required to solve the problem increases exponentially. This exponential growth makes it impractical for deterministic algorithms to find exact solutions within a feasible time frame for large instances. Hence, deterministic approaches are often replaced by heuristic [25] and meta-heuristic [26] methods, which can provide approximate solutions more efficiently to transportation systems management. Today, the rise of Quantum Computing (QC) to solve NP-hard problems as a meta-heuristic approach makes them a promising avenue to explore for such problems in transportation applications. Given the complexities and computational cost of NP-hard problems, exploring the potential of QC as another meta-heuristic solution is highly promising.

The term QC was first coined by Paul Benioff in 1979 when he constructed a microscopic quantum mechanical model to perform computation [27]. In the early 70's, Stephen Wiesner proposed his idea of "Conjugate Coding," which is a cryptographic tool, and submitted an article to the IEEE

Transactions on Information Theory. His paper was rejected because it seemed incomprehensible to the referees [28]. In October 1979, Benioff and Wiesner found a way to use Wiesner's coding scheme with the concept of public-key cryptography [29]. Finally, the result of their collaboration led to the generation of new topics in the field, such as teleporting an unknown quantum state by using dual classical and Einstein-Podolsky-Rosen channels [30], entanglement distillation [31], strengths and weaknesses of QC [32], and quantum cryptography [33]. Richard Feynman's description of how to accurately calculate quantum physical systems on a quantum computer architecture can be seen as the first description of a quantum computer architecture. He submitted the description on the 7th of May 1981 to the International Journal of Theoretical Physics. The article was then published in volume 21, June 1982 [34]. David Deutsch described in 1985 his idea of a universal quantum computer. He mentioned that a computing machine with quantum theory characteristics can be built and that it can have many remarkable properties that cannot be found in Turing machines [35]. QC has been studied for decades and ranges from information theory to details of hardware technologies, computational models implemented on specific hardware, and problem formulations that can be solved on a certain hardware type. It has applications in nearly every field that contains or utilizes computations with high complexity. Quantum computers have the potential to impact many aspects of current domains of science, including transportation applications.

One of the main goals of QC is called "quantum supremacy" or "quantum advantage." John Preskill published a paper in November 2012 about quantum supremacy and the consequences it will have for several critical applications in society, such as cryptography and optimization [36]. In October 2019, Google claimed to have achieved quantum supremacy using a programmable superconducting processor [37]. Google scientists used a quantum processor called "Sycamore" to sample the output of a pseudo-random quantum circuit. They used 53 qubits representing a state space of 2^{53} dimensions. The process took about 200 seconds to solve a complex computation, while that process would have taken about 10,000 years with a classical computer [37]. However, IBM stated that the computation of the Google experiment could be performed on a classical computer in 2.5 days rather than 10,000 years [38]. Having established the context and significance of the research,

the following section now turn the attention to the specific problems we aim to address.

1.2 Problem Definition

Transportation systems face a multitude of challenges that impact their efficiency and effectiveness [39]. Congestion is a significant issue, leading to delays, increased travel times, and higher fuel consumption[40]. The environmental impact of transportation, particularly in terms of air pollution and greenhouse gas emissions, is another major concern [41],[42]. Transportation systems are inherently complex due to the variety of interrelated components and stakeholders involved [43]. These systems must work in different environments each with unique features and behaviors. Additionally, transportation systems operate within an environment influenced by fluctuating demand patterns [44], evolving infrastructure, and external factors such as spatial features[45],[46], [47], and weather conditions [48]. The interplay between these elements creates a web of dependencies that complicate the planning, management, and optimization of transportation networks. Additionally, the dynamic nature of transportation systems causes uncertainty in the system [49]. The term ”dynamic nature” refers to their constant state of flux, influenced by real-time changes in demand, traffic conditions, and environmental factors. This dynamism requires adaptive management strategies that can respond swiftly to unexpected events, such as accidents, road closures, or sudden surges in demand. Real-time data collection and analysis are crucial for understanding and predicting these changes, enabling transportation managers to implement timely interventions and maintain system efficiency and reliability [50]. Spatial data plays a critical role in transportation systems by providing spatial context to various transportation elements. This data includes information about the location of hubs, safe routes, BSS stations, and traffic patterns. Spatial datasets enable the analysis of spatial relationships and interactions within the transportation network, facilitating better planning, optimization, and decision-making. For instance, understanding the spatial distribution of BSS stations and their usage patterns helps in optimizing station placement and improving service availability. In transportation problems, constructing graphs is a fundamental

approach to model and solve various issues related to routing, scheduling, and network optimization [51], [52], [19]. Graphs consist of vertices (nodes) and edges (links) representing locations and paths or routes between these locations, respectively. This structure allows for the efficient application of algorithms to find optimal paths, minimize costs, and maximize the utilization of resources. In vehicle routing problems (VRP), for instance, graphs help in determining the most efficient routes for a fleet of vehicles, considering constraints like capacity, demand, and time windows [53]. Graphs facilitate the analysis of connectivity and the planning of services to reduce travel time and improve accessibility.

Optimization problems in transportation are known for their significant computational complexity due to the vast number of variables and constraints involved [22]. These problems often fall into the category of NP-hard problems, meaning that the time required to solve them increases superpolynomially (often exponentially) with the size of the input data [23]. Examples include vehicle routing, scheduling, and network design, each requiring sophisticated algorithms to find optimal or near-optimal solutions. The dynamic and stochastic nature of transportation systems further complicates these problems, as they must account for real-time data, changing conditions, and uncertainty. Traditional computational methods can become infeasible as the scale and complexity of the problem grow. NP-hard problems cannot be solved using deterministic approaches due to their inherent computational complexity [24]. Deterministic algorithms require a well-defined procedure to arrive at a solution, usually within polynomial time. This means that as the size of the input grows, the time required to solve the problem increases superpolynomially (often exponentially). That makes it impractical for deterministic algorithms to find exact solutions within a feasible time frame for large instances. Hence, deterministic approaches are often replaced by heuristic [25] and meta-heuristic [26] methods, which can provide good enough solutions more efficiently. Solving NP-hard problems using heuristic and meta-heuristic approaches involves employing strategies designed to find good enough solutions within a reasonable time frame, rather than exact solutions which may be computationally infeasible. Heuristics provide quick, practical methods to generate satisfactory solutions based on problem-specific knowledge. Examples include greedy algorithms

[54] and local search techniques [55]. Meta-heuristics, on the other hand, are higher-level procedures designed to guide underlying heuristics to explore the solution space more effectively. They include techniques such as genetic algorithms [56], simulated annealing [57], and tabu search [58]. These methods are particularly useful for large-scale and complex optimization problems because they balance exploration and exploitation of the search space, often yielding near-optimal solutions efficiently. While they do not guarantee an optimal solution, their ability to provide acceptable solutions in a practical amount of time makes them invaluable in real-world applications.

Given the complexities and computational cost of NP-hard problems, exploring the potential of QC as another meta-heuristic solution is highly promising. As near-term meta-heuristic approaches, quantum algorithms, particularly QA, show potential for efficiently finding approximate solutions to optimization problems [59]. This makes QC an attractive avenue for addressing NP-hard problems in transportation systems, where traditional methods may fall short due to scalability and performance limitations. Numerous researchers have been exploring the potential of QC to solve discrete optimization problems [60], [61], [62], [63]. Few researchers have focused on solving transportation problems using QC. Based on a query of titles, abstracts, and keywords in the Web of Science (WOS) database, only 15 out of 39 papers were found to be related to both transportation and QC at the time of writing this dissertation. Before going to identify further gaps, we will dive into details to highlight different computation models in QC in which each has strength and limits applying in transportation applications.

There are four computation models in quantum computers:

1. Gate model (circuit)
2. Adiabatic
3. Measurement-based
4. Topological

In the following sections, these models are described.

Gate-based Quantum Computing

Superposition and entanglement are features of quantum physics that QC makes use them as the foundation of computation. In classical computers, a bit can only be in one of two states, usually represented as 1 and 0. However, a qubit, the basic building block of QC, can be in a superposition state of 1 and 0. Another difference between a bit and a qubit is that measuring a bit does not affect its state. However, the measurement of a qubit changes its state, except if already being in the computational basis state of 0 and 1. Moreover, a classical system, given the same inputs, will always yield the same results. However, QCs with the same system and inputs will give different results with a certain probability due to their inherent probabilistic nature that comes from the principles of quantum mechanics. [64].

There exist different theoretical approaches to depict the state of a qubit, many times related to the technological implementation of a qubit. One of the well-known ways to depict a qubit state is by utilizing the Bloch sphere diagram. As Figure 1.1 shows, every qubit state can be described in polar coordinates by two angle parameters, θ and ϕ, which give us a point on the sphere; the zero state $|0\rangle$ is located on the north pole of the Bloch sphere and the one state $|1\rangle$ is located on the south pole.

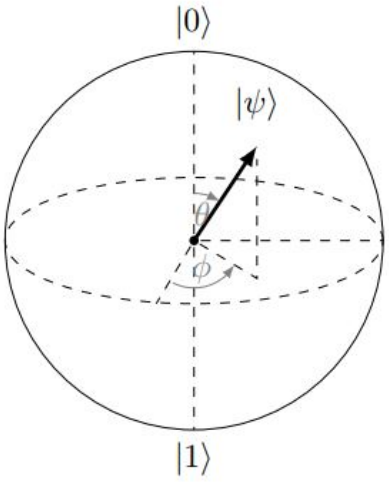


Figure 1.1: The state $|\psi\rangle$ of a qubit on the surface of the Bloch sphere using two polar coordinates θ and ϕ. The poles correspond to the states $|0\rangle$ and $|1\rangle$ by definition.

This allows visualizing 2-dimensional states in the 3-dimensional space. In other words, instead of

considering a qubit as a 2-dimensional complex vector space on the unit sphere, we can equivalently consider the state of the qubit as sitting on the surface of a 3-dimensional unit sphere by introducing the two parameters α and β:

$$|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle \tag{1.1}$$

where $|\Psi\rangle$ represents the state of the qubit and α and β are complex numbers and $|\alpha|^2 + |\beta|^2 = 1$. Based on the definition of complex vectors in polar coordinate systems, the qubit state can be expressed as:

$$|\Psi\rangle = r_0 e^{i\phi_0}|0\rangle + r_1 e^{i\phi_1}|1\rangle, \quad r_0^2 + r_1^2 = 1 \tag{1.2}$$

which leads to:

$$|\Psi\rangle = e^{i\phi_0}[r_0|0\rangle + r_1 e^{i(\phi_1 - \phi_0)}|1\rangle] \tag{1.3}$$

The term $e^{i\phi_0}$ does not change the result after each measurement of the system and can therefore be neglected. Moreover, r_0 and r_1 can be replaced by $\cos\frac{\theta}{2}$ and $\sin\frac{\theta}{2}$. Thus, the equation 1.4 describes the general state of the qubit on the Bloch sphere using two parameters in 3-dimensions:

$$|\Psi\rangle = \cos\frac{\theta}{2}|0\rangle + e^{i\phi}\sin\frac{\theta}{2}|1\rangle \tag{1.4}$$

The variables θ and ϕ parametrize the entire Bloch sphere, and every possible quantum state of a single qubit can be expressed by formula 1.4. A selection of quantum states is illustrated in Figure 1.2.

To change the state of a qubit, quantum gates are applied. There are many kinds of quantum gates as well as many different physical implementations of them. Some of them are basic gates and can transform one or two qubits at a moment. Others are constructed from specific arrangements of several basic quantum gates, called compound gates.

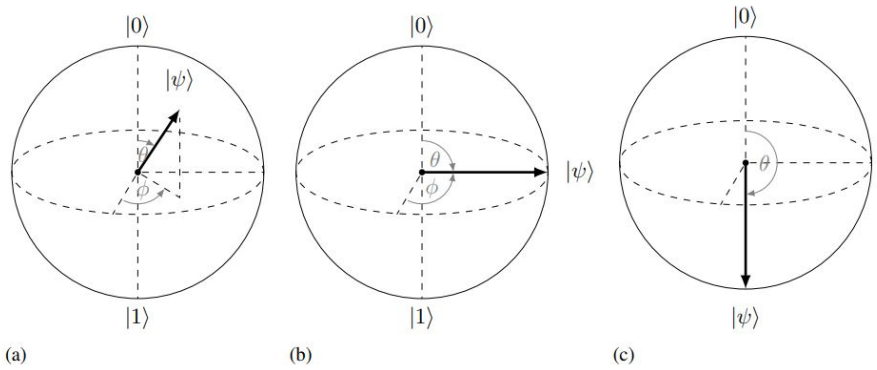


Figure 1.2: Visualization of 3 different qubit states. Figure (a) is described with $\theta = 30°$ and $\phi = 60°$. In Figure (b), the state $|\psi\rangle$ is described by $\phi = \theta = 90°$ and is in an equal superposition of $|0\rangle$ and $|1\rangle$. In Figure (c), the state $|\psi\rangle = |1\rangle$ has no defined value of ϕ, because the first term in equation (4) is zero for $\theta = 180°$, making ϕ part of the global phase variable ϕ_0, which was neglected after equation 1.3.

Any movement on the Bloch sphere is spanned by the 3-dimensional rotation group $SO(3)$, which is isomorphic to the special unitary group $SU(2)$. All elements of this group can be represented as 2-dimensional unitary matrices with unit determinant. In this representation, the qubit basis states are represented as vectors 1.5.

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{1.5}$$

A selection of commonly used single qubit gates is represented in Table 1.1.

A quantum circuit can consist of several gates to construct a sequence of operations on the available multi-qubit system. To perform a quantum computation with multiple qubits, the qubits must interact with each other in some manner. In the gate-based model of QC, this is achieved by introducing gates that act on multiple qubits. When a single qubit gate is represented as a 2-dimensional unitary matrix, then an n qubit gate is represented as a 2^n-dimensional unitary matrix. The simplest and most commonly occurring example is a two-qubit gate called the CNOT gate, given by the matrix 1.6.

Gate	Matrix	Input	Output
Pauli-X	$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$	$\|0\rangle$	$\|1\rangle$
Pauli-Y	$\begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$	$\|0\rangle$	$i\|1\rangle$
Pauli-Z	$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$	$\|0\rangle$	$\|0\rangle$
Hadamard	$\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$	$\|0\rangle$	$\frac{1}{\sqrt{2}}(\|0\rangle + \|1\rangle)$
R_ϕ	$\begin{pmatrix} 1 & 0 \\ 0 & e^{i\phi} \end{pmatrix}$	$\|0\rangle$	$\|0\rangle$
I-gate	$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$	$\|0\rangle$	$\|0\rangle$
S-gate	$\begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$	$\|0\rangle$	$\|0\rangle$
T-gate	$\begin{pmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{pmatrix}$	$\|0\rangle$	$\|0\rangle$

Table 1.1: Single qubit gates with their matrices, their effect on the $|0\rangle$ state, and a general description.

$$CNOT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{1.6}$$

This characteristic is useful for creating entanglement between two qubits, meaning that it can create a two-qubit state that cannot be expressed as a direct product of two single-qubit states. An example is the two-qubit Bell state 1.7.

$$|\Psi\rangle = \frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle \tag{1.7}$$

which can be prepared from the $|00\rangle$ state by applying a Hadamard gate on the first qubit, followed by a CNOT gate controlled by the first qubit and targeted on the second qubit. Both of the qubits are in an equal superposition of $|0\rangle$ and $|1\rangle$, but they are correlated with each other, no matter the measurement outcome, we know that the two qubits will be in the same state.

The physical implementation of quantum gates is specific to each qubit technology. However, there

is a universal trade-off that has to be considered by all of them. Namely, to apply gates to qubits, they must be susceptible to interactions, or else we cannot address them. Unfortunately, this also produces susceptibility to noise. This is especially problematic for entangled states, where each additional degree of entanglement will increase the amount of interference from noise.

Adiabatic Model

Another quantum computation model is called the adiabatic model. In this model, there is an operator called Hamiltonian that determines the total energy of a system and consists of kinetic and potential energy. Let N be the number of qubits available on the quantum hardware system. N qubits are set up in the ground state of H_0. H_0 is gradually transformed into a new Hamiltonian (H_f). In other words, H_0 is the initial Hamiltonian, and H_f is the final Hamiltonian. A gradual transformation can be described with equation 1.8.

$$H(t) = (1-t)H_0 + tH_f \tag{1.8}$$

where t is a given time that gradually increases from 0 to 1 until H_f is reached. The ground state of H_f is $|\Psi_f\rangle$, which is unknown. Finding $|\Psi_f\rangle$ is the problem that adiabatic QC is trying to solve. The quantum adiabatic theorem states that if one starts from the ground state of the initial Hamiltonian and the transformation is done slowly enough, then the final state of the system is the ground state of the final Hamiltonian [65]. One limitation is that the minimal required time needed to perform the transformation scales is as follows 1.9:

$$T = \frac{1}{\min(g(t))^2} \tag{1.9}$$

where $g(t)$ is the difference between the two smallest eigenvalues of $H(t)$ (the difference between the ground energy and the first excited energy). Adiabatic QC is capable of achieving a speedup over classical algorithms [66], where similar to Grover's algorithm [67], finding any given element in an unstructured list could be done in $O(\sqrt{N})$ steps (as opposed to the requisite $O(N)$ for classical

computers).

Quantum Annealing

Quantum Annealing QA belongs to the category of adiabatic QC and is a meta-heuristic algorithm, which can find the minimum of a discrete cost (energy) function. Not all optimization problems can be solved with this approach, but there exist physical implementations that can find the ground state of a specific spin-lattice model called the transverse Ising model [68], where each of the spins on the lattice is represented by a qubit. QA is performed by preparing the ground state of a certain configuration of the spin-lattice model, described by some parameters, and subsequently changing those parameters to adiabatically transform the model such that it has a Hamiltonian that corresponds to the energy function of interest. Classically, finding the minimum of a discrete energy function can be considered as a combinatorial optimization problem.

The Hamiltonian of a transverse Ising model can be defined as 1.10.

$$H = \sum h_i \sigma_i^z - \sum_{ij} J_{ij} \sigma_i^z \sigma_j^z \tag{1.10}$$

where i and j enumerate the different qubits, σ^z are Pauli Z operators, h_i are called qubit biases, and J_{ij} are called coupling strengths. By configuring a network of radio frequency superconducting quantum interference devices (RF-SQUIDs), the values of h and J can be configured and smoothly adjusted during operation [69]. To solve a problem with a discrete energy function $E(s)$, where $s \in \{-1, 1\}$, a corresponding Hamiltonian $H(\sigma)$ with the same functional form needs to be prepared on the quantum computer. This is done by defining a time-dependent Hamiltonian, which adiabatically transforms an easily preparable Hamiltonian $H_X(\sigma)$ into the problem Hamiltonian $H(\sigma)$ 1.11.

$$H(t, \sigma) = A(t) H_X(\sigma) + B(t) H(\sigma) \tag{1.11}$$

where $A(t)$ and $B(t)$ are functions that make up the so-called annealing schedule and $A(t) \gg B(t)$ at the start of annealing $t = t_{start}$ whereas $A(t) \ll B(t)$ at the end of annealing $t = t_{end}$. In the

RF-SQUID based approach, the starting Hamiltonian has a summation form 1.12.

$$H_X(\sigma) = \sum \sigma_i^x \tag{1.12}$$

If the system is prepared in a ground state of $H_X(\sigma)$ at $t = t_{start}$, then it will be in a ground state of $H(\sigma)$ at $t = t_{end}$ if t changes sufficiently slowly following the adiabatic theorem provided before. Therefore, measuring the states of the qubits will yield the ground state of the problem Hamiltonian, which represents the lowest-energy configuration of the original energy function, with high probability.

Quadratic Unconstrained Binary Optimization (QUBO)

Several companies provide cloud-based QC services, making quantum resources accessible to researchers.For instance, D-Wave, IBM Quantum, Google Quantum AI, Microsoft Azure Quantum, Atos, Amazon Braket, etc.

To make use of current QA hardware (e.g., from D-Wave or Atos), one has to formulate the problem at hand as either an Ising problem:

$$E_{Ising} = \sum h_i s_i - \sum_{ij} J_{ij} s_i s_j \tag{1.13}$$

or a Quadratic Unconstrained Binary Optimization (QUBO) problem 1.14.

$$E_{QUBO} = x^T Q x = \sum_{ij} Q_{ij} x_i x_j \tag{1.14}$$

where x is a binary vector and Q is a matrix. The energy function of a QUBO resembles the Ising formulation except that the binary variables are either 0 and 1 in comparison to the spin values of the Ising model which are -1 and 1. The two models are easily convertible to each other, and the QUBO form is often simpler to handle mathematically. Although this format of problem input may seem restrictive, many important NP complexity class problems can be expressed in this manner [70].

Measurement-based Model

Measurement-Based Quantum Computing (MBQC) is an approach to QC, where the computation is implemented by performing a sequence of measurements on a quantum computer. This is possible because measurements change the state of the quantum computer and can thus be used to emulate the action of quantum gates, enabling the processing of quantum information. Universal quantum computation is possible within this framework, first shown by Raussendorf and Briegel in 2001, where they devised a so-called one-way quantum computer, which could implement the universal gate set using single-qubit measurements [71].

Although measurements can be used to create entanglement [72], it is more often the case that measurements destroy entanglement instead. For example, when performing any single-qubit measurement, the state of the measured qubit must necessarily collapse to a pure state, thus removing any entanglement it had in relation to other qubits. Because of this feature, MBQC most often relies on preparing a highly entangled state, called a resource state, on the quantum computer beforehand, where entanglement can be thought of as a resource, which is successively used up by the measurements needed to perform the computation. The specific design of a resource state has to be carefully considered because it was discovered by Gross et al. that if the resource state is chosen randomly (from the Haar measure), then MBQC offers no speedup for computation [73]. Unfortunately, finding a suitable resource state for a given computation is in general more difficult than the computation itself [74]. Moreover, physical hardware and design constraints limit the possible resource states that can be efficiently prepared, as it is very challenging to develop the capability to prepare arbitrarily entangled states. Perhaps one of the simplest examples of a resource state is a cluster state, which consists of a regular lattice of entangled qubits, such as the square lattice used in the original one-way quantum computer [71].

Another crucial aspect to keep in mind with the MBQC scheme is that the results of measurements are inherently random. As a consequence, the unitary transformations induced by those measurements are concatenated with a random Pauli operator called a byproduct operator [75]. To avoid an indeterministic calculation, this randomness can be mitigated by selecting the basis of each mea-

surement based on the results of the measurements preceding it. This extra classical processing of information can be harmful insofar as it may cause a time-delay, which can lead to increased decoherence in the quantum computer.

MBQC is particularly appealing for QC with photons, where creating highly entangled cluster states is possible and performing single-qubit measurements is immensely simpler than performing multi-qubit operations required for the gate-based model [76]. Additionally, Broadbent, Fitzsimons, and Kashefi discovered that MBQC can be used for blind quantum computation, where a client can outsource their computations to a server in a way where the server cannot decipher what computation is being performed [77], which is especially useful for future applications that have a high demand for security and privacy [78]. For a more in-depth review of MBQC, see [79], [80], [81].

Topological Model

A common feature of all the previously described computational models is that information is encoded into qubits locally. Through decoherence, information is lost when qubits interact with their local environment. The time that this process takes is called the decoherence time and it limits the accuracy and length of calculations that can be performed on quantum computers.

The topological model of quantum computation proposes to circumvent decoherence by encoding information non-locally, into global characteristics known as topological invariants. A classical example of this, shown in Figure 1.3, is encoding information by braiding ropes into knots that differ by their knot invariant. The two knots shown in the figure are topologically inequivalent, which means that one cannot be deformed into the other without cutting the rope. This is a global property since no particular part of either rope carries this information.

Certain quantum states could also be braided along space and time dimensions, visualized in Figure 1.4. Superpositions of the occupation number of those states enable the storage of quantum information. Many interactions with the environment occur over small length scales and are exponentially suppressed as the braiding is performed over larger distances. Only specific kinds of

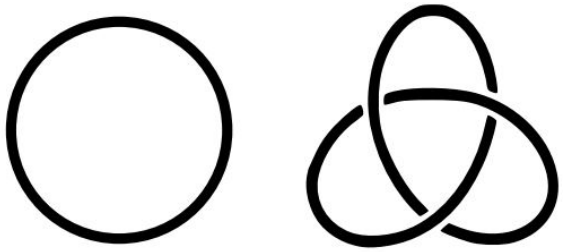


Figure 1.3: The trivial knot (left) and the left-handed trefoil knot (right). These knots are topologically inequivalent. If this difference is used as a basis for encoding information, then the information is protected from all rope deformations besides cutting.

processes may still destroy the information. One such example is quasiparticle poisoning, where the environment introduces additional particles into the system. These particles might braid and fuse with the existing particles in the system, but the rate of this process is still estimated to allow for decoherence times on the order of seconds [82].

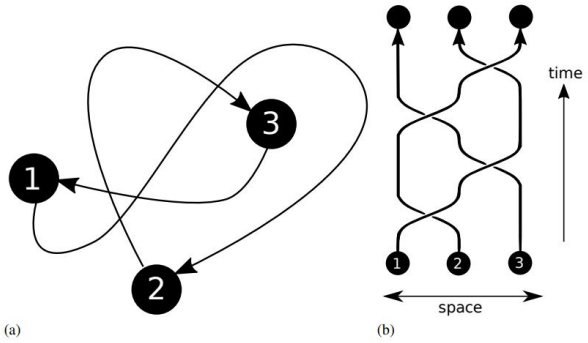


Figure 1.4: Real-space trajectories of three particles (a), and the braiding of their worldlines (b). If the particles are non-Abelian anyons, then the final many-body quantum state is different than the initial one and information can be encoded as states that are related by different, topologically inequivalent braids. Deformation of the trajectories by noise or inaccurate control does not alter the stored information, as long as the particles do not come into contact. This robustness provides resilience to errors.

The quantum states that possess the necessary properties describe particles called non-Abelian anyons [83]. Elementary particles of this type have not been discovered in nature, but it could be possible to create quasiparticle excitations within low-dimensional solid-state systems, which fit into this specific class of particles [84]. Recent experimental research shows evidence for the existence of anyons [85], [86], but so far there has been no conclusive observation of any non-Abelian anyons. Consequently, no topological computers or qubits have been built to date.

Current topological models involve fabricating semiconductor-superconductor hybrid nanostructures to engineer topological superconductors, which are theoretically predicted to host non-Abelian quasiparticles at their boundaries [87], [88]. This approach has the potential to offer a path towards a scalable, fault-tolerant topological quantum computer. Similarly to superconducting qubits, extremely low temperatures are required for operation [89], [90].

QA has shown promising results in solving complex optimization problems, and its development is rapidly progressing, making it a viable option for real-world applications. Moreover, QA can be effectively utilized in hybrid models, combining classical and QC approaches. This hybrid approach allows for leveraging the strengths of both classical and quantum systems, enabling the efficient solving of problems that are intractable for classical computers alone. The integration of QA with classical methods provides a powerful toolset for tackling a wide range of optimization problems. Another important factor driving this research is the increasing accessibility to a high number of qubits. The number of qubits available in QA has been rising, allowing for more complex and larger-scale problems to be addressed. This increase in qubit numbers enhances the computational capabilities of this model, making them more robust and effective for various applications. Furthermore, the availability of cloud-based QC computers makes them invaluable computational resources to explore. Researchers can now access QA computers remotely, without the need for substantial investments in physical hardware.

The complete list of published QC papers in transportation can be accessed here[1]. They have proposed solutions for a small set of datasets, moreover, these studies have overlooked the critical aspect of spatial features within transportation networks. Spatial data, which encompasses the spatial relationships and geographical context of transportation systems, is crucial for accurately modeling and solving transportation problems. Additionally, they have not considered the inherent uncertainty present in real-world datasets. This oversight can significantly impact the robustness and applicability of proposed solutions. Real-world transportation data is often subject to variability and unpredictability due to the abovementioned factors. Traditional deterministic models fail to

[1] https://www.webofscience.com/wos/woscc/summary/0dc64c37-807d-4ba5-a5ce-ee704e696e13-eaf7c791/relevance/1

account for this uncertainty, leading to solutions that may perform well in theoretical settings but fall short in practical applications. Addressing this gap by incorporating real-world spatial features into models is crucial for developing more reliable and resilient transportation systems. Ignoring these features can lead to less effective solutions, highlighting a significant gap in current research that needs to be addressed.

For instance, a significant challenge in BSS is the Bike Routing Problem (BRP), a well-known NP-hard combinatorial optimization problem. BRP focuses on the efficient rebalancing of bikes across a network, ensuring bikes and docking stations are adequately distributed to meet demand. This problem is critical as it directly impacts the availability and convenience of the service, influencing user satisfaction and system efficiency. Further complicating the management of BSS is the need for a comprehensive understanding of the network's overall state to identify and group similar stations. Clustering methods such as $K-means$ [91], $hierarchical clustering$ [92], and *DBSCAN* [93] serve as effective tools in this regard. These techniques help planners gain insights into usage patterns, allowing for the strategic placement of resources in high-demand areas. However, the implementation of these algorithms presents its own set of challenges. They are computationally intensive, particularly when applied to large, dynamic datasets, and the $K-means$ algorithm, for instance, is an NP-hard problem. The difficulty mainly lies in determining the optimal partition of data into k clusters to minimize the sum of squared distances between data points and their respective cluster centroids, a task that becomes increasingly complex with the scale of data. Another critical aspect of enhancing BSS management is the development of predictive models to forecast ridership, which necessitates the use of rich spatial features. While these features can significantly improve the accuracy of predictions, they also introduce complexities in the Feature Selection (FS) process. The FS process is crucial as it determines which features should be included in the model to optimize performance without overfitting or underfitting. However, solving FS through brute force approaches or evaluating all possible combinations of features is computationally prohibitive, especially with high-dimensional data. Checking multiple combinations leads to a combinatorial explosion, making it impractical to find the optimal set without advanced heuristic or algorithmic solutions. Many

researchers have explored FS using QC. The concept of using QA for FS was first introduced in 2017 [94]. They formulated the FS problem using QUBO and applied it to credit scoring and classification. Their method aimed to identify independent and influential features, resulting in a smaller feature subset without compromising accuracy compared to Recursive Feature Elimination (RFE). This demonstrated QA's potential in high-dimensional data mining tasks. Nembrini et al. [95] developed a hybrid FS algorithm for recommender systems using QA, solving the optimization problem with D-Wave's quantum computer. Their results showed effectiveness in identifying critical features, highlighting QC's growing practical applications. Hellstern et al. [96] applied FS in QC for supervised learning, formulating QUBO and solving it with both classical and quantum methods, including QA and Quantum Approximate Optimization Algorithm (QAOA). They compared QUBO-based FS with standard methods like RFE and LASSO. Mücke et al. [97] proposed a novel FS algorithm based on QUBO, optimizing feature importance and redundancy. They solved this QUBO problem using classical computers, Variational Quantum Eigensolver (VQE), and QA. Their experiments on benchmark datasets showed competitive performance with standard FS methods. Vlasic et al. [98] introduced a Quantum Feature Selection (QFS) algorithm using QA and information-theoretic metrics, demonstrating superior model stability and predictive performance compared to classical FS methods. Sabbar et al. [99] proposed a Quantum-Inspired Genetic Algorithm (QIGA) to enhance Automatic Modulation Classification (AMC), leveraging QC principles to improve FS and classification accuracy. This approach showed superior performance by optimizing the search process and extracting enhanced cumulants from modulation signals, demonstrating the potential of quantum-inspired optimization in improving AMC accuracy and efficiency. Additional related works can be found in [100, 101, 102, 103, 104, 105]. While previous works have made significant strides in applying QC and QA to FS, they have often focused on specific applications or problem formulations. However, these approaches have not adequately addressed the need for a versatile, model-agnostic FS algorithm that can be broadly applied across different regression models. In this research, we filled that gap by introducing a novel model-agnostic algorithm for FS in regression models, which, in most cases, outperforms current state-of-the-art

approaches in QA.

1.3 Research Questions

- How can QC be employed to develop efficient algorithms for minimizing optimization problems in transportation?
- How to formulate a QC model for dynamic rebalancing optimization problems such as BSS with spatial data?
- How to understand effectively BSS network as whole and identify the critical parameters to enhance system management and optimize demands?
- How to predict BSS ridership considering spatial features?
- How can QC be used to address the high dimensionality of features in predictive models in BSS ridership demand?

1.4 Objectives

Considering the needs and gaps identified, the main objective of this research is to propose a spatial QC framework to enhance various aspects of transportation systems. The following specific objectives are designed to build upon each other to comprehensively explore and apply QC techniques to various transportation challenges, particularly within BSS.

1.4.1 Explore the potential of QC in transportation applications

This objective investigates the potential of QC in transportation applications through a case study of urban pandemic management. An optimization problem aimed at decreasing the spread of the virus in a road network is formulated into a QUBO problem. A COVID-19 dataset is utilized as a proof of concept to test and validate the proposed method.

1.4.2 Develop a QC-based approach for dynamic rebalancing bike sharing system dynamic: a critical part of multi-modal transportation systems

By analyzing the results of the previous objective and after checking the potential of QC in transportation problems, a new objective is defined to develop an optimization model for dynamically rebalancing BSS a critical part of multi modal tranpostion systems using QC. We first formulate the problem as a QUBO model and solve it using different solvers on the D-Wave QA computer. As proof of concept, Toronto BSS dataset is used.

1.4.3 Enhance BSS management by clustering stations using QC

This objective applies QML techniques to cluster BSS stations in Toronto. It involves developing a hybrid model to address the clustering problem as a Constraint Satisfaction Problem (CSP) using QA. The performance of various dissimilarity functions and methods for determining the optimal number of clusters is assessed. The impact of the proposed clustering method on BSS is evaluated, focusing on improving system efficiency to response to demand and user satisfaction.

1.4.4 Predict BSS ridership to enhance system management

This objective focuses on exploring various machine-learning models to predict BSS ridership. By incorporating temporal, environmental, and spatial factors, the goal is to enhance prediction accuracy. Different models, including Linear Regression (LR), Poisson regression, Random Forest (RF), Extreme Gradient Boosting (XGBoost), and XGBoost with exposure, are compared to identify the most effective method for accurately predicting ridership and improving system management.

1.4.5 Improve the accuracy of the predictive models by proposing a new method for feature selection using QC

This objective aims to develop a model-agnostic FS method leveraging QC. The FS method is implemented on a QC platform and its effectiveness is evaluated. The proposed method is compared with traditional FS techniques to demonstrate its advantages and potential applications, highlighting improvements in the accuracy of predictive models. Toronto BSS dataset is used as case study.

1.5 Proposed Methodology

In this section, the methodology used to conduct the research, detailing our approach to addressing the research questions and achieving the objectives is described. This exposition of the methodology aims to outline the development of a spatial QC framework designed to enhance various aspects of transportation systems. To meet the research objectives discussed in the previous section (Section 1.4), the methodology is divided into five parts.

1.5.1 Explore the potential of QC in transportation applications

To explore the potential of QC in transportation a novel approach to minimize COVID-19 exposure during city journeys by transforming the problem into a QUBO problem and solving it using QA computers is proposed. The methodology consists of several steps, including the construction of a road network, trajectory data reconstruction, and the formulation of the QUBO problem. The road network is constructed using a geo-data frame generated from GeoJSON data. The NetworkX library is then used to create a graph from the geo-data frame, with intersections as nodes and road connections as edges. Trajectory data for road users is reconstructed from origin and destination information. A geodatabase is created from a file containing this data and imported into ArcGIS to generate a feature class for trajectory data. This feature class is then converted to GeoJSON format and subsequently to a geo-data-frame, enabling the mapping of trajectory data onto the road network. To map the trajectory data onto the road network, the nearest node in the road network

to each trajectory point is identified. This helps in accurately mapping the trajectory points to the road network nodes. Moreover, the COVID-19 data is mapped as weights to the nodes and edges. Once the graph is ready, the Q matrix in QUBO is constructed. The Q matrix, an upper triangular matrix, is defined based on the linear and quadratic coefficients of the QUBO problem. Linear coefficients are added to the diagonal entries, and quadratic coefficients to the off-diagonal entries, transforming the real-world problem into a format solvable by quantum computers.

1.5.2 Develop a QC-based approach for dynamic rebalancing of BSS

To design and develop a QC-based approach, a spatio-temporal method is formulated as QUBO to solve a dynamic rebalancing issue in BSS. The methodology to transfer the rebalancing problem into QUBO involves different steps. The road network construction and trajectory data reconstruction are formulated from the previous section. The Point-in-Polygon (PIP) algorithm is used to store the trajectory data, ensuring that the end station is considered as the destination if the buffer for rebalancing for one bike is empty. Once the QUBO is defined, the Ising model is then mapped to the architecture of the quantum computer, such as the chimera or pegasus graph in D-Wave quantum computers. The energy function, or Hamiltonian, is used to describe the system's energy state. Four different solvers are investigated of which three are purely quantum solvers (Advantage system 4.1, 5.2, and 6.1), and one is a hybrid solver (Hybrid binary quadratic model version 2). These solvers are evaluated based on various parameters, including energy, sampling time, readout time, and QPU access time. The model's effectiveness is evaluated using the occupancy index, which measures the distribution of bikes across the network. A comparison map is created to visualize the distribution of bikes for actual trips and QC-suggested routes.

1.5.3 Develop a method to clustering BSS stations using QC to Enhance BSS management

Rebalancing bikes in BSS is necessary but insufficient on its own. To further enhance the efficiency and management of BSS, it is crucial to obtain a comprehensive overview of the station states and identify patterns of similar behavior among them. The goal is to optimize the clustering of stations in BSS by considering real-time characteristics and spatial features. Spatial data of BSS stations are collected and processed to form feature vectors representing each station's characteristics. A hybrid model is developed to solve the clustering problem using QA. The clustering problem is formulated as a Constraint Satisfaction Problem (CSP). Different methods for determining the optimal number of clusters are assessed. Various dissimilarity functions are tested to determine the optimal way to measure differences between clusters.

1.5.4 Develop BSS ridership predictive models to enhance demand and user satisfaction

To advance transportation system management, predictive models are explored to identify the most effective methods for predicting ridership in BSS. For FS, linear correlation-based method and Mutual Information (MI) are used. The following predictive models are tested; Linear Regression, Linear Poisson Regression, RF, XGBoost Regression, XGBoost Poisson, XGBoost Poisson with Exposure.

1.5.5 Develop QC-based feature selection method to improve the accuracy of the predictive models e.g. BSS ridership prediction Models

An essential aspect of the methodology involves the proposed FS model. Given the numerous factors influencing the dynamics of transportation systems, incorporating all these factors into predictive models is inefficient. Therefore, a model-agnostic FS method leveraging QC is introduced to address this challenge. The objective is to select features that maximize correlation with the

target variable while minimizing redundancy among features. To construct the Q matrix in QUBO, the linear coefficient in the cost function encapsulates both linear and non-linear associations between features and the target variable. This includes a weighting factor designed to enhance the significance of each feature. The quadratic coefficient is defined to minimize redundancy and noise among features. The correlation functions for linear and quadratic terms are specified. The quantum sampling is designed to balance classical and quantum computations, iterating to select the optimal number of features. The proposed method is evaluated against traditional MI and other state-of-the-art FS techniques on three datasets: Wine Quality, Insurance, and BSS usage. Performance metrics include R-squared (R^2) and Root Mean Squared Error (RMSE) for the selected features using the three target models.

1.6 Research Contribution

To address the identified gaps and problem statements and to achieve the targeted objectives, this dissertation defines four contributions as follows:

1. Formulated an Optimization Problem for Virus Spread Reduction: Developed an innovative approach to reduce virus spread in urban road networks by converting the problem into a QUBO problem. Utilized a COVID-19 dataset as a proof of concept to validate the effectiveness of QC in addressing complex transportation applications.

2. Developed a Spatial Optimization Model for BSS Rebalancing: Created a dynamic rebalancing model for BSS using QC, formulated as a QUBO problem. Demonstrated its effectiveness with real-world data from Toronto's BSS and tested it using different solvers on the D-Wave QA computer. This contribution highlights the novelty of applying QC to optimize urban mobility and resource allocation.

3. Applied Quantum Machine Learning for BSS Station Clustering: Employed Quantum Machine Learning (QML) techniques to cluster BSS stations based on spatial features. Addressed the clustering problem as a CSP using QA, and evaluated the impact on system

efficiency in Toronto. This work introduces a novel application of QML for enhancing the operational efficiency of BSS through advanced clustering methodologies.

4. Proposed a Novel Feature Selection Method Using QC: Improved the accuracy of predictive models by developing a new method for FS using QC. The model-agnostic FS method, implemented on a QC platform, demonstrated its effectiveness in enhancing predictive model accuracy compared to traditional FS techniques. This contribution showcases the innovative use of QC to advance machine learning methodologies and improve predictive capabilities in transportation systems.

1.7 Dissertation Outline

This is a paper-based dissertation. Thus, each chapter contains its introduction, literature review, specific objectives, methods, results, and discussion.

Chapter 2 explores the potential of QC in solving combinatorial optimization problems, specifically focusing on minimizing exposure to COVID-19 during city journeys. The chapter includes a preface to set the stage for the research problem and its significance, an abstract summarizing the study's aim, a summary of the research providing an overview of the problem definition, contributions, and methodology used, a background review of relevant literature and theoretical foundations, a detailed description of the proposed methodology and its implementation, and a discussion of the results and their implications. The chapter concludes with a summary of its contributions and future research directions.

Similarly, chapters 3, 4, 5, 6 each contain their preface, authors contributions and licences, abstract, introduction, literature review, specific objectives, methods, results, discussion, and conclusion.

Chapter 3 unpacks one of main contributor of multi-modal transportation system: the dynamic rebalancing of BSS using QC. It begins with a preface that introduces the importance of dynamic rebalancing in BSS, followed by an abstract summarizing the chapter's aim and findings.

Chapter 4 describes the potential of QC in ML, specifically for clustering real-world data within

BSS. The preface introduces the significance of station clustering for BSS.

Chapter 5 focuses on the ML approaches for sustainable transportation management, particularly predicting BSS ridership with spatial features.

Chapter 6 introduces a novel approach using QC to enhance FS processes in ML. The proposed method leverages QA to navigate the complex search space of FS more effectively than traditional methods. The development of this QC-based FS method involved experimentation and validation on various datasets.

Chapter 7 includes the summary of the research, final remarks, and contributions. It discusses the limitations and offers recommendations for future research. The dissertation concludes with a bibliography and appendices.

Chapter 2

Exploring Quantum Computing Potentials in Solving a Combinatorial Optimization Problem; Minimizing Exposure to Covid-19 During a City Journey

2.1 Preface

This section describes the first sub-objective, which is an approach to formulate the risk of exposure to a virus (case study; COVID-19) during city journeys through an optimized Vehicle Routing Problem (VRP). One of the most critical aspects of managing the pandemic is minimizing exposure to the virus in various public and private spaces. As an airborne virus, COVID-19 spreads directly through close contact and indirectly through contaminated surfaces, making its transmission a complex spatio-temporal event. This complexity is further compounded by factors such as traffic congestion, which can increase the spread due to increased proximity and environmental conditions like temperature and humidity. By transforming the problem into a QUBO problem and solving it using QA computers, this research aims to minimize the risk of exposure for road users

traveling between various origins and destinations. Utilizing Microsoft Taxi data from Beijing, the proposed model has been tested on different solvers, including classical and hybrid quantum solvers. The work presented here is not just a technical exercise but a response to a global crisis, offering innovative solutions to public health challenges. It builds on the promising advancements in QC, leveraging its capabilities to address real-world problems. The main contribution of this chapter is to formulate an optimization model to reduce COVID-19 exposure risk by solving it as a QUBO problem on a QA computer. Indeed, the objective of the COVID-19 prevention optimization problem is to minimize the risk of exposure for a given set of road users between origins and destinations. This has been done by minimizing the proposed cost function with respect to origin and destination, called segment. The cost function on an individual segment is defined based on a quadratic function with two parameters. The first parameter is the number of COVID-19 cases for two nodes of a segment, and the second is the number of road users in each segment at a specific time interval. The main challenge in this study is to investigate transforming a real-world discrete geospatial routing problem into a quantum computer. To illustrate the proof of the concept, Microsoft Taxi data from the city of Beijing have been used to simulate road user movement.

The content of this chapter is published in the International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, 2022 as follows:

A. Nourbakhsh, M. Jadidi, and K. Shahriari. "Exploring Quantum Computing Potentials in Solving a Combinatorial Optimization Problem to Minimize Exposure to COVID-19 During a City Journey." *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences 43 (2022): 419-425* [1].

2.1.1 Copyright and Licensing

2.1.2 Contributions

The roles of the authors in this chapter are outlined as follows: Amirhossein Nourbakhshrezaei was responsible for the literature review, data collection and organization, methodological development, implementation for analysis and modeling, result validation and visualization, and the initial manuscript writing. Mojgan Jadidi oversaw the research, provided the funding, and contributed to the methodology development, as well as the writing and editing of the manuscript. Kyarash Shahriari supervised the research and assisted in the manuscript editing.

2.2 Abstract

COVID-19 is an airborne virus that can be spread directly or indirectly from one person to another. Spreading the virus strongly depends on the location and time and hence, a Spatio-temporal event. Moreover, traffic congestion will increase the spread of the virus not only because of the vicinity but also because of increased temperature and humidity in these spaces for a short or long time. This paper introduces a vehicle routing optimization model to reduce COVID-19 exposure risk during a city journey by solving it as a QUBO problem on a QA computer. Indeed, the objective of the COVID-19 prevention optimization problem is to minimize the risk of exposure for a given set of road users between origins and destinations. Microsoft Taxi data from the city of Beijing have been used to simulate road users' movement. The problem has been run onto three different solvers. One of the solvers is executed on classical computers, and two other solvers are executed on hybrid quantum solvers. Hybrid solvers return the solution within less than 0.03 seconds on quantum processing unit time. However, the results will be returned at least 5 seconds after the execution in the classical solver. It is worth mentioning that, as there is no direct access to the quantum computers, it is hard to compare the results on the same scale as the queries will go on a queue in D-wave quantum computers. Applying the proposed model on the trajectory data shows a better distribution of the vehicles on the road network.

2.3 Summary of the Research

2.3.1 Introduction

According to the World Health Organization (WHO), approximately 198.9 million people were infected by COVID-19 in 2021 worldwide, among which 3.5 million lost their lives [106]. This infection had adverse mental health effects on society and largely impacted the economy. Controlling the spread of the virus by reducing the exposure to it, is a way to diminish the noxious health and wealth impacts of it. Therefore, this is one of the major tasks that should be taken care of by

both individuals in their everyday decision-making; and longer-term, larger-scale guidelines, and regulations by public and authorities decision-makers. To our knowledge, COVID-19 is an airborne virus which can be spread directly or indirectly from one person to another. The direct transmission is by breathing the exhale air of infected nearby persons with distant of less than 1.5 meter apart [107]. In the indirect way, an infected persons contaminates objects by touching and the virus last on the surfaces for a while depending the type of material and environmental conditions. Therefore, spreading the virus strongly depends on the location and time and hence, a spatio-temporal event. Moreover, traffic congestion will increase the spread of the virus not only because of the vicinity [108] but also because of increased temperature and humidity in these spaces for a short or long time [109, 110, 111].

Solving a routing problem involves both individual and public decision makers; one decides the time and the trajectory for their journey in the city from an origin to a destination. In the public level, the transportation availability, guidelines and regulations impacts directly the decisions in individual level. In a pandemic, additional cost functions and constraints should be added to penalize and restrain the exposure to hot zones to attenuate the spread of the disease. The solutions to this NP-hard problem based on classical computers are well-known and already exist [112, 113]. However, they are not fast enough while complexity increases to provide real-time or close to real-time response to neither individual nor public level problems. Based on the current technological developments in Quantum Computers and the very promising results in solving complex problems, they can be the answer to solving large scale routing problems in real-time [114, 115].

2.3.2 Problem Definition

The core of this study is the Vehicle Routing Problem (VRP), which is one of the well-known NP-hard combinatorial optimization problems in the classical computers. Many heuristic and meta-heuristic algorithms have been proposed to solve the VRP problem in classical computers. However, these algorithms still have trouble solving such problems in real-world applications. On the other hand, quantum computation and, more specifically, adiabatic quantum computation are

promising near-term meta-heuristic approaches to solve combinatorial optimization problems.

To be able to solve the problem on current quantum computer hardware, the problem should be transformed to a QUBO. Finding the minimum value of QUBO is a known NP-hard problem as the Ising model [116]. Therefore, this research proposes a Spatio-temporal model that can be converted to the QUBO, and as a result, it can be run on a quantum computer.

2.3.3 Contribution

The main contribution of this research is to introduce an optimization model to reduce COVID-19 exposure risk by solving it as a QUBO problem on a QA computer. Indeed, the objective of the COVID-19 prevention optimization problem is to minimize the risk of exposure for a given set of road users between origins and destinations. Cost function on an individual segment is defined based on a quadratic function with two parameters. The first parameter is the number of COVID-19 cases for two nodes of a segment, and the second is the number of road users on each segment at a specific time interval. The main challenge in this study is to investigate transforming a real-world discrete geospatial routing problem into a quantum computer. To illustrate the proof of the concept, Microsoft Taxi data from city of Beijing have been used to simulate road users movement.

2.4 Background

There are four main computational models in QC; 1) Gate-based/circuit-based models[117], 2) Adiabatic models[118], 3) Measurement based models[80], and 4) Topological models[119]. In this research, QA is being used. Since QA is a sub-category of adiabatic quantum models, this section first describes adiabatic models and then provides details of the QA. Generally, adiabatic quantum models rely on the adiabatic theorem in quantum mechanics[65]. The main characteristic of the adiabatic models that makes them appropriate for solving optimization models is the energy of the system. The energy of the system consists of kinetic and potential energy. In order to define the energy of the adiabatic models in a system, the Hamiltonian method[120] has been used (Eq

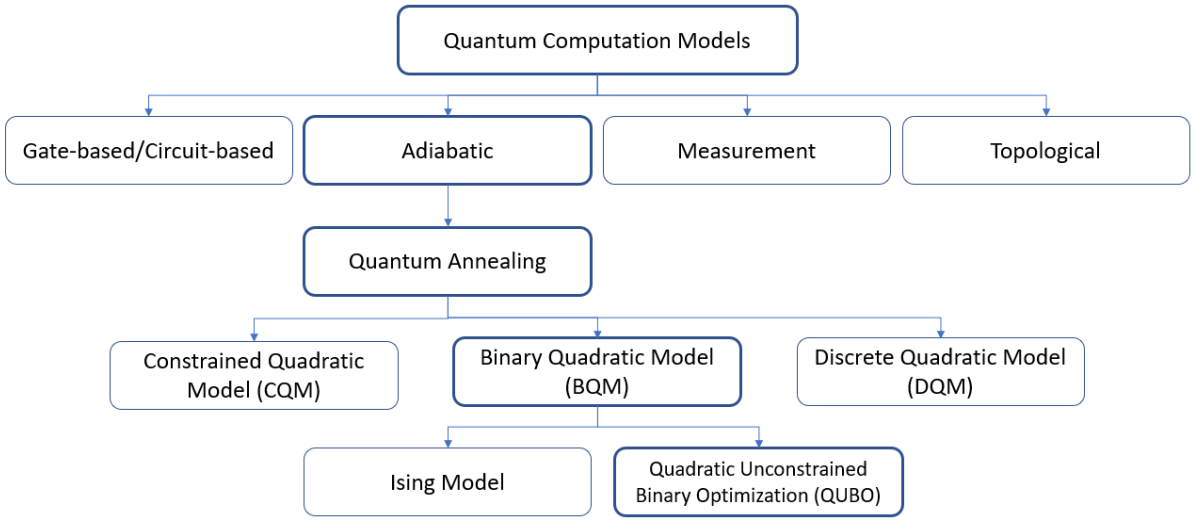


Figure 2.1: The overview of the scope of the paper. The QUBO is used.

2.1). Assuming that the quantum computer system has N qubits and are set up to the ground state of H_0. H_0 should be transferred to a new Hamiltonian that is called final Hamiltonian (H_f).

$$H(t) = (1-t)H_0 + tH_f \tag{2.1}$$

where t is time and it is gradually increasing from 0 to 1. H_0 is an initial Hamiltonian, and H_f is the final Hamiltonian.

QA is a meta-heuristic approach that uses the nature of adiabatic quantum models to find the low-energy of the system. QA does not apply to all problems. The functional problems can be divided into two parts. In some applications, the main goal is to find minimum energy; on the other hand, in other applications, the purpose is to find all low-energy samples. The first problems are called optimization problems, and the second one is called probabilistic sampling problems.

The concepts of energy in physics can bring the optimization models into an energy optimization problem by using the fact that everything tries to reach a minimum energy level. For example, Each object around us slides downhill (local optimal). QA uses a similar concept in quantum physics to determine the low-energy of a problem and finds the optimal or near-optimal combination of

all possibilities. However, in probabilistic sampling problems, the system tries to find samples from many low-energy states and creates a landscape of the energies. This approach is suitable for Machine Learning (ML) problems since it helps build a probabilistic model of the real world. In order to better understand the scope of this research, an overview of the above-mentioned details is shown on the Fig2.1.

In general, to formulate an optimization problem to be able to run on a QA computer, based on the nature of the problem one of the following classes should be used:

- Binary Quadratic Models (BQM)[121]
- Constrained Quadratic Models (CQM)[122]
- Discrete Quadratic Models (DQM) [123]

In this paper, BQM is used. Two main formulas are mostly used to define the objective function of the problem: The Ising model and QUBO. Conversion between these two formulas is trivial. The following section describes the proposed method by using QUBO method.

2.5 Proposed Methodology

This research proposes a Spatio-temporal model that helps decreasing exposure to the Covid-19 virus by reducing traffic congestion and number of Covid-19 cases in each area. The case study of the proposed model is Beijing city. Data layers that are being used in this research contain COVID-19 cases and vehicle's trajectory. The final results of the proposed model is compared to the current state of the city. The evaluation will be done by comparing two different states, with and without using the proposed model.

Quantum computers can be classified based on different technologies and different computational models. In this research, a D-Wave quantum computer is being used. D-Wave quantum computers are built based on a QA model, a subcategory of the adiabatic models. Adiabatic quantum computation models and specifically QA hardware are promising near-term meta-heuristic approaches to

solve combinatorial optimization problems. QA is a meta-heuristic algorithm that finds the minimum of a discrete cost function. The problem should be formulated as either an Ising problem or a QUBO problem to take advantage of current QA computers such as D-Wave. These two models are convertible to each other, and the QUBO is often simpler to handle mathematically (Eq 2.2).

$$Cost\,function = x^T.Q.x \tag{2.2}$$

where x = an N x 1 vector of binary variables

Q = an N x N upper-triangular matrix of real weights

Eq 2.2 can be expressed more precisely as Eq 2.3.

$$CostFunction = \sum_i Q_{i,i}x_i + \sum_{i<j} Q_{i,j}x_ix_j \tag{2.3}$$

where $Q_{i,i}$ = are the linear coefficients

$Q_{i,j}$ = are the quadratic coefficients

For each car, there are multiple alternative routes from which only few of them are considered as shortest paths. We selected three shortest alternative routes in order to identify the safest route. Each route contains multiple segments (edges) of the graph. To formulate the cost function for COVID-19 prevention, we define a summation of the number of COVID-19 cases assigned to the nodes of each edge (segment) and the shared segments between assigned routes for each car. By doing this, both historical data (COVID-19 statistics) and dynamic data (current traffic flow) play a significant role in distributing cars within the road network. We define a new function to minimize the risk of exposure (ROE) to the virus as follows (Eq 2.4).

$$ROE = \sum_{s\in S}(\sum_v \sum_r \sum_{s\in S_r} q_{vr}.(n_1+n_2))^2 \tag{2.4}$$

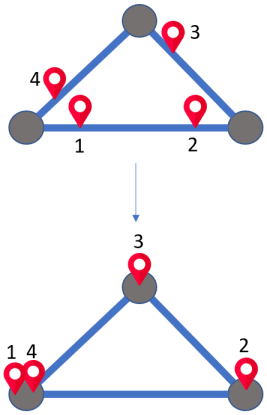


Figure 2.2: Mapping trajectory data to road network. Red icons represent trajectory data and grey nodes are the intersections in the road network.

where s = segment

S = all the segments in road network

v = number of vehicles

r = number of alternative routes

$n1, n2$ = are normalized value of COVID-19 cases of start and end of a segment

q_{vr} = binary variable

The fact that QUBO problems are unconstrained does not mean that it is impossible to add constraints into the model; indeed, there are no constraints on the variables other than those defined in Q. The constraint of the model is defined based on the fact that each car cannot take more than one route at the time. To be more specific, given two cars and two alternative routes for each, all the possible states for qubits are shown in Table 1. In Table 1, the states are acceptable that the row's total values are equal one. In this regard, the final cost function have been revised as Eq 2.5.

$$Cost function = ROE + \sum_{v}(\sum_{r} q_{vr} - 1)^2 \tag{2.5}$$

	q1	q2	Status
Car 1	0	0	No
Car 1	0	1	Yes
Car 1	1	0	Yes
Car 1	1	1	No
Car 2	0	0	No
Car 2	0	1	Yes
Car 2	1	0	Yes
Car 2	1	1	No

Table 2.1: Possible states for two cars and two routes. Only records are acceptable with "Yes" value for Status column that means each car can only take one route at each time.

2.5.1 Pseudo Code

The pseudo code of the proposed method is provided in algorithm 1: Q construction Pseudo Code Table.

Algorithm 1 Q construction Pseudo Code

$G \leftarrow NetworkXGraph$
$Q \leftarrow [emptyMatrix]$
$dataset \leftarrow csvfile$
for i in cars:
- for j in routes:
- for k in segment:
- $node1 \leftarrow k.getNode(0)$
- $node2 \leftarrow k.getNode(1)$
- $riskCovid19 \leftarrow mapCovid19ToEdges(node1, node2)$
- weight ← weight + riskCovid19
- G.addEdge(k,weight)
- weight(j).update
- Q(i,j).update
- $if(Q(i,j) > Q(i,j-1)) \& j > 0:$
- maxValue.update(Q(i,j)
- Q(i,j).update ← maxValue

2.5.2 Dataset

Microsoft Taxi data, called T-drive trajectory data, are used in this study to present the proof of concept. This dataset contains one-week trajectory data of 10,357 taxis in Beijing. This dataset

contains about 15 millions records [2]. Due to the lack of a real trajectory dataset, the T-Drive dataset is used to simulate road users' movement. The COVID-19 dataset is acquired from the official website of the Beijing government [3].

2.6 Implementation

2.6.1 Road Network Construction

The first part of implementation is to construct the road network of the specific area. The road network is needed because it should be converted to a graph, and all other information should be mapped to the nodes and edges of the graph. In this paper, the road network is exported from Open Street Map (OSM). In order to get access to the data structure of the network, OSMnx[4] is used, which is an open-source python package. OSMnx helps to programmatically get access to the geospatial data structure of the OSM.

The next part is data cleansing as not all the imported information from OSM are required in this study. The derived road network has different attribute types for a single line (edge). For example, it can be drivable, walkable, etc. In this study, only walkable roads are used [5].

Once the data is converted as a shapefile format, it should be converted to GeoJSON format. All attributes and geospatial information such as spatial reference and geometry of features must be kept unchanged during the conversion. In this paper, one of the geoprocessing tools, feature to JSON conversion, is used to convert shapefile to GeoJSON [6]. The output of this step is the input of the next step. As in the next step, a geo-data-frame will be generated, GeoJSON format is needed.

The next step is to construct the geo-data-frame by using Pandas and GeoPandas. The GeoJSON format should be converted to the geo-data-frame because a road network graph can be generated from a geo-data-frame. Pandas is one of the robust data analysis and management frameworks

[2] https://www.microsoft.com/en-us/research/publication/t-drive-trajectory-data-sample/
[3] http://wjw.beijing.gov.cn/wjwh/ztzl/xxgzbd/gzbdyqtb/index.html
[4] https://osmnx.readthedocs.io/en/stable/
[5] https://github.com/gboeing/osmnx
[6] https://pro.arcgis.com/en/pro-app/latest/tool-reference/conversion/features-to-json.htm

in Python. Pandas stores data as a Data-Frame that is similar to tables with fields and records. GeoPandas, however, has more functionalities for geospatial analysis. GeoPandas has all the functionalities of Pandas as it is built on top of Pandas. GeoPandas contains many powerful spatial analysis functions, such as spatial aggregation methods and spatial joins that are useful to map trajectory data and COVID-19 data on the dataset.

Finally, the road graph has been constructed from the geo-data-frame by using NetworkX. The generated graph contains intersections of the road network as nodes and connections between intersections as edges. This graph is ready to get trajectory data and COVID-19 data mapped onto it.

2.6.2 Trajectory Data Reconstruction

After constructing the road network, next step is to reconstruct trajectory of of road users using origin and destination information. To do so, first, a geodatabase should be created from the Comma-Separated Values (CSV) file format origin-destination information. The CSV file is imported into ArcGIS, and a feature class is created for trajectory data. The generated feature class is then converted to the GoeJSON. Finally, the GeoJSON, for the same reason in the previous section, is converted to a geo-data-frame. Now, two geo-data-frame are created; one for road networks and the other one for trajectory data. The next step is to map the trajectory data on the geo-data-frame of the road network. The following section describes the process of mapping data.

2.6.3 Find Nearest Node To Each Trajectory Point

To map trajectory data on the road network data, we need to find the nearest node in the road network to one point of the trajectory data as presented in Fig 2.2. To do so, the Dijkstra algorithm has been used for shortest path project identifying alternative paths.

2.6.4 Construct Q matrix of QUBO problem

This is the essential part of the implementation, where the Q matrix should be defined based on the linear and quadratic coefficients of the QUBO problem. The Q matrix is upper triangular. Assume that the matrix is an empty matrix on the first step. The linear coefficients will be added to the diagonal of the matrix, and the quadratic coefficients will be added to the off-diagonal entries.

2.7 Results and Discussion

The results of the paper show that Quantum Processing Unit (QPU) is faster than every other solver. QPU is only applicable for small problems, and this limitation is caused by QPU architectures. Mapping the logical layer (QUBO) into the physical layer (QPU) requires embedding on chimera or pegasus architecture. Before sending the problem to the QPU, the availability of such mapping should be checked. This process can be done by a minorminer[7]. It is essential to mention that if the mapping does not exist, it does not mean that the problem cannot be solved. The obstacle to mapping might be a couple of inactive qubits on the physical layer. It is possible to divide the problem into small batches, map them individually, and put them back together using classical computing. This approach is also known as hybrid QC.

In terms of scalability, hybrid QC is the fastest solution. Different hybrid solvers are available in D-wave quantum computers once the problem is formulated as a QUBO. Three quantum solvers have been investigated in this paper; QBSolv, D-wave hybrid, and Leap Hybrid solvers.

QBSolv is one of the decomposing solvers that find the minimum energy of a QUBO problem by breaking the problem into sub-problems. The sub-problems are solved on a classical computer by running a tabu algorithm. We first run the problem on a QBSolv because it helps us define the QUBO problem on a classical computer and debug it, and once everything works well, it will be passed to the quantum solvers. Figure 2.3 represents the results of the QBSolv solver for the lowest energies in 10 iterations. The horizontal axis shows the 20 lowest energies, and the vertical axis

[7] https://github.com/dwavesystems/minorminer

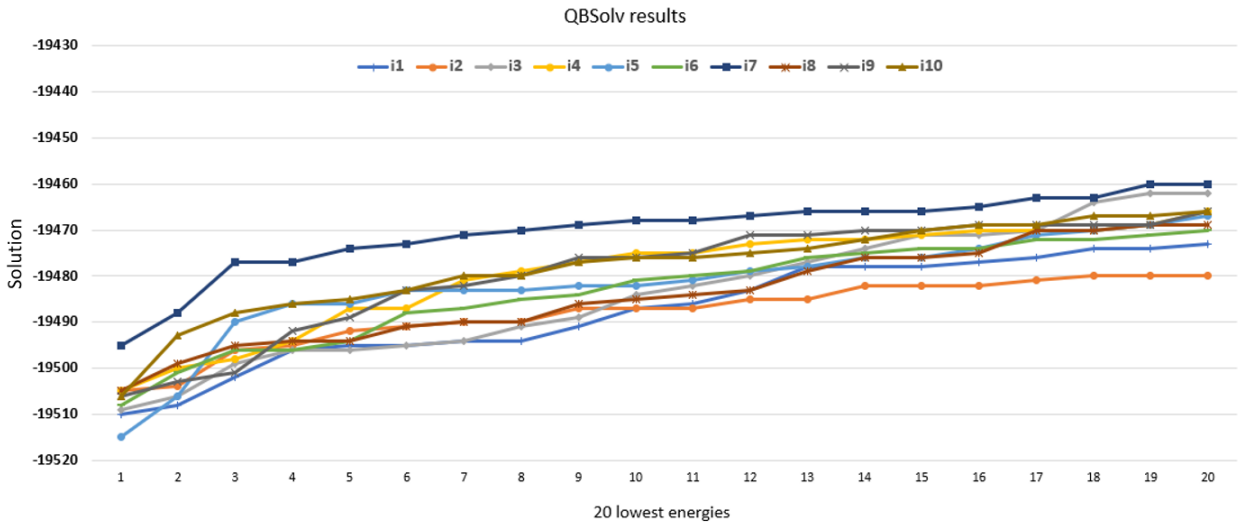


Figure 2.3: Results of the QBSolv solver for the lowest energies in 10 iterations. The horizontal axis shows the 20 lowest energies, and the vertical axis shows the level of energy (J).

shows the level of energy—the lower the energy, the better answer. As it is shown in Figure 2.3, iteration 5 found the best answer, and iteration seven found the worst answer.

Figure 2.4 depicts the result of running the proposed method on the Advantage-system-4.1 quantum computer for four convergence values in 10 iterations. Convergence in this solver means the number of iterations with no improvement that terminates sampling. The vertical axis shows the energy level of the results, and the horizontal axis shows the iteration number. The results show that convergence equals to 3 found the best answers on iterations one and two. Advantage-system-4.1 solver has 5760 qubits, from which 5619 qubits were active at the time of our run.

Figure 2.5 shows the result of running the problem on the Leap Hybrid quantum solver for ten iterations. The left vertical axis represents qpu-time, which is the time it takes to run the problem on QPU, and the right vertical axis shows run-time, which is the time it takes to break the problem into sub-problems the classical computer. As it is shown, qpu-time is always less than run-time.

the key lesson learned from the results are as follows. Since the hybrid solvers give the sub-optimal solution, the final solution is not the same in each iteration. However, the comparison between different solvers helps to better understand the results. The other important note is that

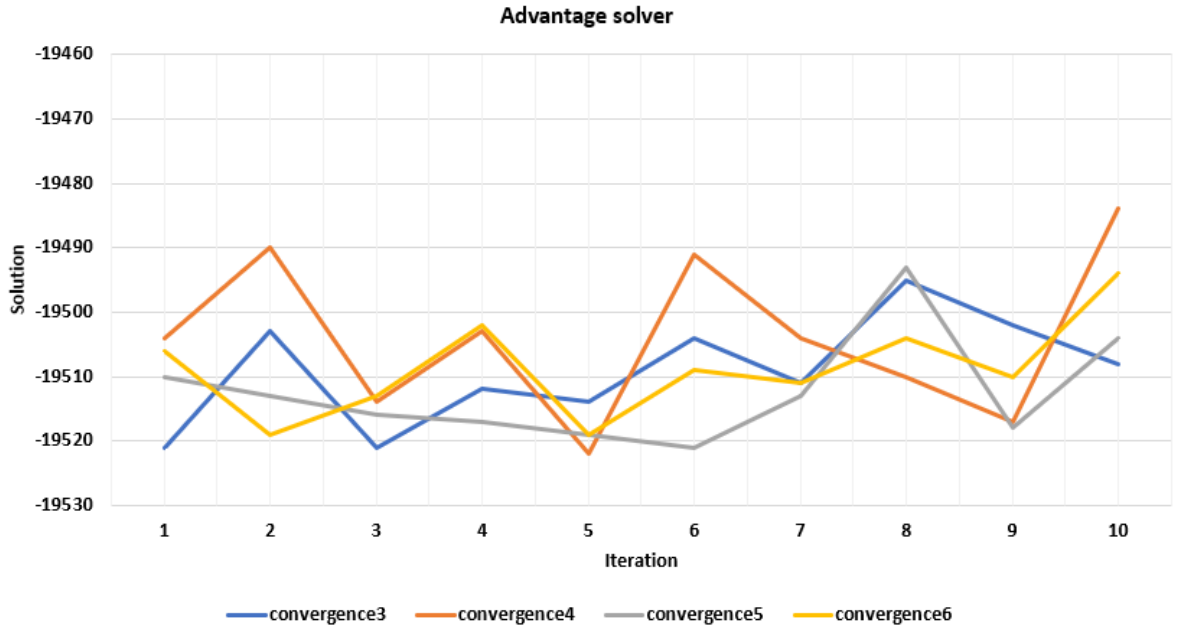


Figure 2.4: Results of running the proposed method on Hybrid-Advantage4.1 quantum solver. The solution is based on energy (J).

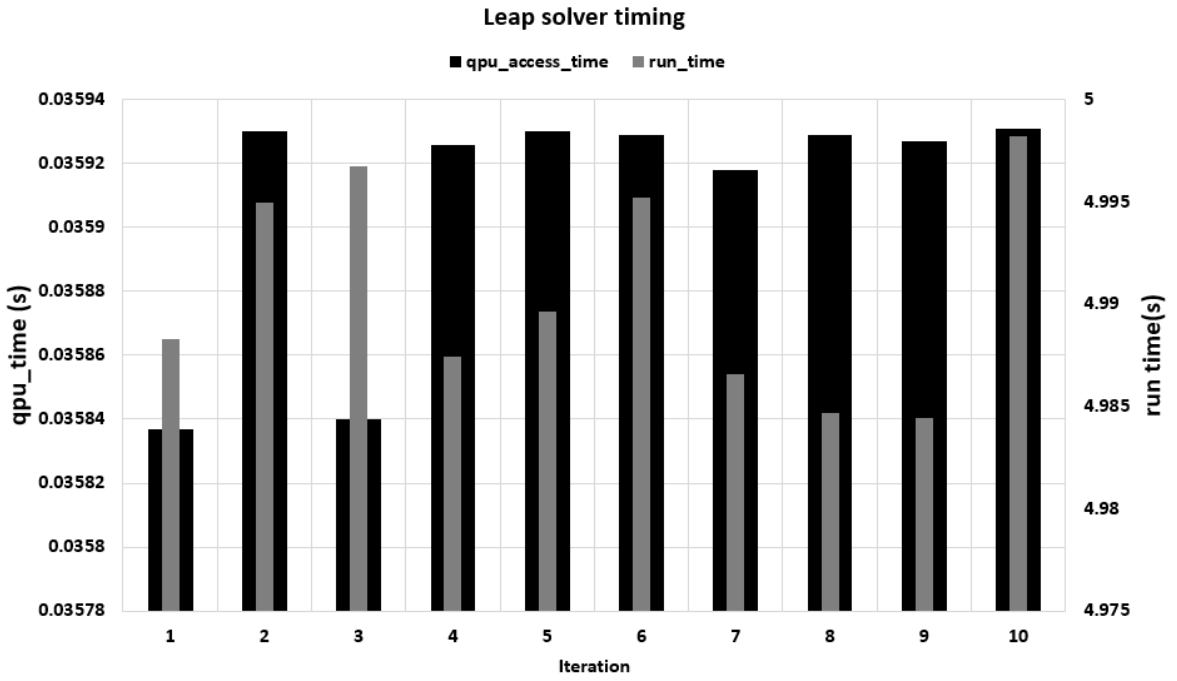
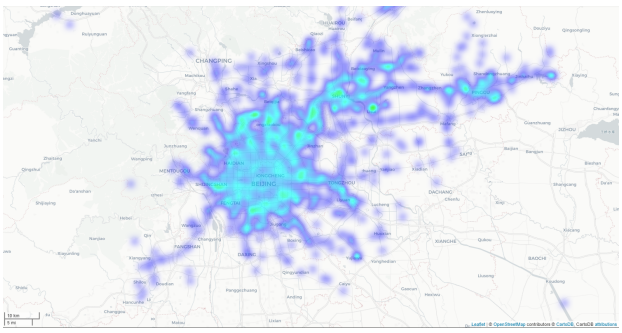
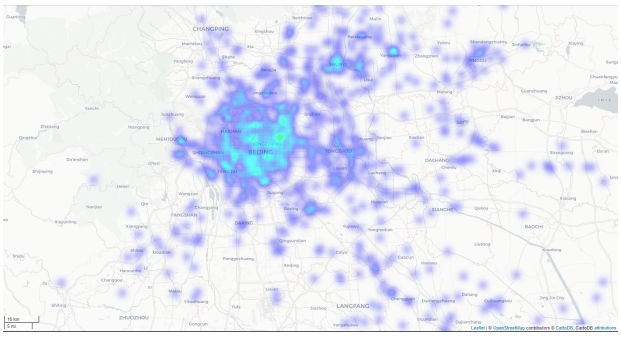


Figure 2.5: Run time, and QPU access time for ten iterations on Leap hybrid solver. The left vertical axis shows the run time values and the right vertical axis shows the QPU access time.



(a) Current state of trajectory data. Heatmap without using the proposed model.



(b) Heatmap of the cars trajectory by using the proposed model on a hybrid solver.

Figure 2.6: A comparison between two states. The expected results for whole datasets will be a heatmap with fewer hotspots for each time unit of the network. Figure 2.6a is used as ground truth to evaluate results.

the number of reads should be different for each problem, and it can be defined by try and error. This parameter forces QPU to run a problem based on its value. As the time scale for each read is microseconds, it is possible to increase the number of reads. Moreover, as the results of the QPU are probabilistic, more reads lead to more diverse solutions. The comparison between different solvers reveals that the solvers with Pegasus architecture are better than Chimera because of the topology and connections of the qubits. It is expected that once QPU can map more complex graphs, they will perform incredibly better than GPUs in these types of combinatorial problems.

In this research, two primary states were defined to evaluate results. The first state represents ROE by integrating the actual data layers; the second state, however, calculates ROE based on the proposed model. A comparison between two states is shown in Figure 2.6 for a subset of the datasets. A visualization of the datasets is available in Github [8].

2.8 Conclusion

In this study, we sought to bring QC into the geospatial problems to take advantage of this meta-heuristic approach and open the door for future works. To do so, we briefly introduce QC's fundamentals, followed by adiabatic QC and QA. The COVID-19 prevention problem has been converted to the QUBO. The results of the paper show that Quantum Processing Unit (QPU) is faster than

[8] https://amirhossein-nourbakhsh.github.io/Quantum-geo-AI/index.html

every other solver. However, the QPU is only applicable for small problems, and this limitation is caused by QPU architectures. Therefor, hybrid quantum solvers have been investigated such as D-wave hybrid and Leap Hybrid solvers. The heat maps indicate the better distribution of the vehicles on road networks after applying the proposed model. The result shows a promising avenue for such complex problem solving using a quantum computer. Further research will expand on broader transportation systems leading to more complex optimization with additional constraints.

Chapter 3

Dynamic Rebalancing Bike Sharing Systems Using Quantum Computing

3.1 Preface

This chapter explains the second sub-objective by presenting an optimized rebalancing algorithm for BSS, implemented on a QC platform.

The main contribution of this research is to introduce an optimization model to rebalance bikes in a BSS by solving it as a QUBO problem with different solvers on a QA computer. This novel approach formulates the problem as a QUBO problem. The primary objective of our optimization model is to minimize the risk of bike unavailability at each station, ensuring a balanced distribution that meets user demand. The proposed model avoids expensive redistributing/rebalancing processes and leads to high user satisfaction. Using real-world data from Toronto's BSS, our experiments demonstrate a significant decrease in the occupancy index by 25.18 percent, indicating improved bike availability. Note that this research is not proof of quantum speedup, however, it demonstrates the usage of QC and its performance to solve such problems as hybrid systems. Among the tested solvers, the Advantage system 4.1 on the Quantum Processing Unit (QPU) emerged as the most effective for our model, underscoring the potential of hybrid quantum solvers in this context.

The content of this chapter is submitted to The Journal of IEEE Transactions on Intelligent Transportation Systems, 2023 as follows:

A. Nourbakhsh, M. Jadidi, and K. Shahriari. "Dynamic Rebalancing Bike Sharing Systems Using Quantum Computing." *Journal of IEEE Transactions on Intelligent Transportation Systems, 2023, under review*.

3.1.1 Copyright and Licensing

This will be provided after the paper is published.

3.1.2 Contributions

The roles of the authors in this chapter are outlined as follows: Amirhossein Nourbakhshrezaei was responsible for the literature review, data collection and organization, methodological development, implementation for analysis and modeling, result validation and visualization, and the initial manuscript writing. Mojgan Jadidi oversaw the research, provided the funding, and contributed to the methodology development, as well as the writing and editing of the manuscript. Kyarash Shahriari supervised the research and assisted in the manuscript editing.

3.2 Abstract

Bike Sharing System (BSS) is an integral part of the Intelligent Transportation Systems (ITS). BSS fills the gaps in public transportation, avoids expensive costs for personal vehicles and also reduces carbon emissions and traffic congestion. The finite capacity for drop-off and the unavailability of bikes for pick-up are the main challenges in BSS leading to exceeding unsupported demands from users and dissatisfaction. To overcome this, an optimized rebalancing BSS algorithm is emerged where the computation using classical computer is costly and timely. In this paper, we introduce an optimization model to rebalance bikes in a BSS by solving it using QC platform as a QUBO problem with different solvers on D-Wave QA computer. Hence, the objective of the BSS rebalance optimization problem is to minimize the risk of the unavailability of bikes in each station for a given set of network users. Toronto bike-sharing data has been used as a proof of concept. Four different solvers have been investigated in this paper. The results showed occupancy-index has been decreased by 25.18 percent, which means more available bikes at each station. The results are also showing that hybrid quantum solvers are more promising. However, the best solver for running our model on a Quantum Processing Unit (QPU) is Advantage system 4.1.

3.3 Introduction

In 1999, the world leader in street furniture advertising company invented a self-service BSS called Cyclocity. The first generation of the system was launched in 2003 in Austria. Now, after more than two decades, the BSS is growing widely in metropolitan cities and is one of the crucial mode of transportation systems. BSS enables users to get a bike from their origin point of interest (called the origin station) and return it to another/same point of interest (called the destination station). BSS considers as foundation of sustainable transportation systems. The primary goal of such systems may differ at the early stage of the release, such as leisure and/or workout oriented; however, from the top-level point of view, it can affect the micromobility and multi-modal transit systems. BSS covers and fills the missing links and gaps in public transportation (i.e. transit) networks. In

addition, the expensive costs for owning personal vehicles can be avoided. Using BSS contributes also to reducing carbon emissions and traffic congestion. It is fast, affordable and easily accessible leading to a more inclusive transit mode. Notwithstanding the benefits of the BSS, it is hard to operate and efficiently manage such systems [124]. The problems in these systems can be categorized with different aspects. The first category is related to the problems that are caused by users, such as keeping the bikes longer than the specified period, or theft and vandalism. Most of these problems have been solved by using credit cards, geofencing, and tracking technologies. The second category is related to the problems caused by weather conditions and the geography of the area specially for places with harsh climate during winter season. however, Having all these problems in the BSS is inevitable, and the environmental conditions might not be appropriate in some areas or during some seasons. The third category contains problems caused by infrastructures in the system, such as the finite capacity for drop-off and storage.

The scope of this paper is limited to a potential solution for the third category that is related to infrastructure supporting BSS. To be more specific, finite capacity for drop-off and unavailability of bikes for pick-up is the main challenges in BSS. Each station cannot accept bikes more than its capacity. In each station, the availability of bikes is either congested (more than required) or starved (fewer than required), in which leads to increasing unsupported demands from users and dissatisfaction. The well-known solution for this issue is moving bikes by truck-fleet services to redistribute them on the stations which are dealing with an on-demand-capacitated constrained vehicle routing problem. Using truck-fleet distributors in the network is not an efficient approach because of traffic congestion, energy consumption, and air pollution caused by trucks fleet. Moreover, the truck fleet needs more human interaction and management, as a result, more cost for the BSS services. The demand for bikes is a complex context as it depends on different parameters such as location (density and topography), weather conditions, time of the day, etc. All these parameters lead to unbalance stations in terms of available bikes. Moreover, the availability of different modes of transportation makes the system more random and hard to simulate as a deterministic model.

Many optimization methods have been proposed to solve the unbalanced bike distribution problem.

Most of the solutions are static approaches that involve a third party to pick up bikes by trucks and moving them to the empty stations[125], [126]. There are also dynamic optimization approaches that work while the users are riding the bikes [127], [128]. Another solution for such a problem exists that is executed by asking users to balance the system. There is a reward for the users to pick up or drop-off at another station close to the user's preferred station. Rebalancing bikes in BSS is a complex and large-scale type of Vehicle Routing Problem (VRP) with a high degree of uncertainty. Moreover, considering the hundreds of bikes and stations, it is hard to be solved by traditional route optimization methods [124].

The classical computer-based solutions to this problem are well-known and already exist. However, as complexity has increased, they are not fast enough to provide real-time or near-real-time responses to problems at the individual or public level. On the basis of the current technological advancements in quantum computers and their extremely promising results in solving complex problems such as the rebalancing of BSS.

In QC, there are four primary computational models: 1) Gate-based/Circuit-based models, 2) Adiabatic models, 3) Models-based or Measurement-based, and 4) Topological models. QC has the potential to significantly impact transportation by enabling the optimization of complex problems more efficiently.

This paper provides an automated and optimized dynamic approach to rebalance the availability of bikes on stations by using a combinatorial optimization problem using QC and more specifically Qauntum Annealing (QA) which is a sub-category of Adiabatic models. The optimization model that is used to rebalance the bikes is formulated to a QUBO. QUBO is a type of optimization problem involving the minimization of a quadratic objective function subject to binary constraints. In other words, QUBO seeks the optimal configuration of binary variables that minimizes a specified cost function.

The rest of the paper is organized as follows. Section II provides the related works of various publications related to the rebalancing of BSS and QC in ITS. Section 3.5 describes the problem definition. Section 3.6 provides the contribution of the proposed method. Section 3.7 describes QC

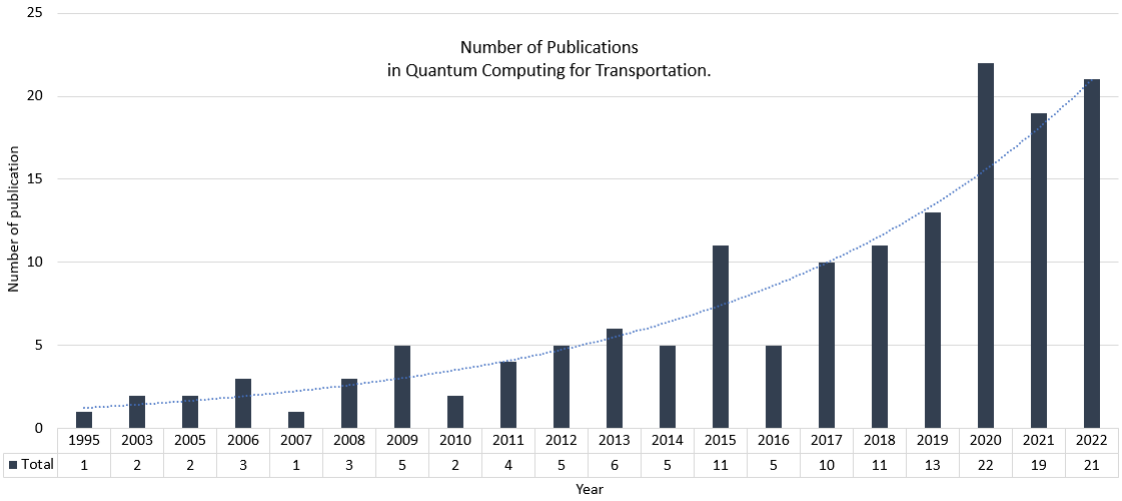


Figure 3.1: The bibliometrics from 1995 to 2022 shows the QC trend in Transportation. The data is derived from the web of science dataset (WOS). Term for query: (Quantum computing AND Transportation) search through the title, abstract, and keywords of the papers.

and, more specifically, QA and all available solvers. Section 3.8 discusses the proposed methodology adopted for rebalancing bikes using QUBO. Section 3.9 describes the implementation and results. Finally, Section 3.10 concludes with remarks highlights, lessons learned, and the perspective for future works.

3.4 Related works

Many researchers have been working on rebalancing, redistributing, and reallocating bikes in BSS. These methods usually use static and/or dynamic approaches to solve the BSS problems[124, 129]. The static approaches have been solved by using the integer programming methods[130]. Federico et. al[131] proposed a dynamic approach for rebalancing the bikes by predicting network conditions based on historical data. Another heuristic approach has been investigated in a research paper written by [132]. They trained a model to learn the traffic pattern and considered other factors such as weather conditions. Rebalancing and redistributing the bikes in BSS can be categorized into three groups depending on whether the bikes have been moved by trucks, trailers, or users (riders)[133]. Another classification can be done base on the location of the stations and the in-

ventory level[128]. Supposedly the bikes are meant to be moved by trucks, in that case, another problem should be solved that is related to the optimal vehicle route for trucks to redistribute the bikes by considering the travel time, station capacity, truck capacity, and any other constraints in Travel Salesman Problem (TSP)[134]. Bike Routing Problem (BRP) is similar to Vehicle Routing Problem, and both are NP-hard[135]. Moreover, considering the computational complexity of the dynamic rebalancing, the solution should be a heuristic or meta-heuristic approach[131, 127].Adish et al.[136] designed and proposed an architecture for BSS and tried to rebalance the bikes by using an incentive system. They engaged users (riders) to rebalance the bikes using regret minimization. They also deployed the proposed system on a large-scale BSS[136].

On the other hand, to augment the computation capacity for such optimization problems, QC has emerged to be used as an integral component of Intelligent Transportation Systems (ITS). Figure 3.1 shows the trend of QC research and publications in transportation. One of the most practical problems is the routing problem. Eneko [137] has published a paper describing a systematic literature review to analyze the publications related to the QC for routing problems. They analyzed different aspects of the literature by considering practical and theoretical approaches, single or hybrid usage of quantum computers, and computing devices[137]. Nourbakhsh et al. [1] have introduced a combinatorial optimization problem to reduce COVID19 risk and solved it on a D-Wave quantum annealer. Their goal was to minimize exposure to the virus in a road network. They used the Microsoft taxi dataset to simulate trajectory data [1]. Another research related to the QC in ITS was conducted by Singh et al. They used a QA method to control events of a real-world intersection, including vehicles and scooters[138]. Crispin [139] has investigated the potential applications of QC in Transportation. He described the possibilities for different applications for simulation and planning in transportation. The paper covered related publications on shortest paths, network analysis, and multi-objective routing optimization problems. Suen et al. [140] have explored a digital annealer for Fujitsu's quantum-inspired computer to solve a constrained routing problem. They worked on the vehicle sharing problem, and the results showed that the QUBO solver is more scalable than the Cplex solver. A novel application of the QC has been conducted by Xiaoming et

al.[141] by using quantum mechanics that is applied to Ant Colony Optimization (ACO) for TSP. Xinping proposed a programming model to minimize travel time in road networks and improve the network capacity. In order to do that, they employed a quantum-inspired model, and the results showed that the proposed algorithm reduces the total travel time [142]. Wang et al. [143] conducted another research related to the potential applications of QC in ITS. They reviewed path planning, transportation operation management, and transportation facility[143].

Ramkumar et al. [144, 145] have investigated the potential applications of Quantum Bayesian approaches for BSS demand prediction. They combined multiple time series models and Bayesian models to predict bike demand in BSS. The model they proposed was a probabilistic model that considers the uncertainties in the time series models[144, 145]. Ramkumar and Saideep have conducted another research to use the potential of quantum approximate optimization in BSS. They used a variational hybrid (quantum and classical devices) approach for solving rebalancing bikes in BSS. The IBM-Qiskit simulator was used in their research[146].

3.5 Problem Definition

Considering all relevant literature, there is an emergent research path on the Bike Routing Problem (BRP), that is a well-known NP-hard combinatorial optimization problem in traditional approaches using QC platforms. Multiple heuristic and meta-heuristic methods have been investigated to solve the BRP problem in classical models. However, the previous approaches still have trouble finding reliable solutions in real-world applications. Moreover, QC, and more specifically, QA computation are promising near-term meta-heuristic approaches to solve combinatorial optimization problems. In this research, we are rebalancing the BSS by solving BRP as a combinatorial optimization problem with different solvers on D-Wave quantum computers.

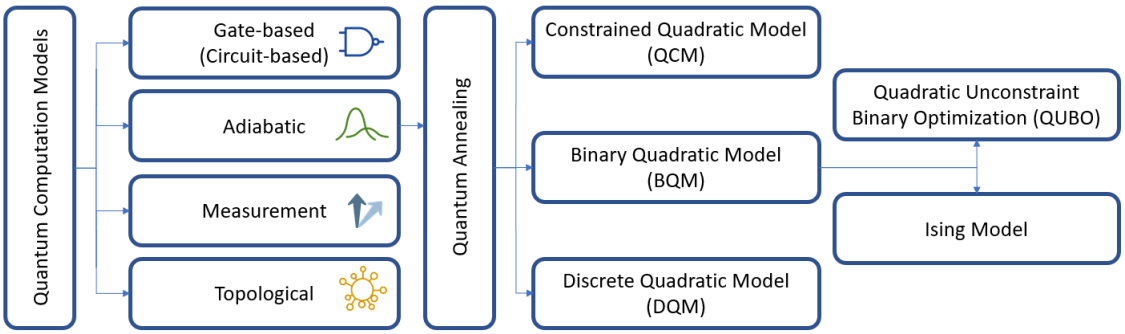


Figure 3.2: The overview of the scope of the paper. The QUBO is used [1].

3.6 Contribution

The main contribution of this research is to introduce an optimization model to rebalance bikes in a BSS by solving it as a QUBO problem with different solvers on a QA computer. Indeed, the objective of the BSS rebalance optimization problem is to minimize the risk of the unavailability of bikes in each station for a given set of network users. The proposed model avoids expensive redistributing/rebalancing processes and leads to high user satisfaction. Note that this research is not proof of quantum speedup, however, it demonstrates the usage of QC and its performance to solve such problems as hybrid systems.

3.7 Quantum Annealing

While the technology of the quantum hardware is scaling up, there is a strong motivation to analyze and discuss current NP-hard small problems that are computationally challenging even in the most powerful classical computers. There are four leading computational models available in QC. The first model is called the Gate-based model, also known as the circuit-based model[117]. The second model is the Adiabatic model[118]. The third type is called the Measurement-based model[80]. The fourth model is the Topological model[119]. QA is a sub-category of adiabatic quantum models3.2, and QA has shown a more practical approach to solving NP-hard problems among the current QC models. This section first explains difference between QC and classical computers and then

describes adiabatic models and provides more details of the QA. The main difference between quantum computers and classical computers is related to the smallest component of each. That is a qubit in quantum computers and a bit in a classical computer. This difference affects the process of solving a problem and has been investigated by many researchers[147].

There is no specific answer to comparing quantum computers and classical computers, and it is still under investigation [148]. The reason is the limited number of qubits or the noises that appear caused by a high number of entangled qubits of the current available quantum processing units (QPUs). A comparison with the MS algorithm [149] shows promising potential as approximately half of the instances have been solved in quantum computers faster than classical computers. Another research has proved quantum speedups for a specific graph optimization problem [114], as well as evidence of a scaling advantage of QA over classical simulated annealing for specific classes of more general problems.

Additionally, many works have been conducted to evaluate the performance of QA compared to heuristics algorithms in classical computers. Davide Venturelli [148] had an experimental comparison between classical computers and QA for job shop scheduling solvers. The results showed that in some cases, D-Wave quantum computers are outperformed by a classical algorithm, and the reason is that the problem sizes that they have worked on are too small for the asymptotic behavior of the traditional algorithms and, as a result, hard to compare.

3.8 Proposed Methodology

This paper proposes a Spatio-temporal approach that works with combinatorial optimization problems to solve QUBO models on an adiabatic quantum computer. It helps to decrease the risk of the unavailability of bikes in each station for a given set of network users. The proposed model avoids expensive redistributing/rebalancing processes and leads to high user satisfaction. The case study of the proposed model is BSS in the city of Toronto. Data layers that have been used in this research contain BSS trajectory, station information, and road network for bikes. The model is solved with

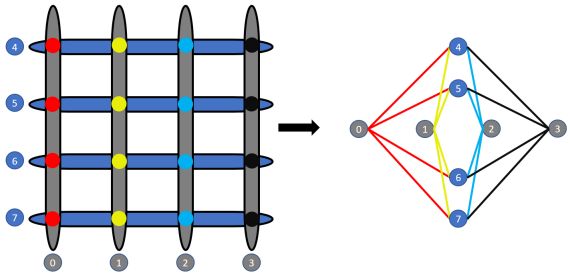


Figure 3.3: The chimera QPU architecture. This architecture has been used in D-Wave 2000Q QPUs. It shows a block of the system that has 8 qubits. Each line on the left side represents the superposition of the corresponding qubit and the circles at the intersection of the lines are showing the entanglement between two qubits.

different solvers in D-Wave quantum computers. To evaluate the model, the run-time parameter of each query on a cloud-based quantum computer is used. The current D-Wave quantum cloud computers are built based on adiabatic quantum models, specifically QA. This hardware is promising near-term meta-heuristic approaches to solve QUBO problems. In QUBO, the minimum of a discrete cost function should be found.

The problem can be formulated as an Ising (Equation 3.3) or QUBO (Equation 3.2) problem to take advantage of current D-Wave computers. The solvers that have been used in this research belong to D-Wave cloud quantum computers and are working with the Ising model. To be more specific, the input of the quantum hardware is an Ising model. However, due to the nature of QUBO, it is easier to formulate problems on QUBO than Ising (the parameters of the Ising model are -1 and +1; however, the parameters in QUBO are 1 and 0). On the other hand, these two models are convertible to each other. Therefore, we formulate our model to QUBO and Quantum Processing Unit (QPU) in D-Wave and map it to either the chimera graph (Figure 3.3) or the pegasus graph (Figure 3.4). As QA works with the energy of the system, an energy function is required. This energy function is named the Hamiltonian function (Equation 3.4).

The process of running a combinatorial optimization problem on QA can be defined in four main categories; 1) Writing the objective function and constraints of the problem. 2) Converting objective functions and constraints to an appropriate mathematical statement with binary variables. 3) Making the objective function and constraints QUBO appropriate (Making Q matrix). 4) Combin-

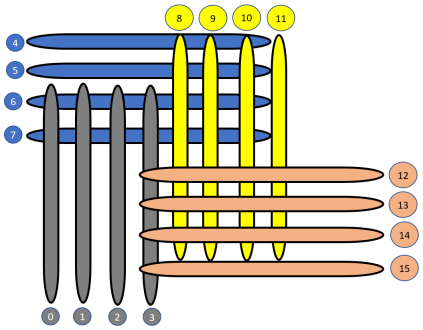


Figure 3.4: The pegasus QPU architecture. This architecture has been used in D-Wave Advantage QPUs. It is similar to chimera but the aligned lines (qubits) are shifted vertically or horizontally. Each line represents the superposition of the corresponding qubit and the intersection of the lines are showing the entanglement between two qubits.

ing objectives and constraints.

$$Cost\,function = x^T.Q.x \tag{3.1}$$

where x = an N x 1 vector of binary variables

Q = an N x N upper-triangular matrix of

real weights

Eq 3.1 can be defined more precisely as Eq 3.2.

$$f(x) = \sum_i Q_{i,i}x_i + \sum_{i<j} Q_{i,j}x_i x_j \tag{3.2}$$

where $Q_{i,i}$ = are the linear coefficients

$Q_{i,j}$ = are the quadratic coefficients or

non-zero off-diagonal terms

$$E_{Ising}(s) = \sum_{i=1} h_i s_i + \sum_{i=1} \sum_{j=i+1} J_{i,j} s_i s_j \tag{3.3}$$

where h_i = are the linear coefficients

$J_{i,j}$ = are the quadratic coefficients or

non-zero off-diagonal terms

Energy = Initial Hamiltonian + Final Hamiltonian

$$H_s(S) = [-\frac{1}{2}\sum_i \Delta(S)\sigma_i^x] + [\varepsilon(S)(-\sum_i h_i\sigma_i^z + \sum_{i<j} J_{ij}\sigma_i^z\sigma_j^z)] \tag{3.4}$$

For each bike, there are multiple alternative stations from which only a few of them are considered feasible destinations. We selected the five nearest alternative stations to the user's destination. Then, the shortest path from one origin to five destinations is found. To formulate the cost function for BSS rebalancing, we defined a cost function based on the number of available bikes and the real-time capacity of the stations. We define a new function to minimize the Risk Of Shortage (ROS) as Equation 3.5.

$$ROS = \sum_{b=1}^{k}\sum_{i=1}^{\frac{n}{2}} \begin{cases} \sum\limits_{j=i+\frac{n}{2}} x_i x_j, & \text{for } i < \frac{n}{2} \\ \sum\limits_{j=i+1} x_i x_j, & \text{for } i = \frac{n}{2} \end{cases} \tag{3.5}$$

where k = is the number of bikes

n = is the number of alternatives

We defined the constraint of the model to minimize the capacity of each station to keep them less than Average Capacity (AC) of the all stations.

$$Constraint = \sum_b (\sum_s x_i - AC)^2 \tag{3.6}$$

$$Cost function = ROS + Constraint \tag{3.7}$$

3.8.1 Pseudo Code

Algorithm 2 Shows the Pseudo code of the proposed model.

Algorithm 2 Q construction Pseudo Code

$G \leftarrow NetworkXGraph$
$dataset \leftarrow csvfile$
$S \leftarrow Size-of-the-input-based-on-number-of-users$
$Q \leftarrow [GenerateanemptyQMatrixbasedonS]$
for i in array-bikes:
- for j in routes:
- for k in stations:
- $node1 \leftarrow k.getNode(0)$
- $node2 \leftarrow k.getNode(1)$
- $node3 \leftarrow k.getNode(2)$
- $node4 \leftarrow k.getNode(3)$
- $node5 \leftarrow k.getNode(4)$
- risk $\leftarrow mapStationInfo(node1, node2, node3,$
- $node4, node5)$
- weight $\leftarrow$ weight + risk
- G.addEdge(k,weight)
- weight(j).update
- Q(i,j).update
- $if(Q(i,j) > Q(i,j-1))\&j > 0:$
- maxValue.update(Q(i,j)
- Q(i,j).update $\leftarrow$ maxValue

3.8.2 Datasets

Three datasets have been used in this research. The first dataset is a trajectory dataset. The second one is the real-time information of the bike stations in Toronto. The third one is the dataset that has been used to create a network of bike roads in Toronto. We used Toronto bike-sharing data as a proof of concept to rebalance and redistribute loads by considering five alternative paths for each trajectory. The dataset contains 157398 trips. Each trip has trip duration, start statino id, start time, start station name, end station id, end time, end statin name, and bike id. The data is acquired in 2021. Real-time bike station data has been used in this research. Once the query is sent to the server, the results will be returned as a JSON object. The dataset includes the ”last updated”

Table 3.1: Dataset format

Stations	Station id	Name	Configuration	Latitude	Longitude	Address	Capacity	Is charging	Rental methods
	7000	Fort York	REGULAR	43.639	-79.395	Fort York	35	False	Key
Trajectory	Trip Id	Trip Duration	Start Id	Start Time	Start Station Name	End Id	End Time	End Name	Bike Id
	10824239	506	7000	8:27	Fort York	7005	8:36	King St W	4022

parameter that shows the time in a Long-Integer format. The JSON object contains another JSON object called "stations". It includes information about each station. Each station has station id, name, physical configuration, latitude, longitude, address, capacity, is charging station, and rental methods (Table 3.1). At the time of writing this paper, the information about 628 stations were available and has been used. Open Street Map has been used to construct a road network graph for bikes. The nodes of the graph represent the road intersections, and the edges of the graph are the bike roads.

3.8.3 Evaluation

To evaluate the final results of the model, an index is defined based on the occupancy of the station to assess the distribution of the bikes through the network. This index is called occupancy-index (3.8). Moreover, a comparison map is created to check the distribution of the bikes at the end of the trips for the actual trips and for the QC-suggested routes.

$$Occupancy\ \ index = \sum_{s} (availablebikes(s)/emptyracks(s)) \tag{3.8}$$

where s = is the number of stations

3.9 Implementation and Results

3.9.1 Road Network Reconstruction

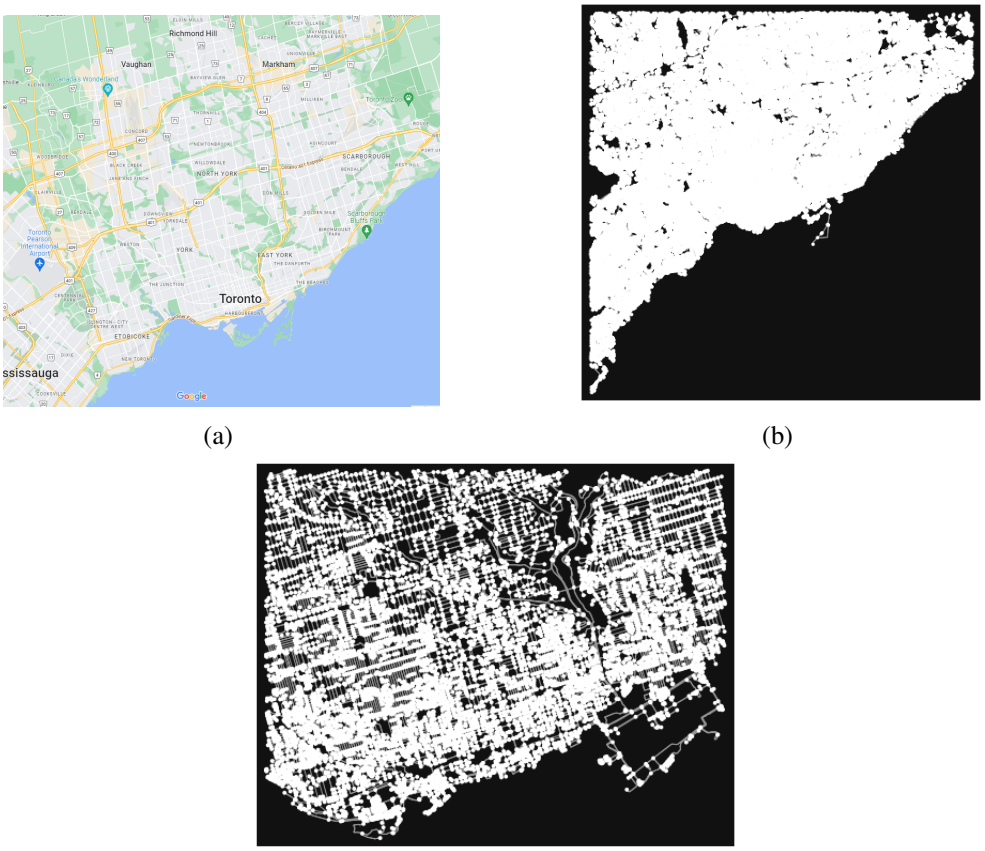
The first step of the implementation is to reconstruct a road network for bikes in a area of case study. The road network works as a base map for the discrete optimization problem, and it must be converted to a graph, while all meta-data and attributes should be mapped to the nodes and edges of the graph.

In this research, the bike's road network is derived from Open Street Map (OSM). OSMnx is used to get access to the OSM dataset. It is an open-source Python library that helps to import, customize, manipulate, and visualize geospatial data from OSM[1]. The next step is data sanitization which allows for cleaning the dataset to avoid redundancy and cut-off. This part has been done by importing the shapefile of the data to ArcGIS Pro [2], and the output is converted to the GeoJSON format.

All features and geospatial details, such as spatial connection and geometry of features, should be kept intact during the conversion. In this research, one of the geoprocessing tools, feature to JSON conversion, is employed to convert shapefile to GeoJSON [3].

Bear in mind that not all the extracted attributes and geometries are required in this research. As an example, in OSM, different edges are available with different attributes based on transport mode (walk, bike, drive, all); thus, we only used edges that are suitable for biking[4]. The outcome of this step is the input of the next stage. In the next stage, a Geo Data Frame (GDF) should be generated. The following step is to create the GDF by using Pandas and GeoPandas. The GeoJSON structure should be converted to the GDF because the bike road network can be generated from a GDF. Pandas is one of Python's most powerful data analysis and management libraries. Pandas transforms the data structure into a Data-Frame equivalent to tables with fields and records. GeoPandas, however, has more pre-defined functions for geospatial and Spatio-temporal analysis such as spatial

[1] https://osmnx.readthedocs.io/en/stable/
[2] https://pro.arcgis.com
[3] https://pro.arcgis.com/en/pro-app/latest/tool-reference/conversion/features-to-json.htm
[4] https://github.com/gboeing/osmnx



(a) (b)

(c)

Figure 3.5: Geocoding result for five keywords (Toronto, Markham, Richmond Hill, Vaughan, and City of Toronto.) a) shows the case study area. b) shows a constructed graph with 20000 nodes. c) the graph with 5000 nodes.

aggregation techniques and spatial joins. Moreover, GeoPandas has most of the functions of Pandas as it is created on top of Pandas. Eventually, NetworkX is employed to construct the graph from GDF. The constructed graph includes intersections of the road network (Nodes) and accessibility between intersections (Edges).

For covering all the area of bike sharing in Toronto, the geocoding process is done with five keywords; 1) Toronto 2) Markham 3) Richmond Hill 4) Vaughan 5) City of Toronto. Figure 3.5a shows the case study area. The generated graph has 2000 nodes (Figure 3.5b). To better visualize the graph, a generalized graph of the area with 5000 nodes is represented in Figure 3.5c. This graph is prepared to get stations information and bike trajectory data mapped onto it.

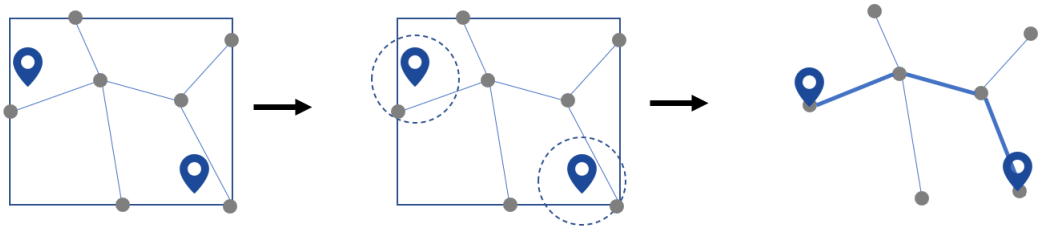


Figure 3.6: Finding the nearest node to each trajectory point on the road network. As the latitude and longitude of the trajectory data do not align with the nodes of the road network, the nearest node to each trajectory record should be calculated.

3.9.2 Trajectory Data Reconstruction

The original dataset contains trip ID, trip duration, trip station ID, start time, start station name, end station id, end time, end station name, bike ID, and user type. On the other hand, a graph is constructed by using the road network in OSMnx, in which the intersections of the networks are the nodes of the graph, and if there is a path between intersections, an edge is added to the nodes. The locations of the start point and end point of the trip are appended to the dataset. For each bike, alternative destinations should be added which are the closed stations to the end station of original trip. Thus, the trajectory data is stored using the point-in-polygon (PIP) algorithm for the main trip and alternative trips. In the PIP algorithm, the center of the polygon is the end station, and if the buffer (polygon) is empty, the end station will be considered as the destination for the alternative path. Then the shortest path function (Dijkstra) in the OSMnx is used to find the path between the origin and alternative destinations.

3.9.3 Find Nearest Node To Each Trajectory Point

Each node of the trajectory data has a latitude and longitude, which are not aligned with the original road network. Figure 3.6 shows the road network graph in gray color and the trajectory data in blue. The blue pins are then matched to the nearest node of the road network graph. If the pin is a start point or end point, the mapping should be done on the exact node of the graph where the station is mapped.

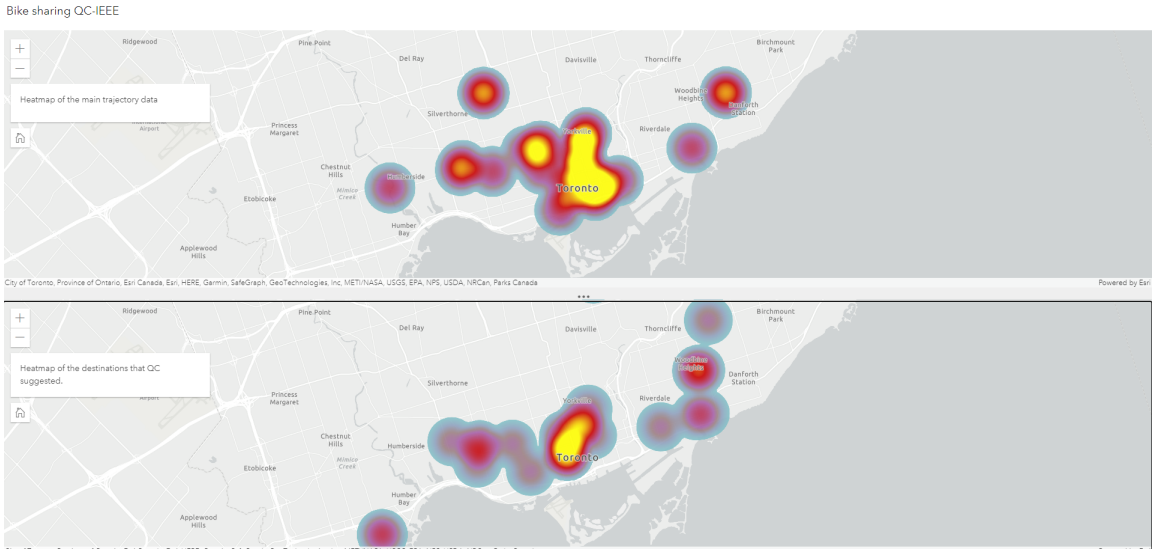


Figure 3.7: Results of trajectory data in two different states. The top image shows the heatmap of the final destination of the actual trips, and the bottom image depicts the heatmap of the suggested destination by QC.

3.9.4 Quantum Computer Solvers, Results, and Discussion

Figure 3.7 indicates the distribution of trajectory data, highlighting areas of high concentration for two scenarios. The one on the top shows the heatmap based on real-world data, in contrast, the bottom image represents the optimized distribution of destinations suggested by QC, showcasing a more even spread of destinations across the area. This suggests that the QC optimization has effectively redistributed the destinations to achieve better balance and potentially reduce congestion.

Four different solvers have been investigated in this research. One of these solvers is a hybrid solver that is called ”Hybrid binary quadratic model version2”. The other three solvers are built and work under QPU architecture, which means these are not hybrid solvers; ”Advantage system4.1”, ”Advantage system5.2”, and ”Advantage system6.1”.

The Advantage system4.1 solver works with QPU and is located in North America. The total number of qubits is 5760. At the time of writing this paper, 5627 qubits were active, and the qubit temperature was 15.4 MK. It supports both the Ising and QUBO models. The topology type is a pegasus graph. The results for 10 iterations and 100 reads showed that the best solution (minimum energy) that was found in this solver is -114.06 J (Figure 3.8).

The Advantage system5.2 solver works with QPU and is located in Europe. The total number of

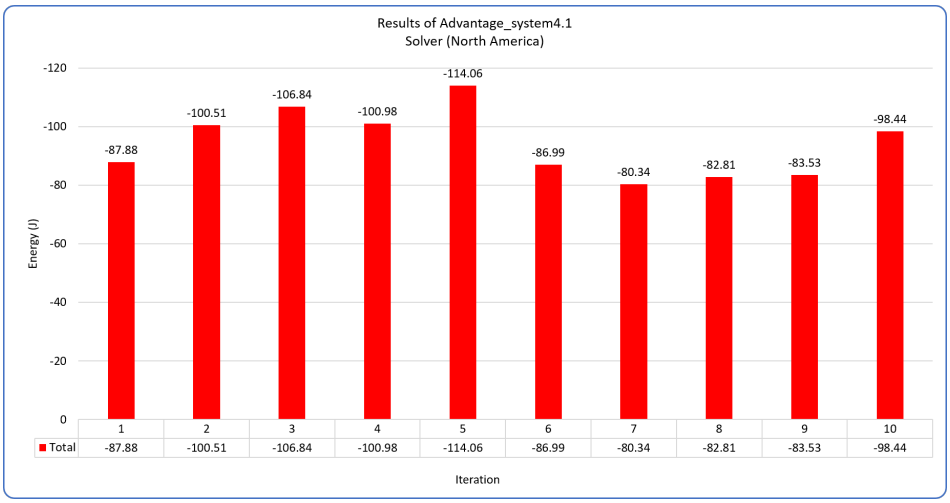


Figure 3.8: Results of running the model on Advantage system4.1. The vertical axis shows the energy in Joule and the horizontal axis represents iteration. Each energy is calculated by taking average of 100 reads.

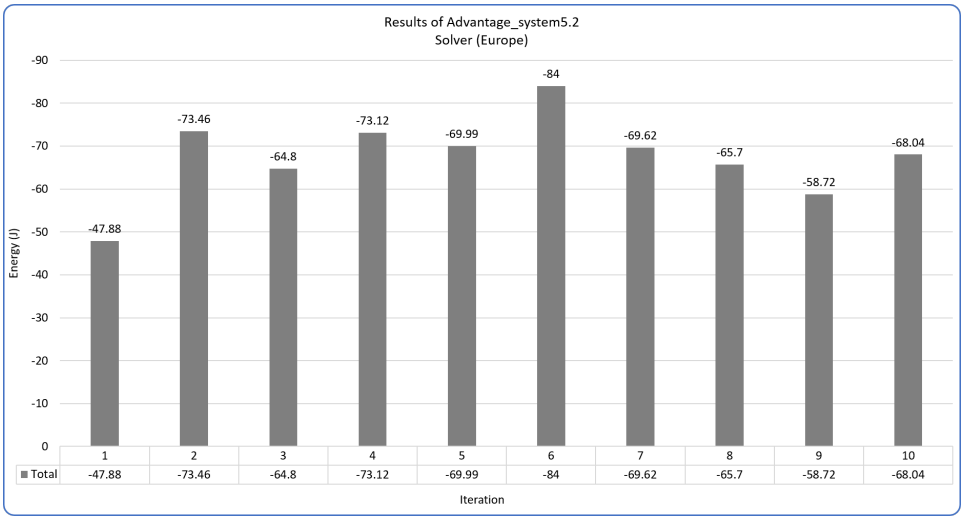


Figure 3.9: Results of running the model on Advantage system5.2. The vertical axis shows the energy in Joule and the horizontal axis represents iteration. Each energy is calculated by taking the average of 100 reads.

qubits is 5760. At the time of writing this paper, 5617 qubits were active, and the qubit temperature was 16.4 MK. It supports both the Ising and QUBO models. The topology type is a pegasus graph. The output of running the model on this solver indicates that the best solution that has been found in 10 iterations and 100 reads is -84 J (Figure 3.9).

The Advantage system6.1 solver works with QPU and is located in North America. The total number of qubits is 5760. At the time of writing this paper, 5616 qubits were active, and the qubit

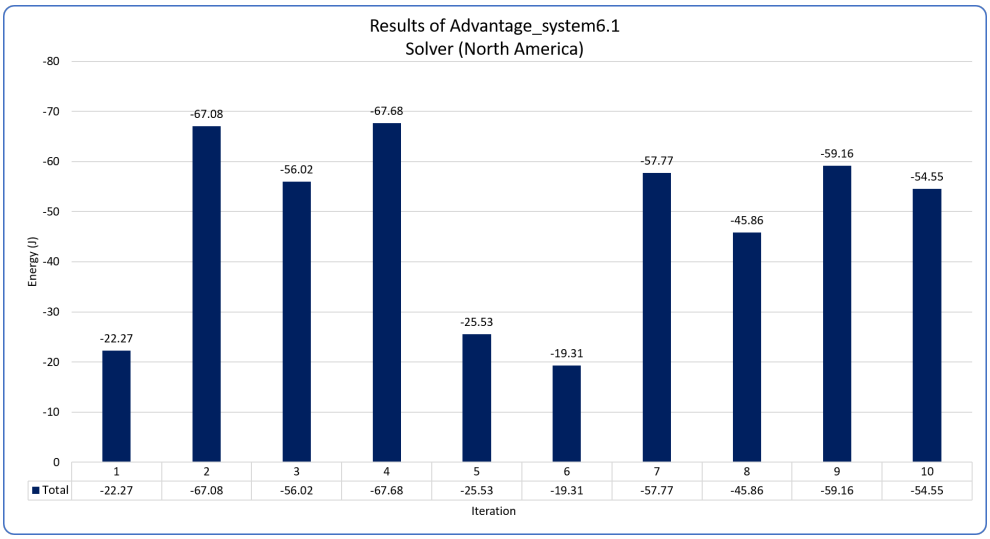


Figure 3.10: Results of running the model on Advantage system6.1. The vertical axis shows the energy in Joule and the horizontal axis represents iteration. Each energy is calculated by taking average of 100 reads.

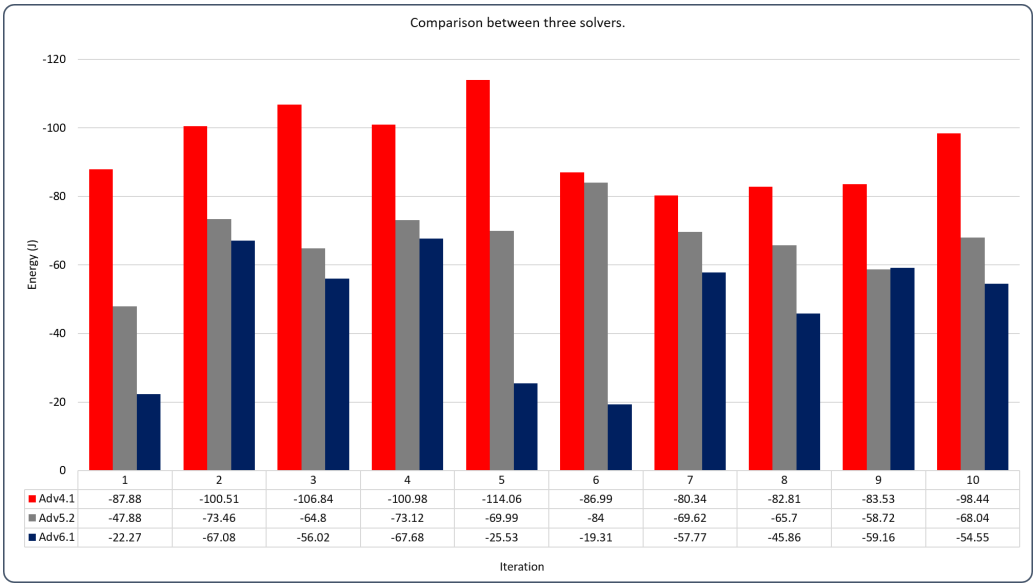
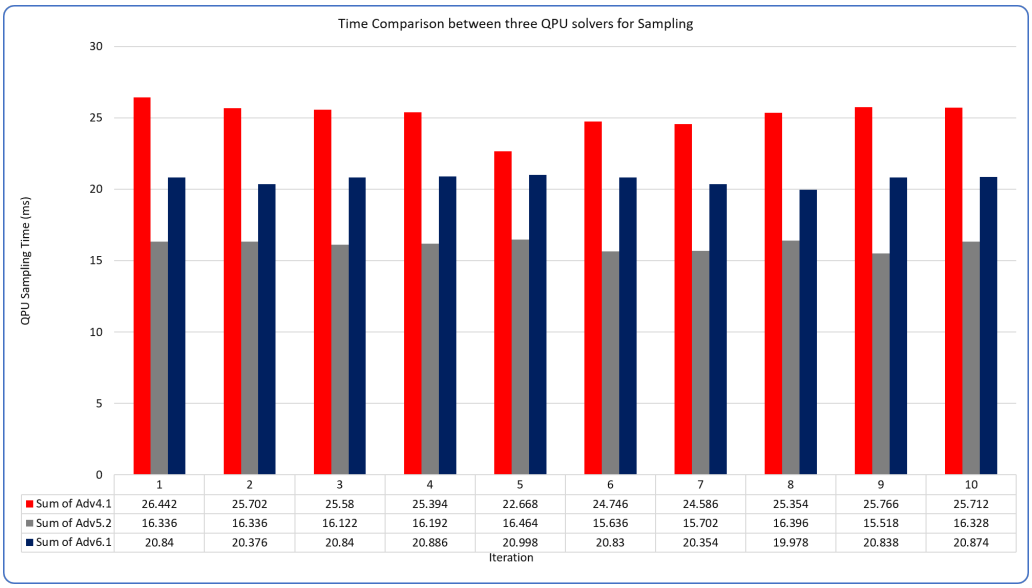


Figure 3.11: Comparison between three QPU solvers in terms of Energy. The lowest the energy, the better answer. As it is shown, Advantage system4.1 worked better than two other solvers.

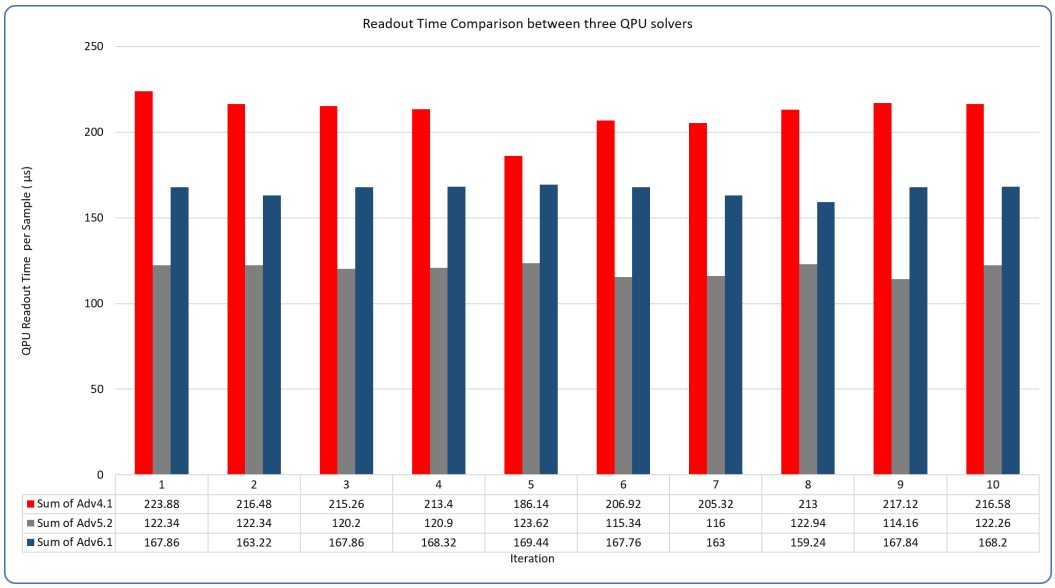
temperature was 16.0 MK. It supports both the Ising and QUBO models. The topology type is a pegasus graph. The results depict that the best solution that this solver has found is -67.68 J (Figure 3.10).

Comparing the energy of three QPU solvers shows that Advantage system4.1 is by far the best QPU solver for our application as it returned lower energy for all iterations (Figure 3.11).



	1	2	3	4	5	6	7	8	9	10
Sum of Adv4.1	26.442	25.702	25.58	25.394	22.668	24.746	24.586	25.354	25.766	25.712
Sum of Adv5.2	16.336	16.336	16.122	16.192	16.464	15.636	15.702	16.396	15.518	16.328
Sum of Adv6.1	20.84	20.376	20.84	20.886	20.998	20.83	20.354	19.978	20.838	20.874

Figure 3.12: Comparison between three QPU solvers in terms of sampling time. As it is depicted, Advantage system5.2 is the fastest one in sampling.

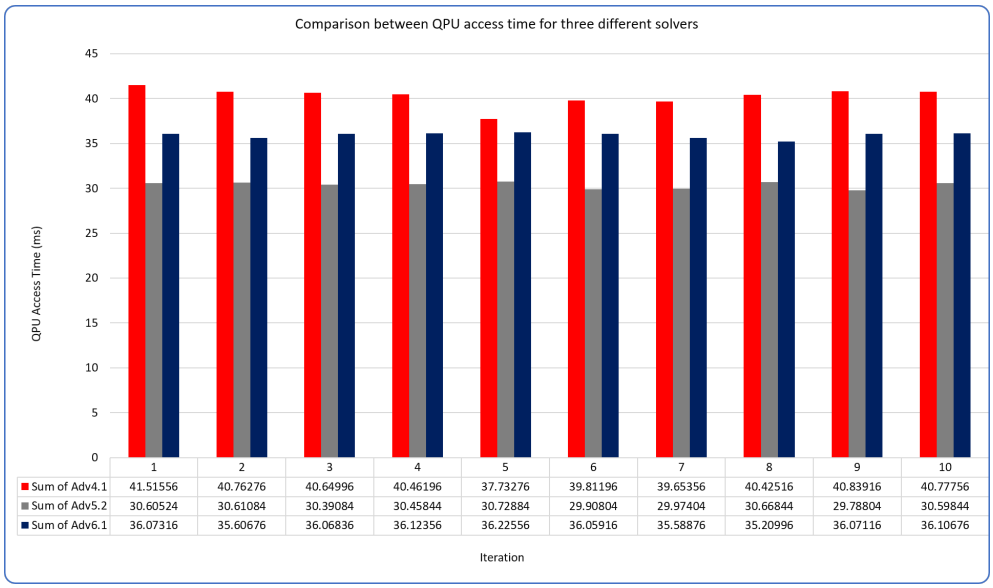


	1	2	3	4	5	6	7	8	9	10
Sum of Adv4.1	223.88	216.48	215.26	213.4	186.14	206.92	205.32	213	217.12	216.58
Sum of Adv5.2	122.34	122.34	120.2	120.9	123.62	115.34	116	122.94	114.16	122.26
Sum of Adv6.1	167.86	163.22	167.86	168.32	169.44	167.76	163	159.24	167.84	168.2

Figure 3.13: Comparison between three QPU solvers in terms of Readout time. As it is shown, Advantage system5.2 is the faster than two other solvers in readout.

Comparing the QPU sampling time of three QPU solvers shows that Advantage system5.2 has the lowest sampling time for all iterations compared to the other QPU solvers (Figure 3.12).

Figure 3.13 shows a comparison between QPU solvers in terms of readout time. It is depicted that Advantage system5.2 has the lowest readout time. However, the values are very close together, and it is negligible.



	1	2	3	4	5	6	7	8	9	10
Sum of Adv4.1	41.51556	40.76276	40.64996	40.46196	37.73276	39.81196	39.65356	40.42516	40.83916	40.77756
Sum of Adv5.2	30.60524	30.61084	30.39084	30.45844	30.72884	29.90804	29.97404	30.66844	29.78804	30.59844
Sum of Adv6.1	36.07316	35.60676	36.06836	36.12356	36.22556	36.05916	35.58876	35.20996	36.07116	36.10676

Figure 3.14: Comparison between three QPU solvers in terms of QPU access time. It takes less amount of time in Advantage system5.2 to run the model on QPU.

Another important parameter in analyzing the potential of solvers is the QPU Access Time (QAT). It indicates the time that model has access to the actual quantum computer. Once a query is sent to the D-Wave cloud computers, it goes to a queue, and the queries will be passed to the QPU in an order. As it is shown in Figure 3.14, Advantage system5.2 is faster than two other solvers. In terms of post-processing time, the results show that Advantage system4.1, with an average of 1.1022 ms, is the fastest one (Figure 3.15).

Hybrid binary quadratic model version2 only supports BQM models, and it has the ability to get 1000000 variables and 200000000 biases.The results show that these solvers return more than one solution, but the total energies of all iterations are the same (-243 J). The final answer of the hybrid solvers are not the same in each iteration because these type of solvers give the sub-optimal solution. However, the comparison between various solvers helps to understand the results reasonably. The results showed occupancy-index has been decreased by 25.18 percent compared to the actual trajectory data, which means more available bike at each station.

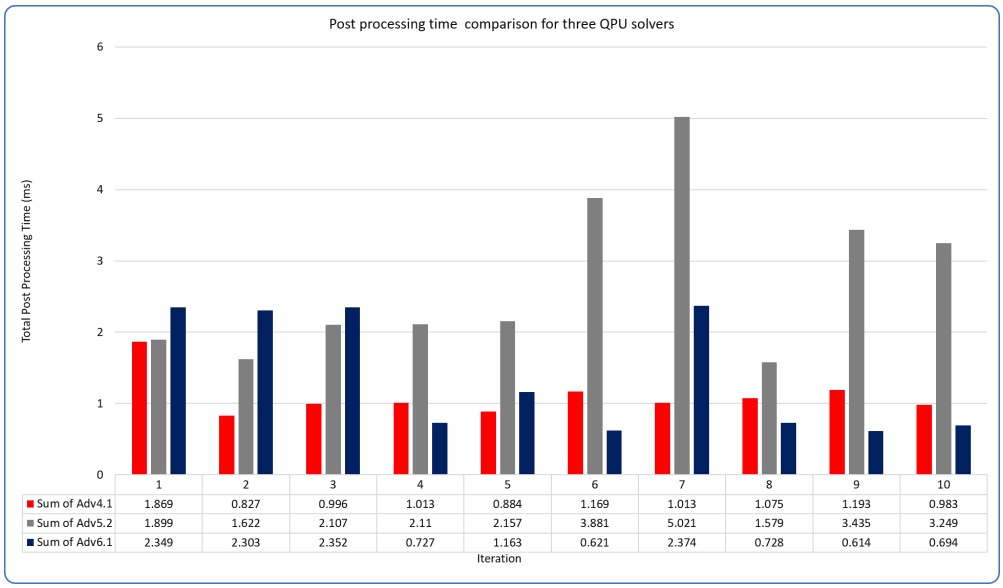


Figure 3.15: Comparison between three QPU solvers for total post processing time. The time is fluctuating in different iterations, however in overall, the Advantage system4.1, with an average of 1.1022 ms is the fastest one.

3.10 Conclusion

In this study the application of QC systems to deal with NP-hard problem such as rebalancing BSS has been investigated. An optimization model to rebalance bikes in a BSS has been proposed by solving such problem as a QUBO problem with different solvers on a QA computer. By doing so, the objective of the BSS rebalance optimization problem to minimize the risk of unavailability of bikes in each station for a given set of network users has been studied in four different solvers. As summary, the results of this study show that QC can be applied to rebalance BSS as it was investigated in a real-world application. The best solver for running our model on a QPU is Advantage system4.1. However, comparing the QPU with a hybrid approach shows that the hybrid solvers are more reliable, and they could find a better solution (lower energy) for the problem. The other significant note is that the number of reads should be defined for each problem and can be defined by try and error. Since the time unit for each read is microseconds, it is feasible to increase the number of reads. It is expected that once QPU can map more complex graphs, they will perform incredibly better than GPUs in these types of combinatorial problems. The proposed model provide avoiding expensive redistributing/rebalancing process and leads to high user satisfaction.

In this research, we aimed to bridge the gap between QC and geospatial problems to take advantage of this meta-heuristic approach and open the door for future research. Further research will extend to more comprehensive transportation systems leading to more complex optimization with additional constraints.

Chapter 4

Clustering Bike Sharing Stations Using Quantum Machine Learning: A Case Study of Toronto, Canada

4.1 Preface

This chapter describes the third sub-objective, exploring the potential of QC in ML particularly in clustering real-world data within BSS. As QC technologies rapidly evolve, identifying applications that can fully leverage their capabilities becomes increasingly critical.

To cluster stations in BSS considering their real-time characteristics and geospatial components identifies as a combinatorial optimization problem. The k-means algorithm is a well-known clustering approach for such systems. However, K-means is an NP-hard clustering algorithm that cannot be solved as a deterministic approach. In other words, in practical applications, the earlier methods are still unable to find deterministic solutions. This issue has been investigated using a variety of heuristic and meta-heuristic techniques in classical models. QC and, more specifically, QA computation offer promising short-term meta-heuristic solutions to combinatorial optimization issues. The main contribution of this research is to introduce a hybrid model to cluster stations in

a BSS by solving it as a CSP problem with different methods on a QA computer using real-time dataset. In addition to the practical contribution, this research also offers theoretical advancements in the field of computational optimization by defining a new topology for the input data that is compatible with QC topology (Chimera topology). The workflow of converting classical feature vectors of stations into a new graph is also described.

Utilizing real-time data from Toronto's BSS, we employ various clustering methods and dissimilarity functions to evaluate the model's performance. The results demonstrate the feasibility and efficiency of QC in solving complex clustering problems, achieving a notable reduction in the occupancy index and ensuring better bike availability across stations.

The content of this chapter is accepted for The Journal of Transportation Research Interdisciplinary Perspectives, 2024 as follows:

Amirhossein Nourbakhsh, Mojgan Jadidi, Kyarash Shahriari. "Clustering Bike Sharing Stations Using Quantum Machine Learning: A Case Study of Toronto, Canada" *Journal of Transportation Research Interdisciplinary Perspectives, 2024*.

4.1.1 Copyright and Licensing

4.1.2 Contributions

The roles of the authors in this chapter are outlined as follows: Amirhossein Nourbakhshrezaei was responsible for the literature review, data collection and organization, methodological development, implementation for analysis and modeling, result validation and visualization, and the initial manuscript writing. Mojgan Jadidi oversaw the research, provided the funding, and contributed

[1] https://www.sciencedirect.com/journal/transportation-research-interdisciplinary-perspectives/publish/open-access-options

to the methodology development, as well as the writing and editing of the manuscript. Kyarash Shahriari supervised the research and assisted in the manuscript editing.

4.2 Abstract

Quantum Machine Learning (QML) is a field that combines the principles of QC and Machine Learning (ML). QC works by taking advantage of the properties of quantum physics. Since QC technologies and devices continue to develop quickly, it is important to identify the use cases and applications that benefit the most. This paper investigates the potentials of QC, and more specifically, QA, for clustering real-world data in transportation systems. The BSS is used as a case study applying a clustering model on QA computers. The main contribution of this research is to introduce a hybrid model to cluster stations in a BSS by solving it as a Constraint Satisfaction Problem (CSP) problem with different methods on a QA computer using a real-time dataset. In addition to the practical contribution, this research also offers theoretical advancements in the field of computational optimization by defining a new topology for the input data that is compatible with QC topology (e.g., Chimera topology). Three different methods have been used to determine the number of clusters, and Euclidean, Manhattan, Pearson, and Spearman dissimilarity functions have been applied to cluster the stations. The evaluation is done using the magnitude vs. cardinality approach. The distribution of the stations, magnitude, and cardinality of the results indicate the potential to use QC for clustering for a real-world application, e.g., BSS.

4.3 Introduction

In 1965, the first generation of BSS was introduced in Amsterdam. The BSS offered ordinary bicycles painted white for public utilization, allowing individuals to locate a bike, ride it to their desired destination, and then leave it for the next user [150]. Since then, BSS has witnessed remarkable expansion, becoming an integral component of transportation networks in major metropolitan areas. BSS empowers users to access bicycles from stations located at points of interest (referred to as origin stations) and subsequently return them to the same or different points of interest (termed destination stations). BSS is regarded as the cornerstone of sustainable transportation systems. While the specific objectives of these systems may have varied during their early adoption, initially

catering to leisure and exercise needs, their overarching impact extends to influencing micromobility and multi-modal transit systems at a broader level. In order to enhance the experience of users within BSS, tracking the state of the entire BSS network is crucial. Identifying stations with similar behavior and status concerning location and bike availability provides decision-makers with valuable insights to enhance service quality. Employing clustering algorithms offers a solution to this challenge by grouping stations exhibiting comparable characteristics into the same cluster. This approach enables decision-makers to gain a comprehensive understanding of station dynamics and facilitates targeted improvements to optimize user experience.

The clustering of BSS stations is important for several reasons such as Optimizing User Accessibility, Efficient Resource Management, Enhanced User Experience, Data-Driven Decision Making, Sustainability, and Urban Planning.

The computational complexity of clustering in real-time for BSS presents a multifaceted challenge due to the dynamic nature of system data and the need for timely decision-making. BSS stations generate vast streams of data regarding user demand, station availability, and environmental factors, requiring continuous analysis to optimize system performance. Traditional clustering algorithms often struggle to cope with the high-dimensional and evolving nature of BSS data, leading to inefficiencies in resource allocation and user experience [151]. Moreover, the real-time nature of BSS operations imposes computational constraints, necessitating algorithms capable of rapid processing and adaptation to changing conditions. Addressing the computational complexity of real-time clustering in BSS demands the development of innovative algorithms tailored to the specific requirements of urban mobility systems, leveraging advancements in computational techniques, such as QC, to unlock new areas for efficient and scalable clustering solutions for BSS.

Although the primary objective of BSS may change early on, depending on whether it is leisure or exercise-oriented, from a high-level perspective, it can have an impact on micro-mobility and transportation management [152]. It can bridge gaps and connect missing links in public transportation networks [124]. Additionally, it avoids high costs for personal vehicles, as well as lowering carbon emissions and reducing traffic congestion [153]. Despite the BSS's advantages, managing and

running the system effectively can be challenging [154]. To manage these systems efficiently, it is necessary to use clustering to find similar stations that have similar characteristics that are changing dynamically in real-time [155].

This paper investigates the potentials of QC, and more specifically QA, for clustering real-world data. The BSS is used as a proof of concept for executing a clustering model on QA computers. The goal of the relatively new practical branch of computer science known as QC is to create computer technologies and algorithms by applying the concepts of quantum physics. Two physicists, Richard Feynman [156] and Yuri Manin [157], devised the first theoretical concepts regarding QC in the 1980s, but it was not until the 1990s that the first actual QC demonstrations were carried out [158, 159]. Quantum computers with a few quantum bits (qubits) were created in the early 2000s, and in the 2010s, work on quantum computers with over 100 qubits started [160, 161]. Although quantum computers are still in their infancy, they already show enormous potential for overcoming a variety of issues that are now insurmountable for classical computers [139].

QML is a field that combines the principles of QC and ML [162]. QCs are able to perform certain types of calculations much faster than classical computers by taking advantage of the properties of quantum physics, such as superposition and entanglement [163]. Computers can be taught to learn from data without being explicitly programmed using ML by analyzing datasets using algorithms to define a decision-making system based on patterns in the data. By using QC for ML tasks, it may be possible to process and analyze large amounts of data more efficiently and quickly, leading to new and improved ML techniques. To fully realize the potential of this technology, more research is necessary, as the field of QML is still in its early stages. Since QC technologies and devices continue to develop quickly, it is important to know what kinds of uses can take advantage of their power. On the other hand, ML on conventional computers has made great strides, displaying excellent results in real-world transportation applications, with increased computational power resulting in consistently better performance.

To cluster stations in BSS considering their real-time characteristics and geospatial components identifies as a combinatorial optimization problem. The k-means algorithm is a well-known clus-

tering approach for such systems. However, K-means is an NP-hard clustering algorithm that cannot be solved as a deterministic approach. In other words, in practical applications, the earlier methods are still unable to find deterministic solutions. This issue has been investigated using a variety of heuristic and meta-heuristic techniques in classical models. QC and, more specifically, QA computation offer promising short-term meta-heuristic solutions to combinatorial optimization issues.

The main contribution of this research is to introduce a hybrid model to cluster stations in a BSS by solving it as a CSP problem with different methods on a QA computer using real-time dataset. In addition to the practical contribution, this research also offers theoretical advancements in the field of computational optimization by defining a new topology for the input data that is compatible with QC topology (Chimera topology). The workflow of converting classical feature vectors of stations into a new graph is also described.

This research tries to answer three main questions; 1) What are the potential usages of QC, specifically QA, for real-time clustering in BSS? 2) How can we formulate algorithms suitable for execution on quantum computers to solve clustering algorithms with geospatial features? 3) What are the differences in solving the real-time clustering problem on quantum computers compared to classical computers?

The rest of the paper is structured as follows. The related works of various publications regarding the QML and transportation are presented in Section 4.4. Section 4.5 discusses the proposed methodology adopted for clustering stations in the BSS by using a hybrid QC approach. Section 4.6 describes the results and discussion, and section 4.7 concludes the paper and provides future works.

4.4 Related works

Figure 4.1 illustrates the trend in the number of publications related to BSS from 1996 to 2023, derived from the Web of Science dataset (WOS). The query term used for data extraction was

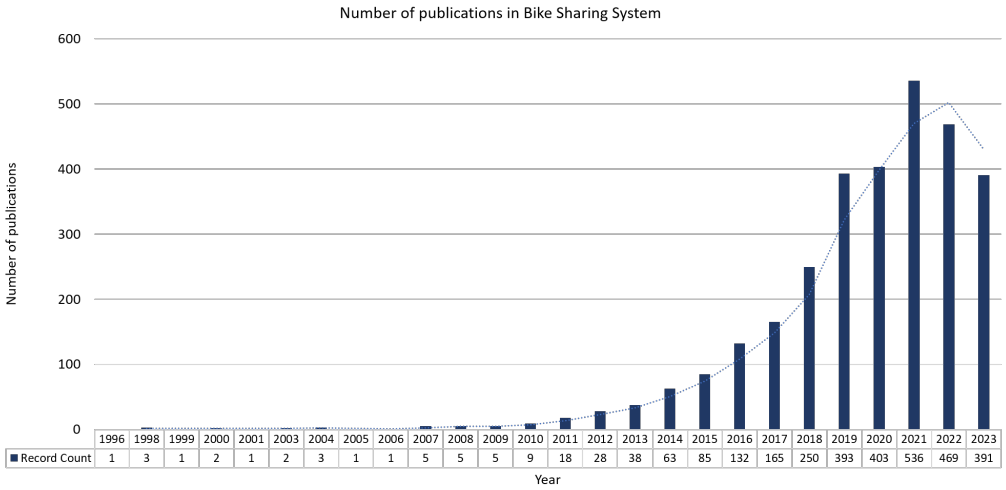


Figure 4.1: The number of publications from 1996 to 2023 shows the BSS trend. The data is derived from the web of Science dataset (WOS). Term for query: (Bike Sharing System OR Bike Sharing) search through the title, abstract, and keywords of the papers.

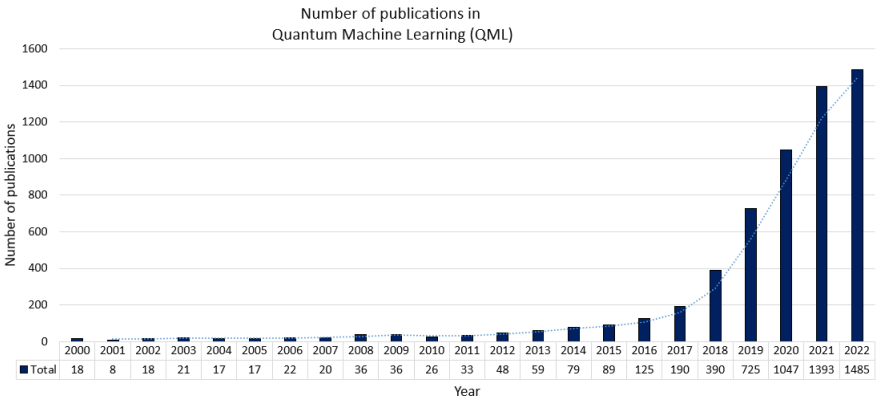


Figure 4.2: The number of publications from 2000 to 2022 shows the QC trend in ML. The data is derived from the Web of Science dataset (WOS). Term for query: (Quantum Machine Learning) search through the title, abstract, and keywords of the papers.

"(Bike Sharing System OR Bike Sharing)", searching through the titles, abstracts, and keywords of papers. As shown in Figure 4.1, the number of publications on BSS has been increasing in recent years. Many researchers have been working on different aspects of BSS.

Jingjing Yang et al. [164] proposed a method to enhance prediction accuracy in BSS. They utilized network representation learning and hierarchical prediction techniques to address the challenge of uneven station usage. Their method involved clustering of stations in BSS based on migration trends and geographic data, followed by hierarchical prediction of system activity. Through evalu-

ation against baseline methods, they demonstrated significant improvements in prediction accuracy, indicating the effectiveness of their approach in addressing the challenges of BSS.

Wang et al. [165] proposed a method to enhance the accuracy of predicting bike usage in BSS. They employed a two-level fuzzy c-means clustering algorithm, integrating geographic location data and bike migration trends between stations. This approach aimed to cluster shared bike stations more effectively and improve clustering accuracy.

Zhao Tong et al. [166] proposed a method to analyze the spatio-temporal patterns of dockless shared bikes in Wuhan and identify factors influencing these patterns. They utilized a dynamic time-warping distance-based two-dimensional clustering method to quantify these patterns and applied a multiclass explainable boosting machine to explore their main related factors. The study concluded that spatio-temporal patterns could help identify associated problems such as imbalance or lack of users and guide planners in prioritizing strategies to enhance the usage and efficiency of BSS.

Shangaranarayanee et al. [167] presented a review of BSS publications that use different methods for analyzing time series datasets. They developed a model to address traffic congestion in urban areas by leveraging shared transportation modes. They highlighted algorithms such as the K-Means and Fuzzy C-means as effective tools for visualizing resulting clusters and enhancing forecasting accuracy in BSS.

Marta Galvani et al. [168] proposed a novel clustering method for BSS, focusing specifically on the Vélo'v BSS in Lyon, France. They analyzed a dataset comprising loading profiles of 345 bike stations over one week, treating the data as continuous functional observations throughout the day. Their primary objective was to uncover spatio-temporal usage patterns of BSS, thereby identifying groups of stations and days exhibiting similar behavior. To achieve this goal, a novel bi-clustering algorithm is proposed for functional data analysis, extending existing non-parametric algorithms designed for multivariate datasets.

Feng et al. [91] addressed the issue of imbalanced bicycle usage patterns in BSS. Recognizing the importance of bike demand prediction for effective bike pre-allocation, they proposed a hierarchi-

cal traffic prediction model aimed at forecasting the check-out/in numbers of each station cluster. The approach begins with station clustering using an iterative spectral clustering algorithm, capitalizing on the regularity of bike usage patterns among closely located stations. Additionally, they introduced an inter-cluster transition proportion model to characterize bike rent-return relationships between pairs of clusters and predict check-in numbers for these clusters.

Many researchers have been working on different aspects of QML. The number of publications from 2000 to 2022 shows the QC trend in ML is presented in figure 4.2. The data is derived from the WOS. The keywords for query in WOS are considered as "Quantum Machine Learning" search through the title, abstract, and keywords of the papers.

Indeed, QC models can be broken down into four main groups, each of which has its own way of using quantum physics to do calculations: A) circuit-based or gate-based models [117, 160], which are based on the idea of quantum circuits. In this model, quantum computers are composed of quantum gates, which are special types of operations that can be performed on qubits. Quantum gates can manipulate the states of qubits, allowing quantum computers to perform calculations. B) measurement-based or one-way, are models in which superpositions and entanglements are based on the idea of measurements [80]. In this model, quantum computers are made up of a special type of quantum state called a cluster state. A cluster state is a type of quantum state that can be created by performing measurements on a group of qubits. Once the cluster state has been created, it can perform various QC tasks by making additional measurements on the qubits. C) Topological models, use non-Abelian forms of matter to store quantum information and implement a more durable qubit [169]. In such a scheme, the quasi-particles themselves do not contain the information; rather, it is contained in the interactions and braiding of the quasi-particles. Qubits are initialized as non-abelian anyons in topological QC, which have their own antiparticles. D) adiabatic models, which are based on adiabatic evolution theory. According to this theory, quantum computers are made to find the state of qubits gradually over time while traveling down a specific path that solves a specific problem. Adiabatic quantum computers can find the ground state of a system by slowly cooling it down [118]. In other words, QA is a subcategory of adiabatic QC which

is a promising near-term meta-heuristic approach to solving combinatorial optimization problems [170].

Quantum algorithms in QML are defined in a different way to deal with typical problems in classical ML by using the efficient characteristics of QC. Running an algorithm can be an extended version of a classical algorithm or a completely reformulated version of the algorithm. Based on the current available QC devices, in most cases, hybrid QC is used when it comes to real-world applications [171]. In other words, working on solely QC devices would result in wrong results due to the noisy nature of the current devices [172]. Therefore, it is crucial to explore and analyze the potential of hybrid approaches to correctly take advantage of QC devices. The algorithms that are formulated to be run on a hybrid QC model, are called Quantum-Classical variational algorithms (QCVA) [173]. The recent development of QCVA was caused by the implementation of the VQE [174], which led to an enormous amount of research related to the near-term algorithms.

Theoretically speaking, QML can be defined as models of quantum information that can be trained on a set of inputs and recognize the patterns in the training data [175]. Mohammad H. Amin et al.[176] proposed a novel ML method based on the Quantum Boltzmann Distribution (QBD). Training in Quantum Boltzmann Machines (QBM) may become challenging due to quantum mechanics' non-commutative nature. They tackled this issue by using bounds on the quantum probabilities. Therefore, they effectively trained the model through sampling. They used exact diagonalization, to demonstrate trials of QBM training with and without the bound and contrast the outcomes with those of traditional Boltzmann training. They also covered the use of QA processors for QBM training. Nathan Wiebe and M'aria Kieferov'a [177] proposed a cutting-edge method for training QBM, a type of Recurrent Quantum Neural Network (RQNN). They generalized the previous approaches, defined new methods to train RQNN, and compared the results with the state of the art. Furthermore, their findings demonstrated that QBMs generate a type of quantum state tomography that not only estimates a state but also provides instructions for producing copies of the reconstructed state, which previous machines are not able to complete. Patrick Rebentrost et al. [178] have shown that a Support Vector Machine (SVM), which is an optimized binary classi-

fier, can be used on QC devices. The complexity of SVM depends on the size of the vectors and training examples in a logarithmic way. Whereas polynomial time is needed for classical sampling algorithms, they achieved exponential speedup.

Seth Lloyd et al.[179] explored the potential and usage of Quantum Principal Component Analysis (QPCA) in QC. They demonstrated how a quantum system with a density matrix (ρ) can be duplicated to build the unitary transformation $e^{-i\rho t}$. In their illustrations, they showed how QPCA can be crucial to a range of quantum algorithms and measurement applications. They gave a quantum cluster assignment example that demonstrates how QPCA could help accelerate ML issues like clustering and pattern recognition. A novel approach for ML from the perspective of QC was proposed by Vedran Dunjko et al.[180]. The three main branches of ML, supervised, unsupervised, and reinforcement learning are all covered in their research. However, since many researchers have reported quantum improvements in supervised and unsupervised learning, they have mainly concentrated on Quantum Reinforcement Learning (QRL). They demonstrated that for a large class of learning issues, quadratic improvements in learning efficiency and exponential improvements in performance over short periods can be obtained.

While existing literature on QML has made significant strides in addressing various ML tasks, a notable gap exists regarding the applicability of these approaches to real-time scenarios, particularly in the context of dynamic systems in BSS. The majority of published works in QML have primarily focused on solving problems using static data, employing post-processing approaches that may not apply to real-time applications. However, the dynamic nature of BSS demands adaptive solutions capable of managing stations in real-time to ensure efficient resource allocation and user satisfaction. Recognizing this gap, the proposed approach aims to solve this issue by introducing a novel methodology tailored specifically to address the real-time clustering requirements of BSS.

4.5 Methodology

This section describes how clustering stations in BSS is formulated as a QUBO. The various methods for calculating the dissimilarity between the items in each cluster and the clusters themselves are then explained. The evaluation method is then defined.

4.5.1 Quantum Annealing (QA)

In order to formulate the model, we need to first target one of the QC models. There is a strong incentive to analyze and discuss current NP-hard problems that are computationally difficult even in the most powerful classical computers while the technology of the quantum hardware is scaling up [181].

Different QC models are based on entirely different technology. As a result, each model has a different limited number of qubits as well as noises that seem to be brought on by a large number of entangled qubits. Among the current QC models, QA has demonstrated a more useful method for resolving NP-hard problems. That is why QA has been used in this research.

The Hamiltonian function for a QC device is a function that links particular states, referred to as eigenstates, to energies [182]. The energy of the system is only well defined and referred to as the eigenenergy when it is in an eigenstate of the Hamiltonian. All the other states of the system make their energies uncertain. The eigenspectrum is the set of eigenstates with known eigenenergies [Eq .4.1].

$$H_{Ising}(s) = -\frac{A(s)}{2}\sum_i \hat{\sigma}_x{}^{(i)} + \frac{B(s)}{2}\left(\sum_i h_i \hat{\sigma}_x{}^{(i)} + \sum_{i>j} J_{i,j}\hat{\sigma}_z{}^{(i)}\hat{\sigma}_z{}^{(j)}\right) \tag{4.1}$$

where $J_{i,j}$ is coupling strengths between qubits i and j, and, h_i is biased for qubit i, and $\hat{\sigma(i)}$ is Pauli matrices. There are two terms in the formula 1. The initial Hamiltonian is the first term, and the final Hamiltonian is the second. The first term, which occurs when all qubits are in a superposition state, is the lowest-energy state of the states (0, 1). The tunnelling Hamiltonian is another name

for this term. The second term is the lowest-energy state of the final states and is the answer to the problem that needs to be solved. The σ (qubit biases) and the couplings between qubits are present in this classical state. The term is also known as problem Hamiltonian.

$$f(x) = \sum_i Q_{i,i} x_i + \sum_{i<j} Q_{i,j} x_i x_j \tag{4.2}$$

where $Q_{i,i}$ = the linear coefficients

$Q_{i,j}$ = the quadratic coefficients or

non-zero off-diagonal terms

$$E_{Ising}(s) = \sum_{i=1} h_i s_i + \sum_{i=1} \sum_{j=i+1} J_{i,j} s_i s_j \tag{4.3}$$

where h_i = the linear coefficients

$J_{i,j}$ = the quadratic coefficients or

non-zero off-diagonal terms

To run the model on D-Wave QA computers, the model needs to be reformulated as either the Constrained Quadratic Model (CQM), Binary Quadratic Model (BQM), or Discrete Quadratic Model (DQM). In this research, we proposed a hybrid approach that works with BQM as the variables are binary, and a constraint that defines each qubit can only be part of one cluster. The model can be executed as a QUBO (Eq. 4.2) or Ising (Eq. 4.3). An Ising model, to be more precise, serves as the quantum hardware's input. However, because of the way QUBO works (its parameters are 1 and 0 as opposed to the Ising model's -1 and +1), it is simpler to formulate problems using QUBO than Ising. However, these two models can be converted into one another. To map to either the chimera graph or the pegasus graph, the model should be first formulated to QUBO and the Ising model will be passed to the Quantum Processing Unit (QPU) in D-Wave.

To define biases and couplers in Equation 4.2, we need to define our problem properly. First, each station in our dataset can only be part of one cluster. If C clusters are available, for each station, C qubits should be initialized. If station i belongs to cluster m then $q_m = 1$, and for $n = 1, ..., C - mq_n = 0$. In other words, the sum of all possibilities for one station equals to 1 (Eq. 4.4).

$$\sum_{i=1}^{C} x_i = 1 \tag{4.4}$$

where C = number of clusters,

x_i = qubit.

The line in Eq 4.4 needs to be formulated as a quadratic model to be able to use it in QUBO (Eq. 4.5).

$$\sum_{s \in S} (\sum_{i=1}^{C} x_i - 1)^2 \tag{4.5}$$

where S = Number of stations.

Another crucial point is that similar stations ought to be in the same cluster. Thus, according to Eq 4.6 (for two stations), $g(x)$ is a similarity function between two points. In this research, we have investigated four different functions.

$$\sum_{i} (\sum_{j} g(x) x_i x_j) \tag{4.6}$$

where $g(x)$ = a similarity distance between two points.

Finally, we can define the objective function as Eq 4.7.

$$min(\sum_{i=1}^{C} (\sum_{j} g(x) x_i x_j) + \gamma \sum_{s \in S} (\sum_{i=1}^{C} x_i - 1)^2) \tag{4.7}$$

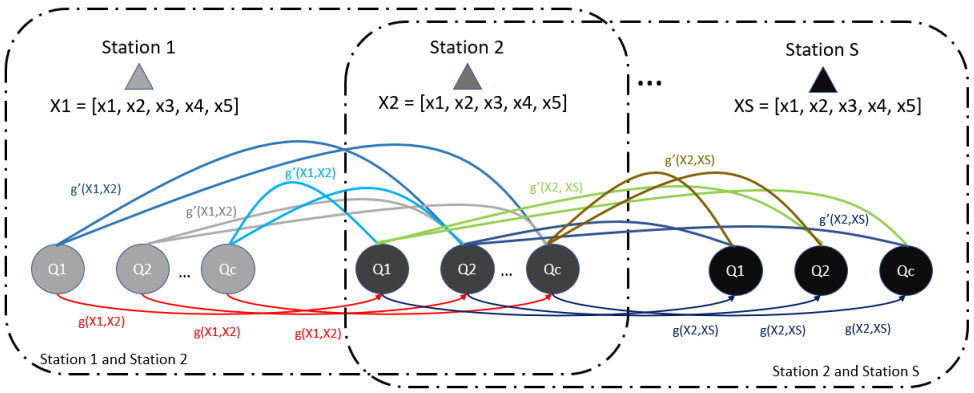


Figure 4.3: The overview of Constraint Satisfaction Problem for S stations and C clusters. For each station C qubits are needed.

4.5.2 Constraint Satisfaction Problem (CSP)

The $g(x)$ function can be considered as weights for the edge of a graph in which the nodes are the stations. The BQM can be constructed from a CSP by using the D-wave Binary CSP library [183]. It uses a model called the penalty model, in which each constraint in the CSP is mapped to an Ising model or QUBO. Therefore, for. For each pair of nodes (stations) in the graph, g(x) is defined based on the similarity of the nodes (Eq 4.8, Figure 4.3).

$$g(X_i, X_j) = \begin{cases} -cos(d\pi) & \text{if } i = j \\ -tanh(d) * 0.5 & \text{otherwise} \end{cases} \tag{4.8}$$

The BQM that is represented in Figure 4.3, needs to be mapped into a real QC device (in our case, into a Quantum Processing Unit (QPU)). Minor embedding is the conversion of problem formulation variables to qubits on the QPU. Objective functions can be represented graphically in minor embedding, allowing you to map the objective function to the QPU. In this case, physical qubits will be initialized based on the nodes that are variables in the model with a linear coefficient. Moreover, the edges are the quadratic coefficient of the model that will be mapped to couplers between qubits (Figure 4.4).

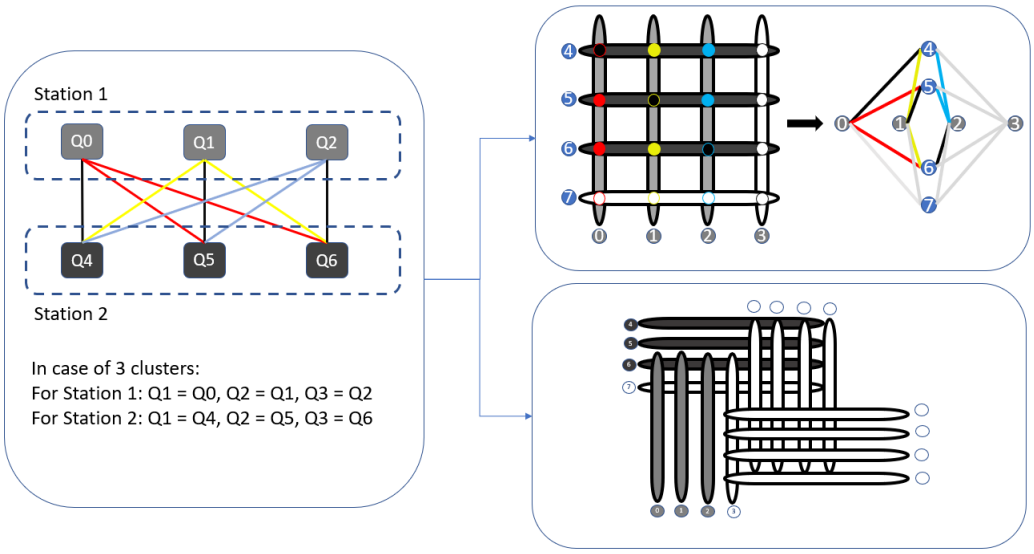


Figure 4.4: The overview of mapping from theory into real physical QC devices. Two different topologies that are being used by D-wave devices is visualized. On the top-right, the Chimera topology is depicted. On the bottom right, the Pegasus topology is depicted.

4.5.3 Geospatial Analysis

BSS in Toronto is used to run our model as a proof of concept. To create a feature class for each station, five features have been selected; 1) latitude, 2) longitude, 3) capacity, and 4) available bikes 5) population. A real-time application programming interface (API) is used to obtain the status of capacity and available bikes, which is a dynamic dataset. In order to find the population around each station, first the population of Toronto is classified based on different neighbourhoods and they are geocoded by address, and a geodatabase is populated that contains a polygon for each area. Then a 500 meter buffer is generated around each station. Finally, the intersection of the buffers and polygons is created. The results are stored as attributes for each station (point dataset). The map is available through this link.

4.5.4 Defining Number of Clusters

There are different approaches to determining the number of clusters. In this paper, the three most popular methods have been used to find the best number for clustering bike stations. In this section,

these three methods are described.

Elbow method

The elbow method determines the optimal number of clusters for a given dataset. It operates by plotting the variation explained as a function of the number of clusters and selecting the elbow of the curve as the number of clusters to use [184].

$$min \sum_{k=1}^{k} W(c_k) \tag{4.9}$$

where k = is the number of clusters,

c_k = is the k^{th} cluster, and

$W(c_k)$ = is the variation of cluster number k.

The total WSS should be minimized. Different values should be tested for k. For each k, the WSS should be calculated. Then the results for all k values will be visualized, and the knee (bend) of the plot will be an appropriate number of clusters.

Silhouette

The silhouette is another well-known technique for determining the optimal number of clusters and their interpretation, as well as for assessing the consistency of data clusters. The silhouette approach calculates silhouette coefficients for each point, which measure how similar a point is to other points in its own cluster relative to points in other clusters [185]. In other words, it relies on cohesion and separation for each station of clusters. The range of the score is [1, -1], where a higher value means that the station is well classified within its own cluster and poorly matched to other clusters around it.

$$s(i) = \frac{b(i) - a(i)}{max(b(i), a(i))} \tag{4.10}$$

where $a(i)$ = is the average distance of point i^{th} with
other points in the same clusters,
$b(i)$ = is the average distance of the point with
other points in the closest cluster, and
$s(i)$ = is silhouette score.

Silhouette Analysis

Similar to the previous method, silhouette analysis uses the score for each station as a visual measure of how a clustering algorithm has performed. Once the silhouette score is calculated for each station in the dataset, it plots the results to get a visual representation of how accurately the stations are clustered into k clusters. It creates a way to visually evaluate parameters like the number of clusters.

4.5.5 Dissimilarity Measurement

The dissimilarity function indicates how dissimilar the dataset objects are. In addition, these terms are frequently employed in clustering, where similar data samples are grouped into a cluster. All other data samples are categorized separately. There are numerous types of dissimilarity functions for clustering. In this paper, four well-known functions are investigated.

Euclidean Distance

The Euclidean distance, or Norm L2 (Equation 4.11) is one of the main dissimilarity functions that can be used for numeric features. The result of the Euclidean function is the smallest distance between items in the dataset [186]. This is also can be considered the shortest path between two items. This function works well in 2-dimensional Cartesian coordinate systems. However, if the items are far from each other, the Euclidean function is not useful because it does not consider the curvature of the earth.

$$d(P,Q) = \|P - Q\| = \sqrt{\sum_{i=1}^{n}(p_i - q_i)^2} \tag{4.11}$$

where $P = (p_1, p_2, ..., p_n)$ and

$Q = (q_1, q_2, ..., q_n)$

Manhattan Distance

The Manhattan function is also called Norm L1 and City Block. This function finds the distance between two pairs of items when there is an obstacle between the items [187]. In a road network, for instance, the distance from intersection A to intersection B cannot be calculated directly by a straight line because there are many obstacles between A and B, such as buildings. The equation 4.12 defines the Manhattan formula for n-dimensional space.

$$d(P,Q) = \|P - Q\| = \sum_{i=1}^{n}(p_i - q_i)^2 \tag{4.12}$$

where $P = (p_1, p_2, ..., p_n)$ and

$Q = (q_1, q_2, ..., q_n)$

Pearson Correlation

Pearson correlation is a correlation-based distance that calculates the strength of the linear correlation between two items in the feature vector [188]. In addition, the covariance value is used as the initial step in the computation. However, covariance is difficult to interpret and does not indicate how close or far the data are from the line representing the trend between measurements. The equation 4.13 shows the correlation distance formula.

$$CorrelationDistance = 1 - \frac{Correlation(P,Q)}{\sqrt{Variance(P)}\sqrt{Variance(Q)}} \tag{4.13}$$

Spearman Correlation

Similar to Pearson correlation, Spearman correlation is used whenever we are dealing with bivariate analysis. It is applicable to categorical and numeric attributes. It is applicable to categorical and numeric attributes. The Spearman correlation formula is defined by the expression 4.14.

$$\rho = 1 - \frac{6\sum_{i=1}^{n}(r(p_i) - r(q_i))^2}{n(n^2-1)} \tag{4.14}$$

where $r(p_i)$ = rank of p_i

$r(q_i)$ = rank of q_i

n = number of pairs

4.5.6 Dataset

Table 4.1 provides an overview of the total number of regions, stations, and trips within the BSS network in Toronto, Canada. Additionally, Table 4.2 presents metadata related to BSS stations, while Table 4.3 contains metadata concerning individual trips. The population density of each region is calculated using data from the 2016 census. Figure 4.5 visualizes the distribution of BSS's stations in Toronto for 2021. The latitude and longitude of the stations are available as a static dataset as a shapefile in the City of Toronto dataset portal. An application programming interface (API) in the public bike system provides real-time information about capacity and available bikes for each station. The data format is a JSON array. Toronto's population is available in the Census dataset. The format of the data is Shapefile. ArcGIS Pro is utilized to aggregate these datasets, resulting in the creation of a feature vector for each station, comprising latitude, longitude, capacity, available bikes, and population.

Table 4.1: Total number of records in regions, stations, and trips

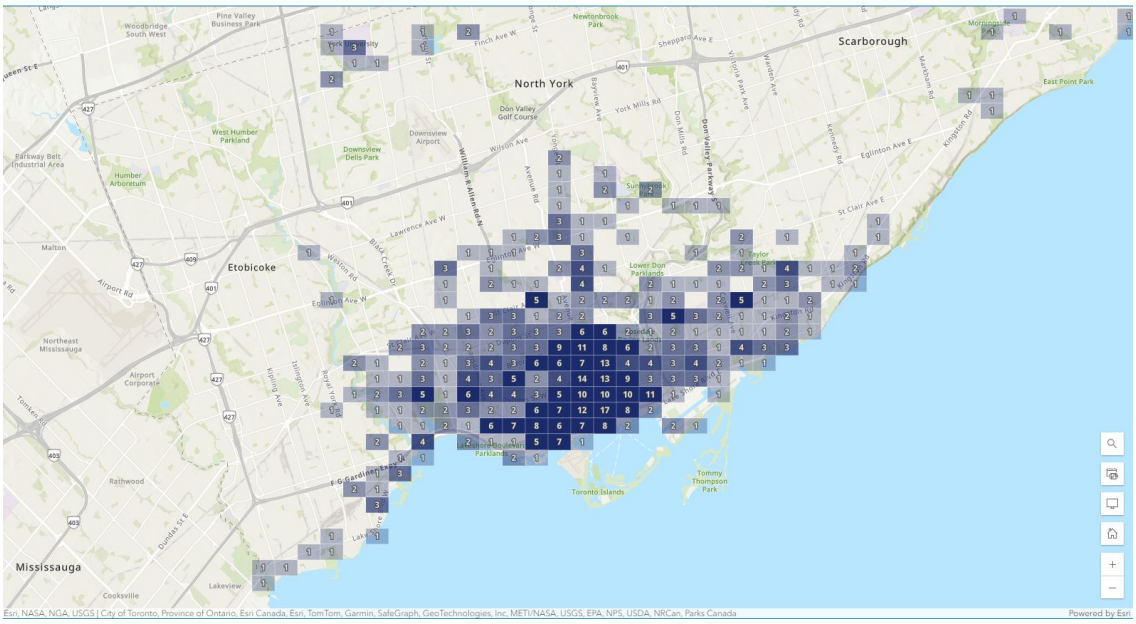
Category	Value
Total number of regions in Toronto	141
Total number of stations	625
Total number of trips	950969

Table 4.2: Fields in the meta data of the station information

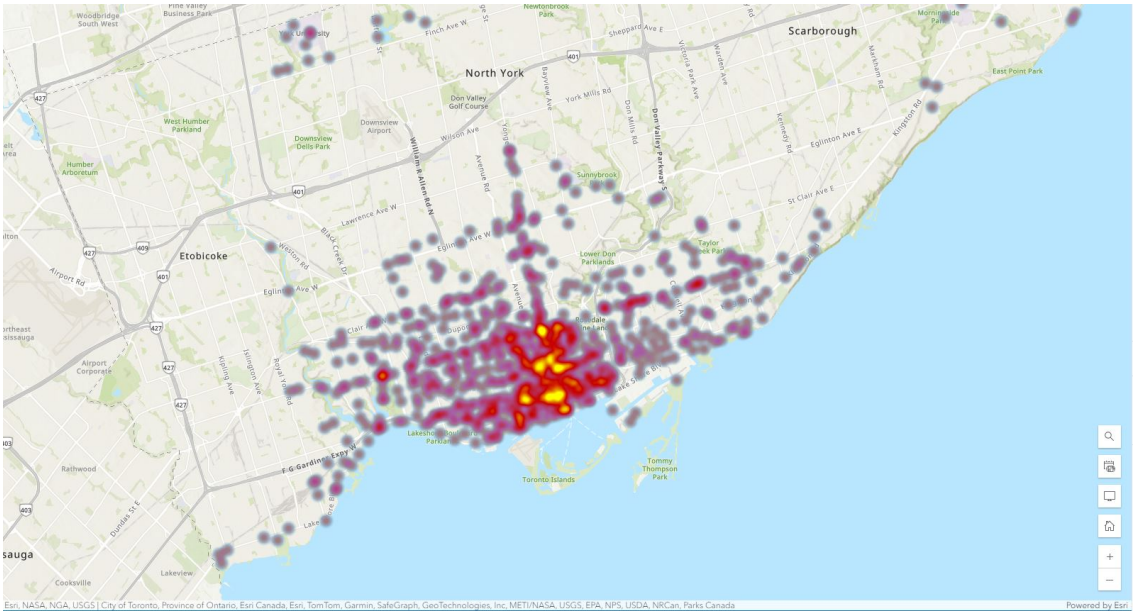
Field Name	**Description**
station_id	ID of the station
name	Name of the station
physical_configuration	Regular—Electrical
lat	Latitude
lon	Longitude
altitude	Altitude
address	Address of the station
capacity	Capacity of the station
is_charging_station	Indicator if it's a charging station
rental_methods/0	Key
rental_methods/1	Transit Card
rental_methods/2	Credit Card
rental_methods/3	Phone
obcn	On-board Communication Number
nearby_distance	Distance to nearby locations
_ride_code_support	True or False
post_code	Postal code

Table 4.3: Fields in the meta data of the trips

Field Name	**Description**
Trip Id	Unique identifier for the trip
Trip Duration	Duration of the trip
Start Station Id	ID of the starting station
Start Time	Time when the trip started
Start Station Name	Name of the starting station
End Station Id	ID of the ending station
End Time	Time when the trip ended
End Station Name	Name of the ending station
Bike Id	ID of the bike used for the trip
User Type	Type of user (Annual, Casual)

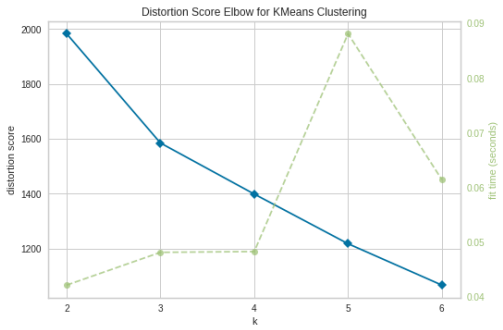


(a)

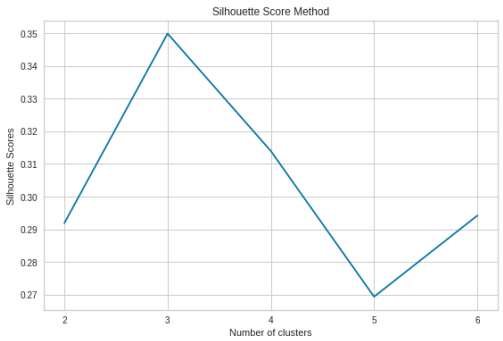


(b)

Figure 4.5: (a) Distribution of the BSS's stations in Toronto based on an aggregated index for each region. (b) Heatmap of the BSS's stations in Toronto.



(a) Elbow method results for clustering, which indicates that the best number of clusters for the given data is 3 (elbow point).



(b) The result of the silhouette method. The maximum score in the silhouette shows the best number of clusters, which is 3.

Figure 4.6: (a) Elbow method results for clustering, which indicates that the best number of clusters for the given data is 3 (elbow point). (b) The result of the silhouette method. The maximum score in the silhouette shows the best number of clusters, which is 3.

4.6 Results and Discussion

4.6.1 Defining Number of Clusters

The results of the elbow method for clustering provide valuable information about the structure of the data and can be used to make informed decisions about the number of clusters. Figure 4.6a shows the results of the elbow method for clustering, which indicates that the best number of clusters for the given data is 3. This means that when the data is divided into 3 clusters, the variance within each cluster is minimized, and the clustering solution is optimal. The elbow method was applied to the data, and the range for the distortion score was found to be between 1050 and 2000. The range for the number of clusters, k, was between 2 and 6. The fit time for the clustering algorithm was between 0.04 and 0.09 seconds.

However, while the elbow method gives us a useful overview to make a decision about the number of clusters, it does not take into account the presence of outliers or the shape of the clusters. Thus, it is better to analyze other approaches as well to make the best decision for the number of clusters. Figure 4.6b depicts the result of the silhouette method. The vertical axis shows the silhouette score, and the horizontal axis represents the number of clusters, for which in this case we checked the results for 2 to 6 clusters. The maximum score in the silhouette shows the best number of clusters,

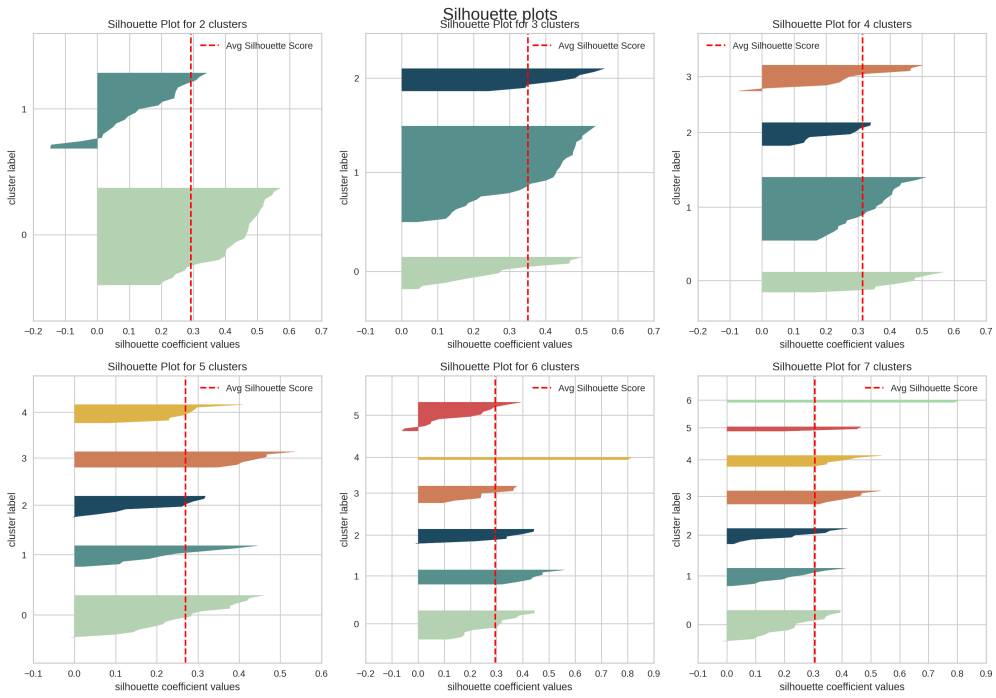


Figure 4.7: The results of the silhouette analysis method for six different numbers of clusters.

which is 3, the same as the elbow method.

Figure 4.7 shows the results of the silhouette analysis method for six different numbers of clusters. The vertical axis is the cluster label, and the horizontal axis is the coefficient values. These sub-figures in Figure 4.7 depict an index of how close each station in one cluster is to stations in the other close clusters. This index is the average of the silhouette scores. The index has a range of [-1, 1], and the best number of clusters is the one with the maximum index, which in this case is the top-middle sub-figure where the index equals 0.35.

4.6.2 Distribution Of Stations In Clusters

Figure 4.8 includes five sub-figures, each of which is related to one feature of the feature vector that has been used for clustering. Each sub-figure represents a box plot of the three clusters, represented by a bar. Each bar represents the distribution of feature values for a particular cluster. The box of the bar represents the Inter Quartile Range (IQR), which is the range between the lower quartile and the upper quartile. The lower and upper hinges of the box correspond to the first and third quartiles,

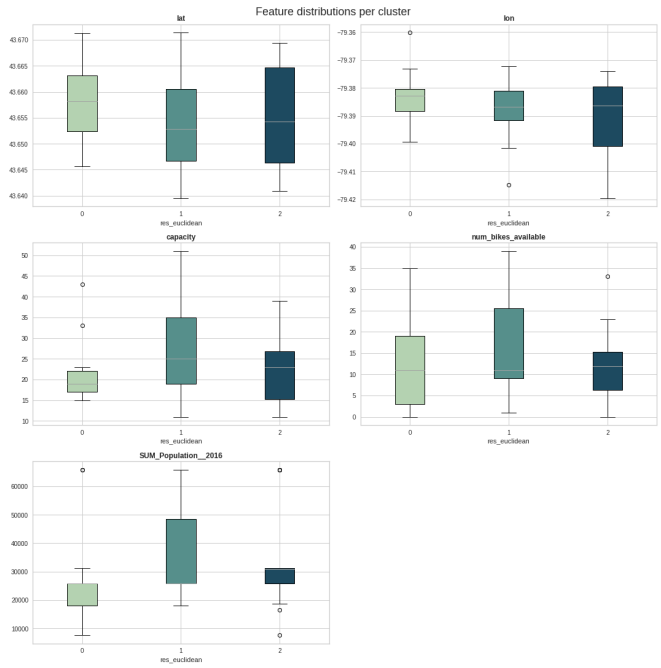


Figure 4.8: Results of box bar for Euclidean method that includes five sub-figures, each of which is related to one feature of the feature vector that has been used for clustering.

respectively. The line inside the box represents the median of the distribution. The whiskers of the bar extend from the hinges to the minimum and maximum values of the distribution. Any points outside the whiskers are considered outliers.

The horizontal axis in all sub-figures is the same and represents the labels of each bar or cluster. The vertical axis, however, is different and shows the values of a feature. For instance, the vertical axis in the top-left sub-figure represents latitude, with values ranging from 43.646 to 43.673. For the first cluster in the top-left sub-figure, the latitude values range from 43.646 to 43.673, with a median value of 43.660. The lower quartile is 43.651, and the upper quartile is 43.667. The IQR is 0.016. For the second cluster, the latitude values range from 43.639 to 43.668, with a median value of 43.654. The lower quartile is 43.649, and the upper quartile is 43.659. The IQR is 0.010. For the third cluster, the latitude values range from 43.641 to 43.669, with a median value of 43.6525. The lower quartile is 43.6452, and the upper quartile is 43.6625. The IQR is 0.018. The results show that the latitude values for all three clusters are relatively close, with a range of only 0.028. There is a small difference between the median values of each cluster, but overall the distributions

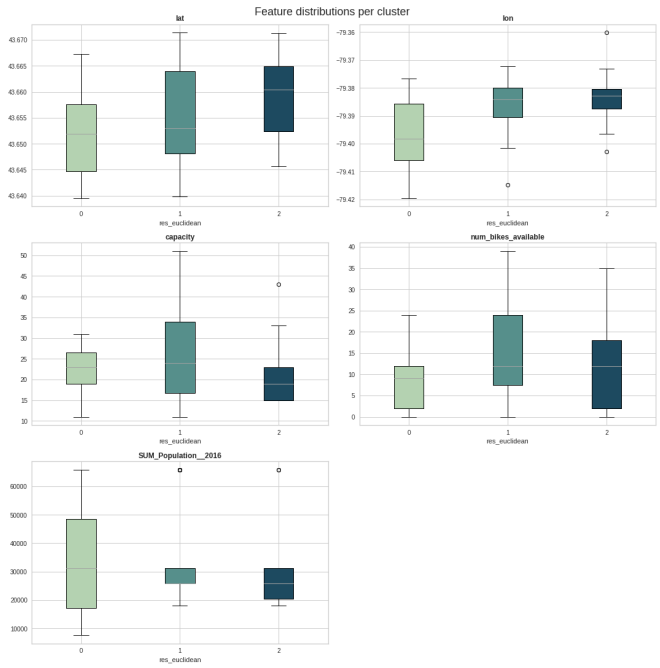


Figure 4.9: Results of box bar for Manhattan method that includes five sub-figures, each of which is related to one feature of the feature vector that has been used for clustering.

are similar. No outliers were detected in any of the clusters, indicating that the latitude values are relatively similar across the clusters.

By interpreting all box bar figures for Euclidean, Manhattan, Pearson, and Spearman, it is visualized that there are a couple of outliers in Figures 4.8-4.11 for longitude, capacity, number of available bikes, and population. A box bar with outliers in two of the bars suggests that the distributions of the data for those two clusters have values that are significantly different from the majority of the data. Outliers are defined as values that fall outside the range defined by the lower and upper hinges of the box plot. They are plotted as individual points and are often indicative of unusual or extreme observations in the data.

The results of the box chart show that the best approach for clustering the stations is Spearman, which has the best distribution of the data in the clusters and the lowest outliers.

It is worth mentioning that in some cases, outliers can be considered important and deserving of further investigation. In other cases, they may be considered errors or anomalies that should be excluded from further analysis, which is outside the scope of this research.

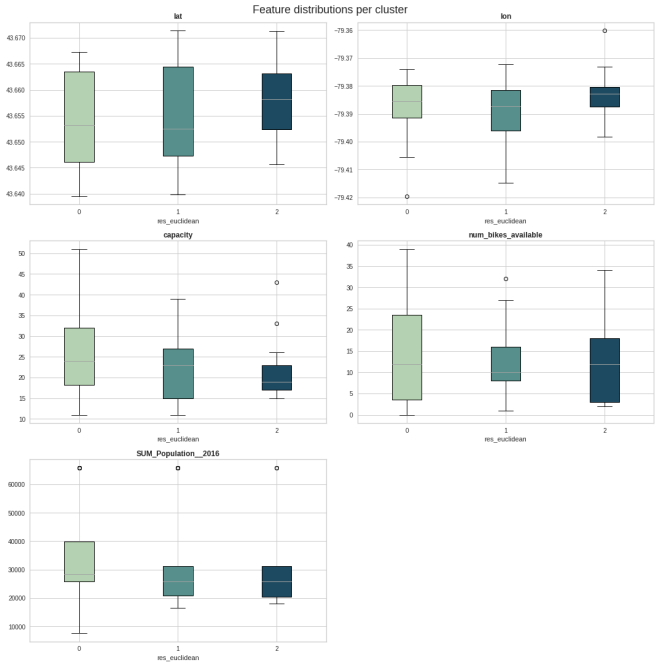


Figure 4.10: Results of box bar for Pearson method that includes five sub-figures, each of which is related to one feature of the feature vector that has been used for clustering.

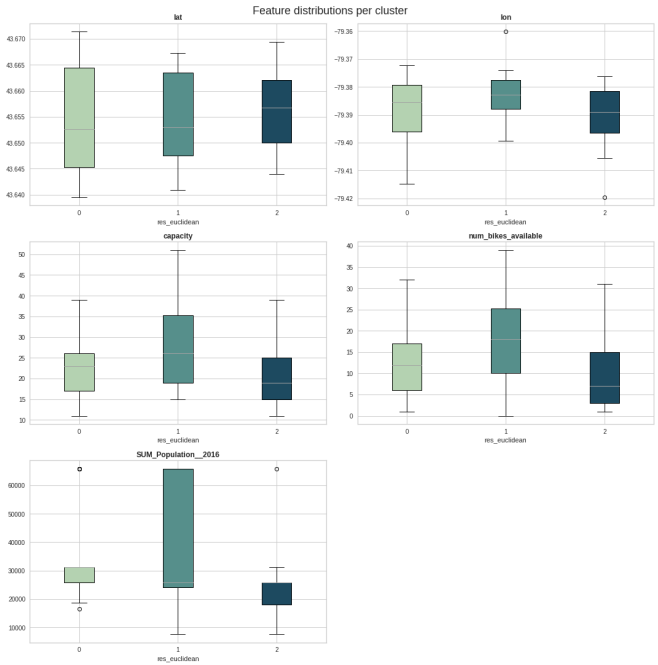


Figure 4.11: Results of box bar for Spearman method that includes five sub-figures, each of which is related to one feature of the feature vector that has been used for clustering.

4.6.3 Relative Difference

Another method to check the result of the clustering is a method that shows the types of characteristics that represent each cluster and has the ability to show which features distinguish the clusters. This method is called the relative difference. It calculates the mean value for each feature in each cluster. Then it calculates the difference between each feature per cluster with over average in the cluster in percent. Figures 4.12, 4.13,4.14,4.15 visualize the result of relative difference for four methods with five features.Each feature is depicted in its respective chart, providing insights into the clustering patterns based on these attributes. If the mean per cluster to overall mean in percent for a cluster for a feature is positive, it indicates that the average value of that particular feature within the cluster is higher than the overall average value of the same feature across all clusters. In other words, the cluster exhibits a higher-than-average value for that specific feature compared to the entire dataset. This could imply that the cluster has a relatively higher concentration or abundance of the feature being analyzed compared to the average distribution across all clusters. If the mean per cluster to overall mean in percent for a cluster for a feature is close to zero, it suggests that the average value of that feature within the cluster is approximately equal to the overall average value of the same feature across all clusters. In other words, the cluster does not exhibit a significantly higher or lower value for that specific feature compared to the dataset's overall average. This could indicate that the feature's distribution within the cluster is similar to the average distribution across all clusters, or that the feature may not play a significant role in distinguishing that particular cluster from others in the dataset. If the mean per cluster to overall mean in percent for a cluster for a feature is negative, it means that the average value of that feature within the cluster is lower than the overall average value of the same feature across all clusters. For instance, in figure 4.12 regarding latitude, the first cluster has a notably high value, while the second and third clusters display negative values.

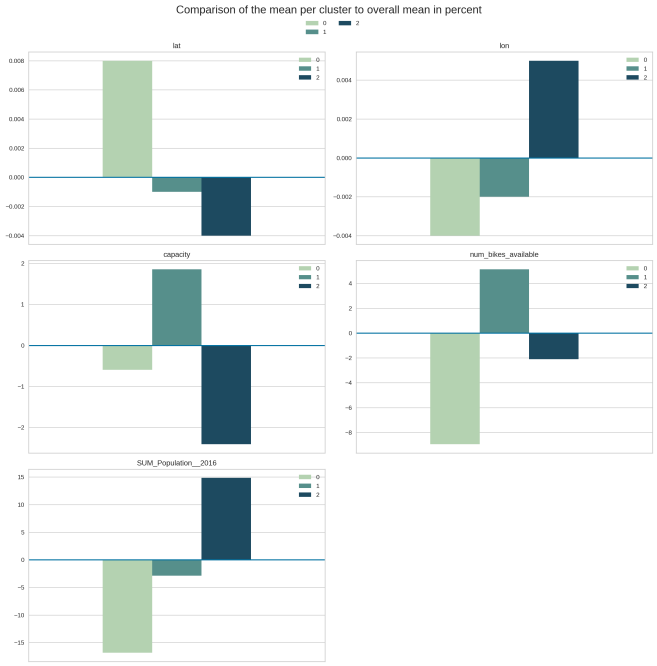


Figure 4.12: Comparison of the mean per cluster to overall mean in percent for the Euclidean method.

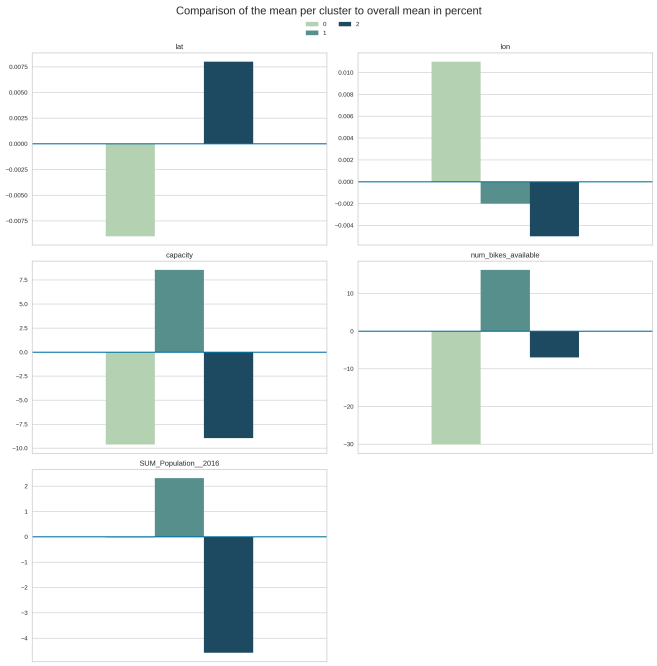


Figure 4.13: Comparison of the mean per cluster to overall mean in percent for the Manhattan method.

4.6.4 Dimension Reduction Method

As the size of the feature vector is greater than three, in order to visualize the stations in an n-dimensional space, we need to use an approach to reduce the dimensionality. One of the well-

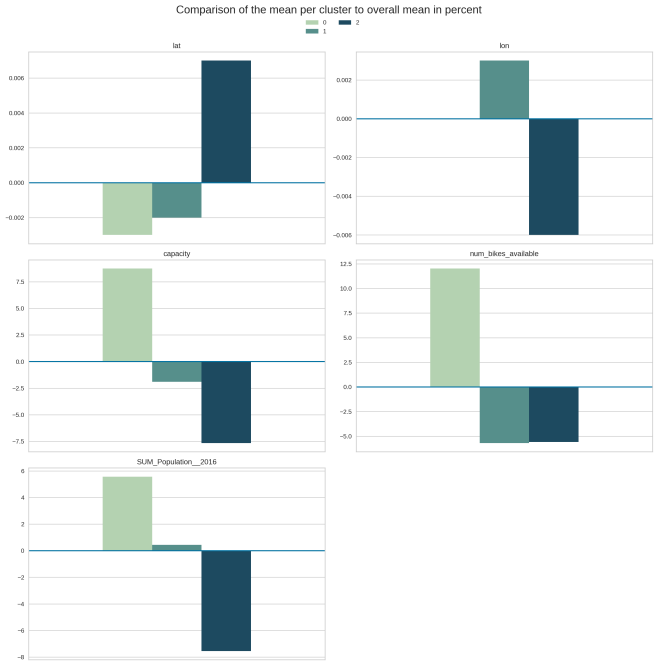


Figure 4.14: Comparison of the mean per cluster to overall mean in percent for the Pearson method.

Figure 4.15: Comparison of the mean per cluster to overall mean in percent for the Spearman method.

known methods is called Principal Component Analysis (PCA). When employing these dimension reduction techniques, there is a trade-off between preserving local and global structure. While PCA preserves global structures, it does not preserve local structures. However, there is a tech-

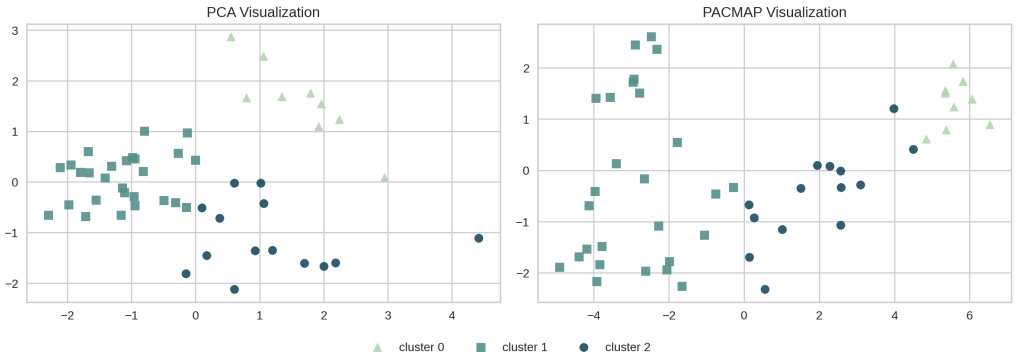


Figure 4.16: Results of Principal Component Analysis (PCA) and Pairwise Controlled Manifold Approximation (PaCMAP) of the Euclidean method.

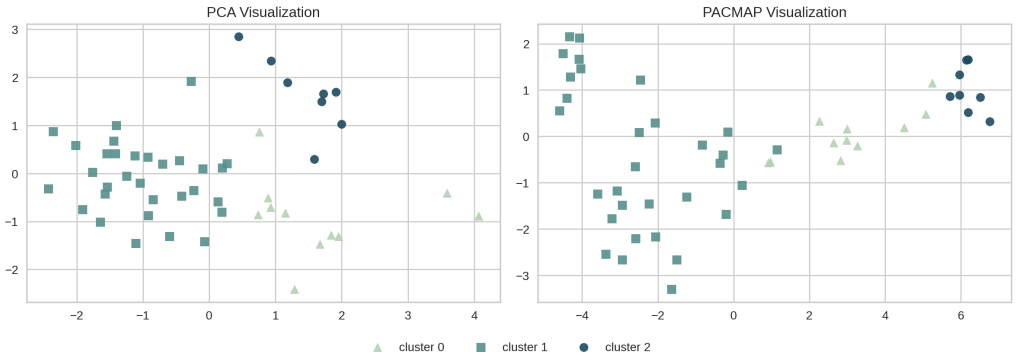


Figure 4.17: Results of Principal Component Analysis (PCA) and Pairwise Controlled Manifold Approximation (PaCMAP) of the Pearson method.

nique called Pairwise Controlled Manifold Approximation (PaCMAP) that claims to preserve both local and global structures. Figures 4.16, 4.17, 4.18 show the results of PCA and PaCMAP for Euclidean, Pearson, and Spearman methods. The vertical and horizontal axes are the principal data components. Principal components are the directions of greatest data variability, and they are used to represent the original variables in a lower-dimensional space. The first principal component is the direction in which the data varies the most. Each data point's position on the graph shows its coordinates in the new coordinate system defined by the principal components. By plotting the data in this manner, structures, and relationships that are not immediately apparent in the original variables can be identified. In the PCA Figures (4.16, 4.17, 4.18), the units of the vertical and horizontal axes differ from the units of the original variables. They represent the principal components, which are linear combinations of the initial variables. The principal components are orthogonal, or uncorrelated, with one another. The unit of measurement for the first principal component is

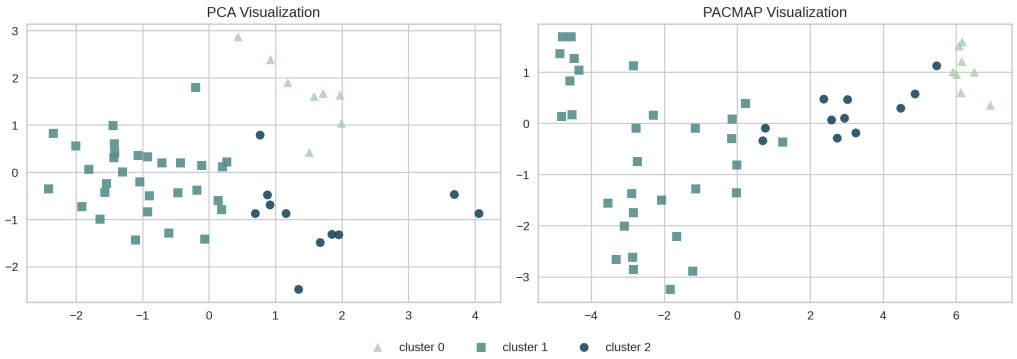


Figure 4.18: Results of Principal Component Analysis (PCA) and Pairwise Controlled Manifold Approximation (PaCMAP) of the Spearman method.

a combination of the original variables, while the unit of measurement for the second principal component is orthogonal to the first principal component.

4.6.5 Magnitude and Cardinality

Figures 4.19,4.20,4.21, and 4.22 represent the cardinality, magnitude, and cardinality per magnitude of the clusters. By looking at cardinality, the cluster sizes or the number of stations in each cluster can be determined. It helps us find clusters with significantly fewer or more stations than the other clusters. For instance, in Figure 4.20 the cardinality of cluster 1 is more than clusters 0, and 2. However, in Figures 4.22 the cardinality of the clusters are close, which means the distribution of the stations in the clusters is good. By comparing all the cardinalities, it is visualized that the order of suitability of the approaches is as follows: Manhattan, Pearson, Euclidean, and Spearman.

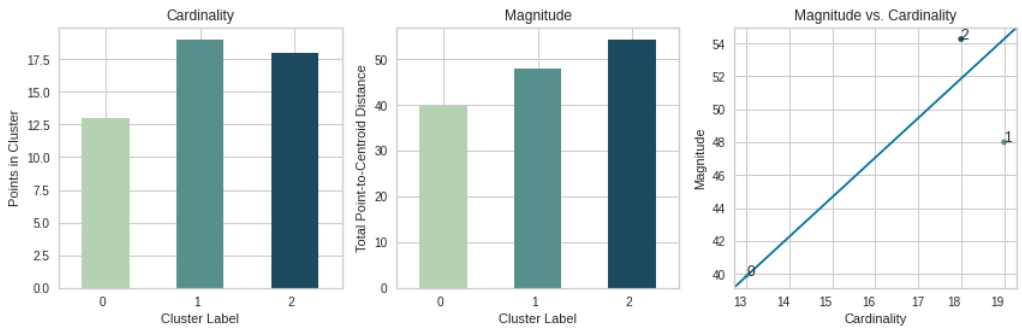


Figure 4.19: Cardinality, Magnitude, and Magnitude-Cardinality charts for the Euclidean method.

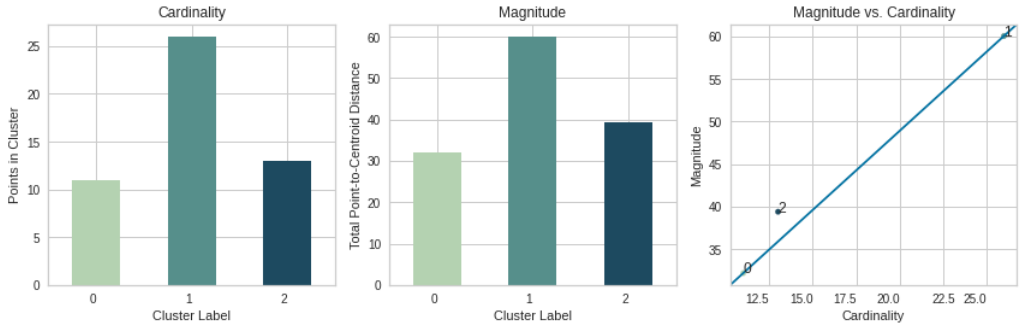


Figure 4.20: Cardinality, Magnitude, and Magnitude-Cardinality charts for the Manhattan method.

The magnitude charts in Figures 4.19,4.20,4.21, and 4.22 represent the total distance from the station to the centroid of the clusters. This would help to determine the level of spread of the stations in the clusters. By comparing all the magnitudes, it is visualized that the order of suitability of the approaches is as follows: Manhattan, Euclidean, Spearman, and Pearson.

However, cardinality and magnitude are two useful characteristics to compare and analyze the results, so it is better to check the cardinality vs. magnitude plot. The vertical axis is the magnitude, and the horizontal axis is the cardinality. In an ideal cluster, the output lies on or will be very close to the y=x line. Any anomalies in the clusters will deviate from the y=x. For instance, in Figure 4.20, clusters 0, and 1 lie on the y=x line, however, cluster 2 is far from the line. By comparing the results of these charts, the order of suitability of the approaches is as follows: Euclidean, Pearson, Manhattan, and Spearman.

Our results demonstrate the effectiveness of this approach in grouping stations based on multiple features, offering valuable insights into the distribution and characteristics of these clusters. We employed various dissimilarity functions, including Euclidean, Manhattan, Pearson, and Spearman, to

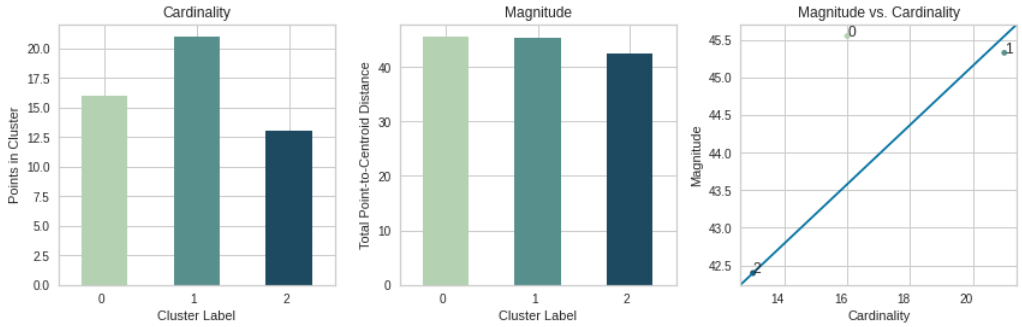


Figure 4.21: Cardinality, Magnitude, and Magnitude-Cardinality charts for the Pearson method.

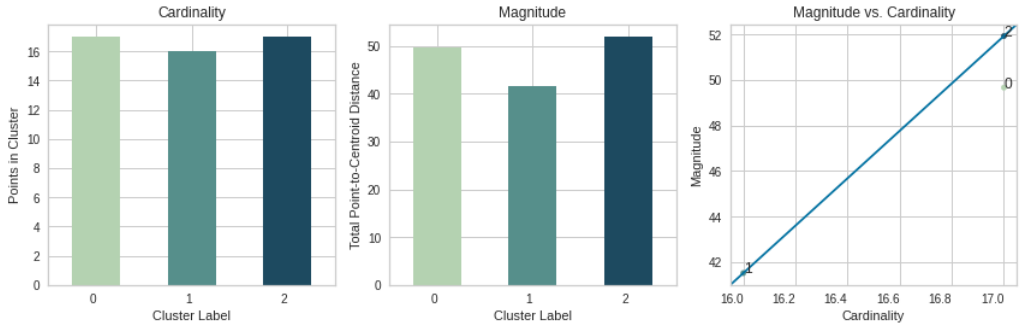


Figure 4.22: Cardinality, Magnitude, and Magnitude-Cardinality charts for the Spearman method.

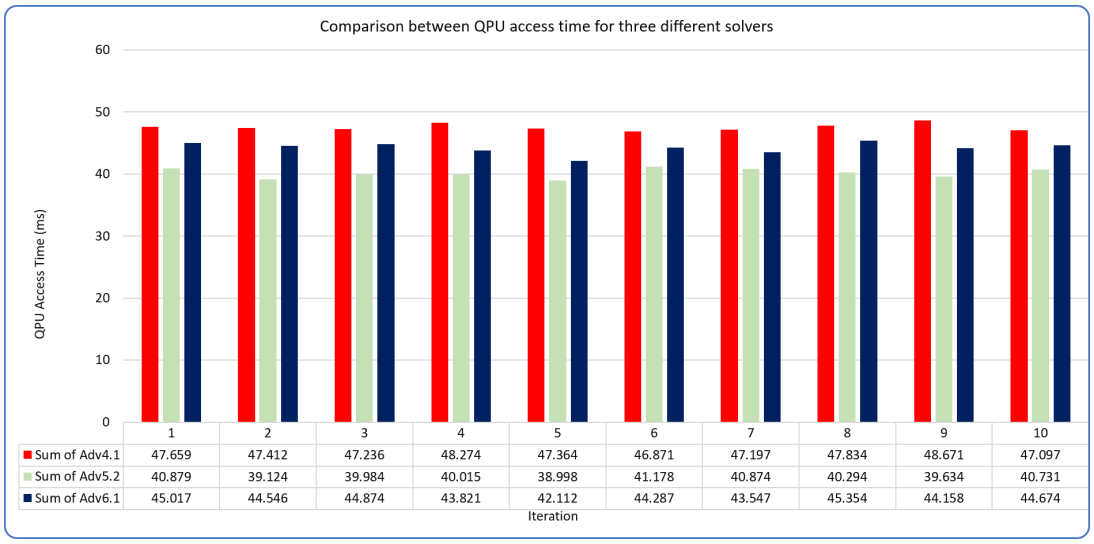
assess their impact on clustering quality. Among these, Spearman emerged as the most promising method, showcasing its suitability for real-world applications like bike-sharing systems. Furthermore, we analyzed the relative difference to identify the distinguishing features of each cluster, providing a deeper understanding of the station groupings. Moreover, dimensionality reduction techniques such as Principal Component Analysis (PCA) and Pairwise Controlled Manifold Approximation (PaCMAP) were also explored to visualize the clusters in lower-dimensional spaces. Our analysis of cardinality and magnitude offered valuable insights into the distribution and spread of stations within the clusters.

4.6.6 Performance of QC Solvers

This paper examined three QC solvers operating within the QPU architecture: ”Advantage system4.1,” ”Advantage system5.2,” and ”Advantage system6.1.” These solvers are implemented and function within the QPU framework. ”Advantage system4.1” is situated in North America and

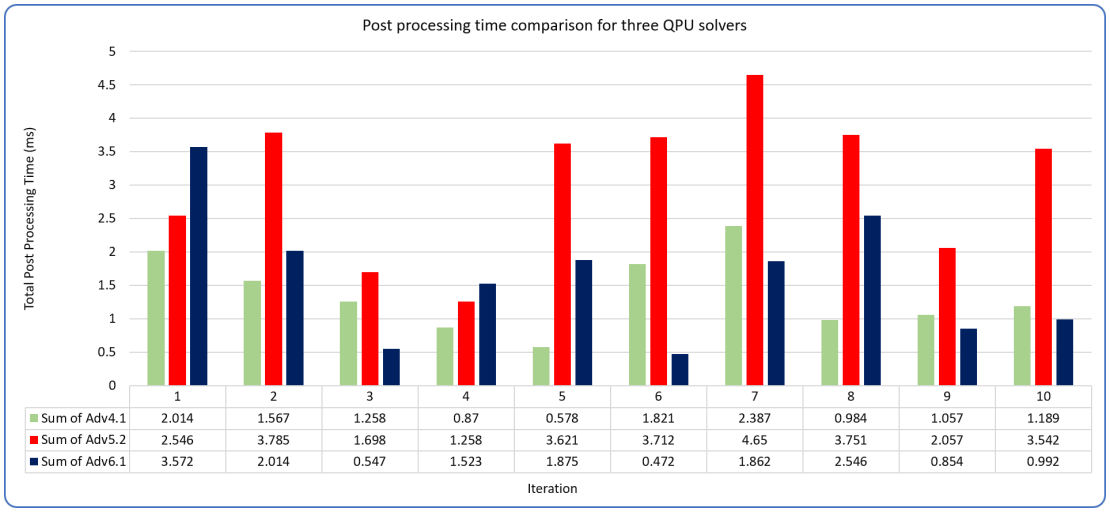
consists of 5760 qubits, with 5627 active qubits at the time of writing this paper, supporting both Ising and QUBO models, utilizing a pegasus graph topology. Similarly, ”Advantage system5.2” is located in Europe, featuring 5760 qubits, with 5625 active at the time of writing, also supporting both Ising and QUBO models with a pegasus graph topology. ”Advantage system6.1,” also located in North America, comprises 5760 qubits, with 5612 active at the time of writing, supporting both Ising and QUBO models with a pegasus graph topology. One crucial metric in evaluating solver performance is the QPU Access Time (QAT), indicating the duration during which the model can access the quantum computer. Once a query is dispatched to the D-Wave cloud computers, it enters a queue and is sequentially processed by the QPU. As illustrated in Figure 4.23, ”Advantage system5.2” demonstrates faster processing compared to the other two solvers. Regarding post-processing time, the findings reveal that ”Advantage system4.1” exhibits the shortest average duration at 1.372 ms, as depicted in Figure 4.24.

To compare the results of QC with classical computers, it is essential to acknowledge the challenge of finding a reliable method for comparing performance in terms of runtime. This challenge arises from the need to run the algorithms multiple times, as QC results require instance embedding, unlike classical algorithms. The runtime in QC includes internet latency, worker queue, pre-processing, QPU access time, and post-processing time (Figure 4.25). Figure 4.26 displays the runtime results for clustering on QC and k-means clustering on classical computers. The results are collected for 12 iterations. The QC runtime does not include the time that is required for embedding the data before sending it to the actual quantum computers. At the time of writing this paper, adding embedding time to the QC runtime concludes that classical computers outperform QC because the query to run the algorithm on D-Wave quantum cloud computers is queued. However, if we disregard the embedding time and the time that the query waits to be processed on quantum computers, the results of QC outperform the classical computers.



	1	2	3	4	5	6	7	8	9	10
Sum of Adv4.1	47.659	47.412	47.236	48.274	47.364	46.871	47.197	47.834	48.671	47.097
Sum of Adv5.2	40.879	39.124	39.984	40.015	38.998	41.178	40.874	40.294	39.634	40.731
Sum of Adv6.1	45.017	44.546	44.874	43.821	42.112	44.287	43.547	45.354	44.158	44.674

Figure 4.23: Quantum Processor Unit (QPU) Access Time for three different solvers in 10 iterations.



	1	2	3	4	5	6	7	8	9	10
Sum of Adv4.1	2.014	1.567	1.258	0.87	0.578	1.821	2.387	0.984	1.057	1.189
Sum of Adv5.2	2.546	3.785	1.698	1.258	3.621	3.712	4.65	3.751	2.057	3.542
Sum of Adv6.1	3.572	2.014	0.547	1.523	1.875	0.472	1.862	2.546	0.854	0.992

Figure 4.24: Post processing time for three different solvers in 10 iterations.

4.7 Conclusion

In this paper, a hybrid QC model to cluster stations in a BSS by solving it as a CSP problem with different methods on a QA computer is proposed. Formulating the problem into a QUBO model is explained step by step. Three different methods have been used to determine the number of clusters and Euclidean, Manhattan, Pearson, and Spearman dissimilarity functions have been used to cluster the stations. The evaluation is done using the magnitude vs. cardinality approach. The distribution of the stations, magnitude, and cardinality of the results indicate that it is possible to use QC for

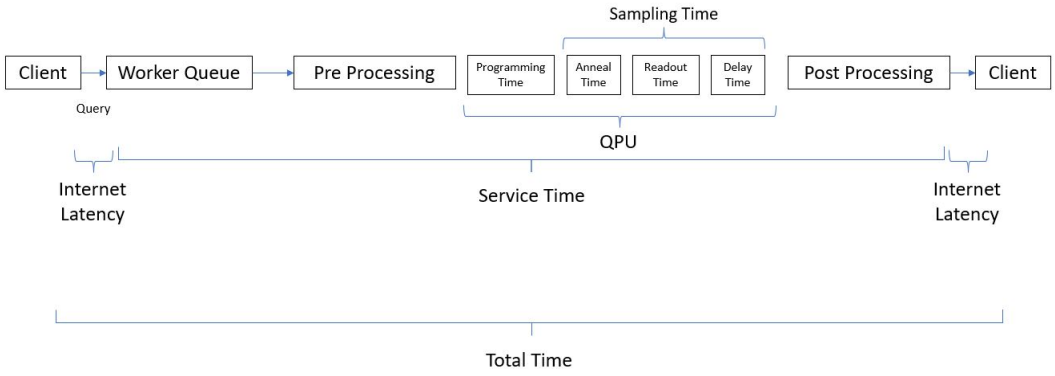
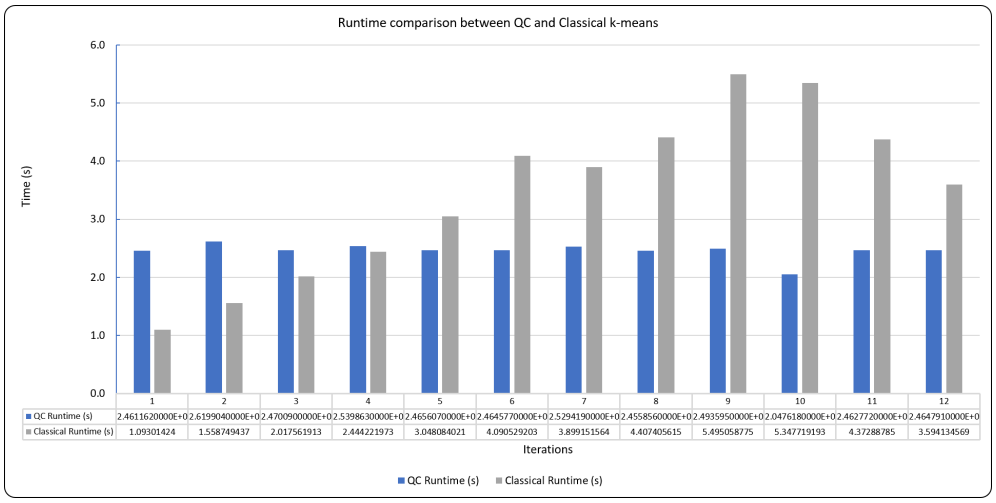


Figure 4.25: QC runtime overview. In this research total time is considered as runtime.



	1	2	3	4	5	6	7	8	9	10	11	12
QC Runtime (s)	2.461162000E+0	2.619904000E+0	2.470090000E+0	2.539863000E+0	2.465607000E+0	2.464577000E+0	2.529419000E+0	2.455856000E+0	2.493595000E+0	2.047618000E+0	2.462772000E+0	2.464791000E+0
Classical Runtime (s)	1.09301424	1.558749437	2.017561913	2.444221973	3.048084021	4.090529203	3.899151564	4.407405615	5.495058775	5.347719193	4.37288785	3.594134569

Figure 4.26: Comparison between runtime of QC and classical computing without considering the time that is needed for embedding in QC.

clustering for a real-world application, which is in this case BSS. Moreover, the Spearman dissimilarity function showed better performance compared to the other functions for this purpose. Thus, the proposed model has the potential to be applied to real-world applications for clustering items with multiple features. Despite the advantages of QC, several challenges and limitations exist that require attention. Hybrid QC requires transferring data between classical and quantum units which affects the overall computational speed. Moreover, QA is very sensitive to noises and environmental factors which may lead to inaccurate solutions, which is why different iterations are required to validate the solution. While this study presents promising results in utilizing QC for clustering BSS

stations, several avenues for future research remain open. These include optimizing data transfer between classical and quantum units to enhance computational efficiency. Additionally, mitigating the impact of noise and environmental factors on QA outcomes by refining error correction techniques and exploring novel approaches for solution validation through multiple iterations presents an intriguing research direction.

Chapter 5

Machine Learning for Sustainable Transportation Management: Predicting Bike Sharing System Ridership with Geospatial Features

5.1 Preface

This chapter is important as a requirement for the next chapter, where a new Feature Selection (FS) method is proposed. The insights and results from this chapter form a critical foundation for developing and validating the proposed FS method, ensuring it is robust and tailored to the specific needs of BSS ridership prediction. Moreover, the methodologies and findings presented in this chapter highlight the complexities of BSS ridership prediction, such as the interplay between weather conditions, geographical features, and historical usage patterns. This detailed understanding is crucial for developing an FS method that can effectively capture these dynamics and improve model performance. Ultimately, the work in this chapter lays the groundwork for advancing predictive modeling in BSS, contributing to more effective system management and enhanced user

satisfaction.

The main focus of this chapter is to investigate various modeling approaches to predict BSS user numbers, focusing on Toronto's BSS as a case study. Accurate ridership prediction is essential for managing these systems effectively, optimizing resource allocation, and enhancing user experience. To find the best approach for this case, different techniques, including LR, RF, and XGBoost are investigated by analyzing historical usage patterns, weather conditions, and geographical features. Our findings reveal that XGBoost Poisson regression with exposure is the most effective method for predicting BSS ridership, providing valuable insights into managing these systems efficiently.

The content of this chapter is published in the Canadian Society for Civil Engineering (CSCE) 2024 Annual Conference as follows:

A. Nourbakhsh, M. Jadidi, and K. Shahriari. "Machine Learning for Sustainable Transportation Management: Predicting Bike-Sharing System Ridership with Geospatial Features." *Canadian Society for Civil Engineering (CSCE) 2024 Annual Conference, 2024*.

5.1.1 Copyright and Licensing

5.1.2 Contributions

The roles of the authors in this chapter are outlined as follows: Amirhossein Nourbakhshrezaei was responsible for the literature review, data collection and organization, methodological development, implementation for analysis and modeling, result validation and visualization, and the initial manuscript writing. Mojgan Jadidi oversaw the research, provided the funding, and contributed to the methodology development, as well as the writing and editing of the manuscript. Kyarash Shahriari supervised the research and assisted in the manuscript editing.

5.2 Abstract

Bike Sharing Systems (BSS) have become one of the main components of urban transportation networks, providing convenient and sustainable alternatives for commuters. Accurate prediction of ridership is crucial for efficiently managing these systems, ensuring optimal resource allocation, and enhancing user experience. This paper explores various modeling approaches to predict the number of users in BSS, focusing on the Toronto BSS as a proof of concept. These foundational steps are essential for extracting meaningful insights from the raw data generated by BSS. Understanding the intricate relationships between temporal, environmental, and spatial factors enables the development of robust predictive models. Our study applies a diverse set of ML techniques, including LR, linear Poisson regression, RF, XGBoost regression, XGBoost Poisson regression, and XGBoost Poisson regression with exposure. Each model brings a unique perspective to the task, leveraging its strengths in capturing complex patterns within the data. The analysis encompasses historical usage patterns, weather conditions, and geographical features to enhance the predictive capabilities of our models. By comparing and contrasting different approaches, we provide valuable insights into the strengths and limitations of each method. Results indicate varying degrees of predictive performance among the models, with XGBoost Poisson regression with exposure emerging as the most effective method for anticipating the number of users in the BSS.

5.3 Introduction

In contemporary urban landscapes, where sustainability and efficiency intertwine, BSS have emerged as integral components of transportation networks, offering a viable and eco-friendly alternative for daily commuters [189]. As metropolitan areas deal with the challenges of congestion, pollution, and the ever-growing demand for accessible transportation, the need for effective management of BSS becomes paramount [190]. Accurate prediction of ridership is a must in achieving this management efficacy, facilitating optimal resource allocation, and ultimately enhancing the overall user experience [191].

Since the new methodologies in time series problems are evolving, this paper describes the realm of predictive modeling for BSS ridership, with a particular focus on the vibrant and dynamic city of Toronto as a compelling case study. The intricate interplay between temporal, environmental, and spatial factors in the context of bike-sharing necessitates a nuanced and comprehensive approach to modeling.

Our exploration extends beyond conventional predictive methodologies [192], incorporating a diverse set of ML techniques to unravel the underlying patterns within the data. Fundamental to our investigation is the recognition that BSS ridership is shaped by multifaceted influences, including historical usage patterns, meteorological conditions, and the geographical context in which these systems operate [193]. To this end, we explore a spectrum of ML models, ranging from traditional LR to more sophisticated ensemble methods like RF and XGBoost [194]. Each model, characterized by its unique strengths and adaptability, contributes to our holistic understanding of the complex relationships embedded in BSS ridership dynamics.

Throughout our exploration, this paper aims to not only showcase the efficacy of diverse ML techniques but also to provide a comparative analysis of their strengths and limitations in predicting BSS ridership [195]. The culmination of our efforts reveals valuable insights, with XGBoost Poisson regression with exposure emerging as the most effective method for anticipating the number of users in the BSS. As we navigate through the intricacies of modeling sustainable transportation management, our study contributes to the broader discourse on leveraging data-driven approaches for a more resilient and responsive urban mobility landscape.

The identified superior performance of XGBoost Poisson regression with exposure highlights its potential as a robust tool for stakeholders in the transportation sector seeking accurate ridership predictions to optimize system efficiency and user satisfaction.

5.4 Literature Review

Many researchers have been working on predicting ridership in BSS and transportation mobility. In this section, the most relevant research is summarized, and at the end of the section, the difference between this paper and other papers is described.

The study by Soheil Sohrabi et al. (2020) focused on the development of generalized extreme value (GEV) count models to predict hourly bike arrivals and departures in BSS. These models consider various factors such as time-of-day, weather, built environment, infrastructure, temporal, and spatial dependencies. By analyzing demand patterns in the BSS, the study demonstrates the models' ability to predict demand at both aggregate and disaggregate levels with reasonable accuracy [196].

A study on forecasting usage and bike distribution in Dockless Bike-Sharing (DBS) systems, which lack fixed stations or docks, is written by Mingzhuang Hua et al. The study, based on Mobike journey data from Nanjing, China, employs K-means clustering to divide virtual stations and processes usage and bike count data from 4000 such stations. Utilizing RF, the study predicts real-time passenger departure, passenger arrival, and bike count in these virtual stations. Results indicate a positive correlation between bike count and usage, with RF demonstrating superior predictive accuracy compared to benchmark methods [197].

Another research extracts novel time-lagged variables from real-world bike usage datasets, including graph structures and flow interactions, and utilizes them as inputs to various ML algorithms to predict short-term bike demand. The experiments reveal that graph-based attributes are more influential in demand prediction than commonly used meteorological information [198].

Another study examines the factors influencing the ridership of BSS and free-floating bike sharing (FFBS) at rail transit stations, aiming to address the "first/last mile" problem. Based on user transaction records from two BSSs in Nanjing, China, the study first classifies rail transit stations into five types using the k-means cluster method based on temporal bike sharing usage profiles [199].

A team of researchers presented a study aimed at predicting station-level demand for pickup and drop-off of bicycles in BSSs, crucial for enhancing system efficiency and reducing operational costs. Leveraging station activity information, including time and weather data, alongside the

number of pickups and drop-offs 1–3 hours before prediction, a RF ML technique is employed for demand prediction [200].

[201] discussed the spatio-temporal patterns of dockless bike sharing demand and factors influencing these patterns to enhance operational efficiency. Utilizing bicycle trip data from Mobike, Point of Interest (POI) data, and smart card data in Beijing, the study constructs a spatially embedded network and applies the Infomap algorithm for community detection to uncover usage patterns.

5.5 Methodology

5.5.1 Data Exploration

In this section, data exploration is described, which is the first step to analyzing data to get a better understanding and complete insight into the dataset. Figures [5.1, 5.2] show the histogram of ridership at start and end stations. The histograms show that the number of users per station at the beginning of a trip is approximately the same as at the end of the trip in a station. That is why this paper considers start stations as the number of ridership.

Figure [5.3, 5.4] are time series plot showing the duration of trips available in BSS over a period of one year. Figure 5.3, categorized dataset based on user type; annual and casual members. Figure 5.4, classified the dataset based on time of a day; AM, and PM. The x-axis represents time, labeled with months from January to December. Before doing any analysis, visually it is obvious that there is a seasonal trend in the data, with rid-ership increasing during the summer months (June, July, August) and decreasing dur-ing the winter months (December, January, February). This suggests that more people use the BSS when the weather is good. Additionally, the ridership is consistently high-er in the fall (September, October, November) than it is in the spring (March, April, May). This could be due to a number of factors, such as more people commuting to work by bike during the fall months, or students returning to school. There are some short-term fluctuations in ridership throughout the year. These fluctuations could be due to a variety of factors, such as weather events, special events, or holidays.

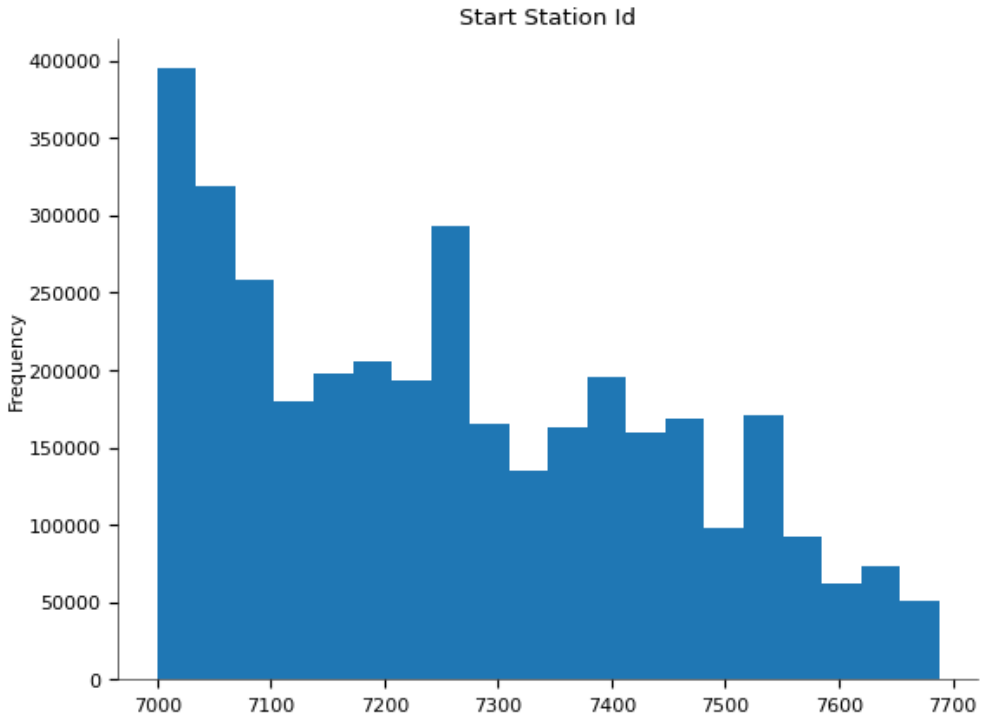


Figure 5.1: Histogram of ridership in start stations.It shows frequency based on start station id

.

Figures [5.5, 5.6] represent a visual exploration of the temporal dynamic of ridership over a year within the Toronto BSS. Divided into two panels, the Figure 5.5 shows the monthly ridership trends over the course of a year for five randomly selected stations. This granular representation allows for an in-depth analysis of the ridership patterns at individual stations, unveiling potential nuances and localized variations. Moreover, Figure 5.6 synthesizes the collective ridership across all stations, presenting a simple view of the system's overall usage throughout the year. A compelling observation arises from the comparative analysis between these two perspectives: the apparent correlation between ridership and months. The dynamic trajectories exhibited by the five random stations collectively mirror the trend observed in the total monthly ridership, underscoring a strong and cohesive association between ridership patterns and the passage of time. This insight not only illuminates the systemic behavior of the BSS network but also underscores the potential impact of temporal factors on ridership fluctuations across diverse stations within the system.

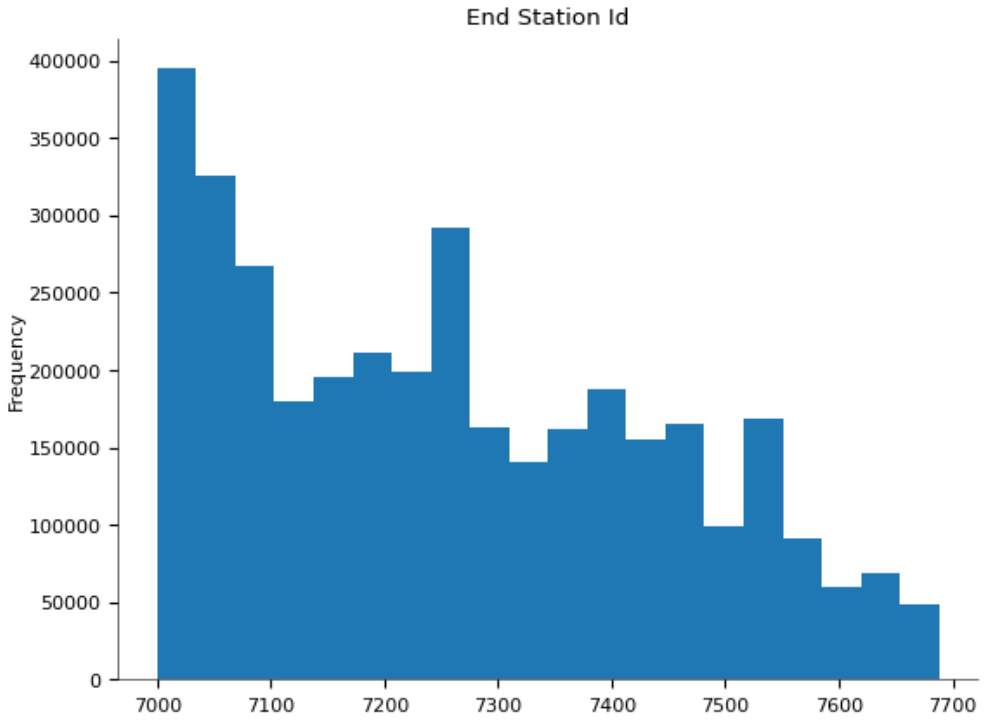


Figure 5.2: Histogram of ridership in end stations. It shows frequency based on end station id

.

5.5.2 Machine Learning Algorithms

After checking preliminary exploration of our dataset in the preceding section, where visual inspections and statistical summaries provided valuable insights, we now tran-sition into more numerical and analytical approaches. The focus of the paper shifts towards robust numerical checks and analyses to extract correlations within the data. At the forefront of this methodological progression lies the critical step of feature engineering. Unlike the visual explorations employed earlier, feature engineering in-volves a deeper examination of numerical attributes that can enhance the understanding of the underlying patterns. This process is fundamental for refining the data and preparing it for subsequent modeling stages. Feature engineering is described in the following sub-section.

5.5.3 Feature Engineering

In this paper, feature engineering takes center stage as the goal is to refining the da-taset for a more effective analysis. This process involves not just tweaking existing features but also choosing new

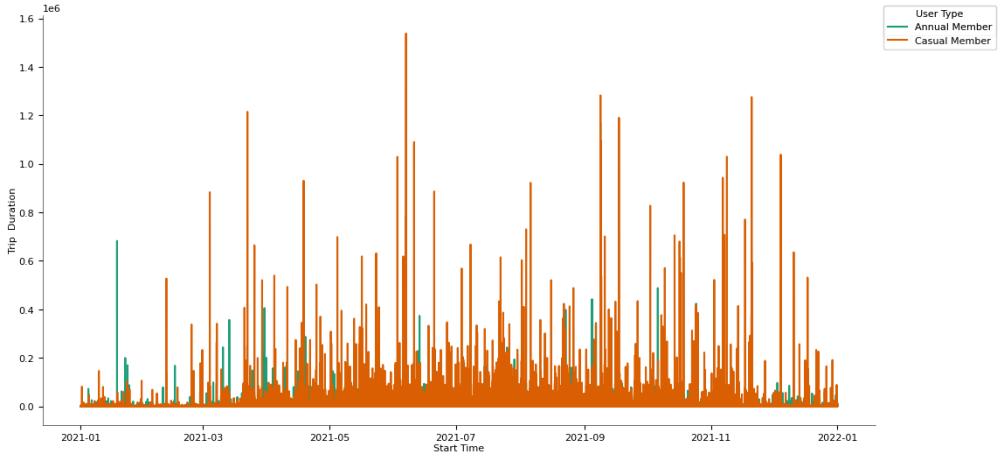


Figure 5.3: Distribution of trip duration over a year classified based on annual, and casual members.

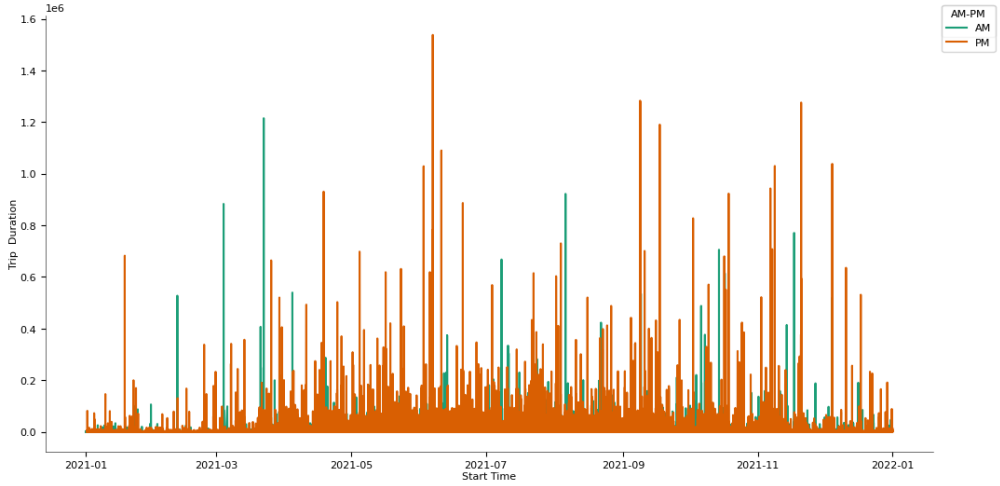


Figure 5.4: Distribution of trip duration over a year classified based on AM, and PM in a day.

ones to uncover meaningful patterns. A crucial part of feature engineering is “feature selection”, where the most relevant attributes for the predictive models should be selected. By assessing each feature’s importance, we avoid unnecessary complexity in the models and make them easier to understand. In this paper, linear correlation-based method, and MI are used to check the features, keeping only the ones that really matter. This not only makes our models more accurate but also helps manage the amount of information that need to be considered.

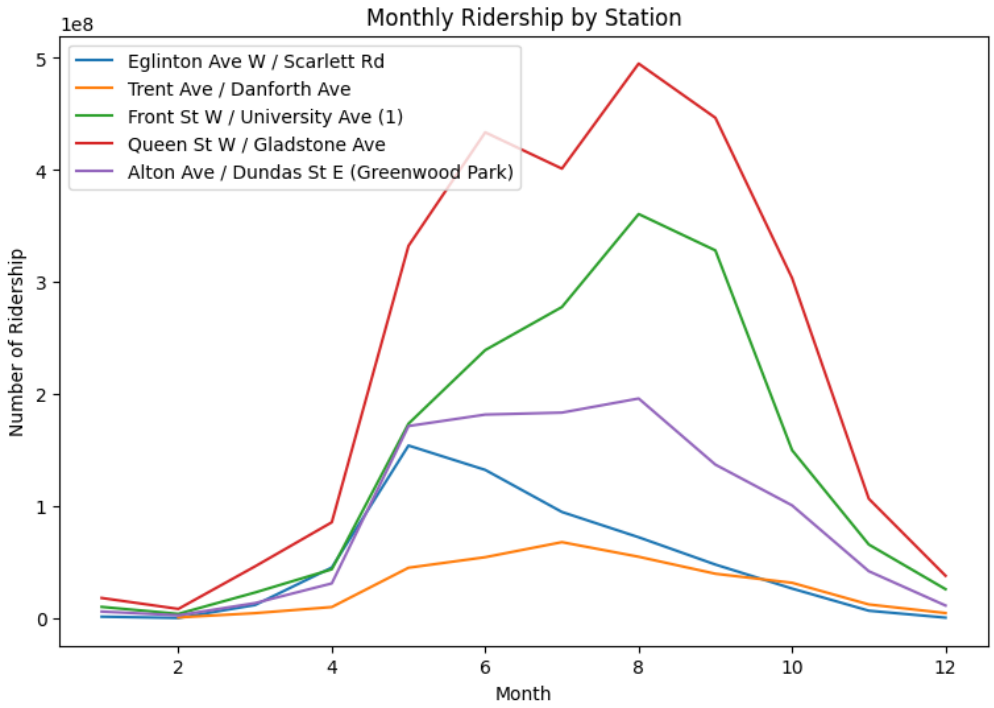


Figure 5.5: Monthly ridership over a year for five random stations.

5.5.4 Linear Correlation-based Method

TheCorrelation-based Feature Selection (CFS) method is an approach for selecting sets of features that exhibit strong correlations with the target variable while maintaining low correlations among themselves. The primary aim of CFS is to pinpoint a subset of features that maximizes information about the target variable while minimizing redundancy within the chosen features.

The algorithm for CFS initiates by computing the correlation between each feature and the target variable. Subsequently, it evaluates the correlation among all pairs of features. The algorithm then proceeds to identify the subset of features that displays the highest correlation with the target variable and, simultaneously, the least correlation among themselves. In this paper, Pearson correlation coefficient is used since the goal is to use a regression model to predict the ridership (Equation 5.1).

$$r_{zc} = \frac{k\bar{r}_{zi}}{\sqrt{k + k(k-1)\bar{r}_{ii}}} \tag{5.1}$$

where r_{zc} is the correlation between the components and the outside variable,

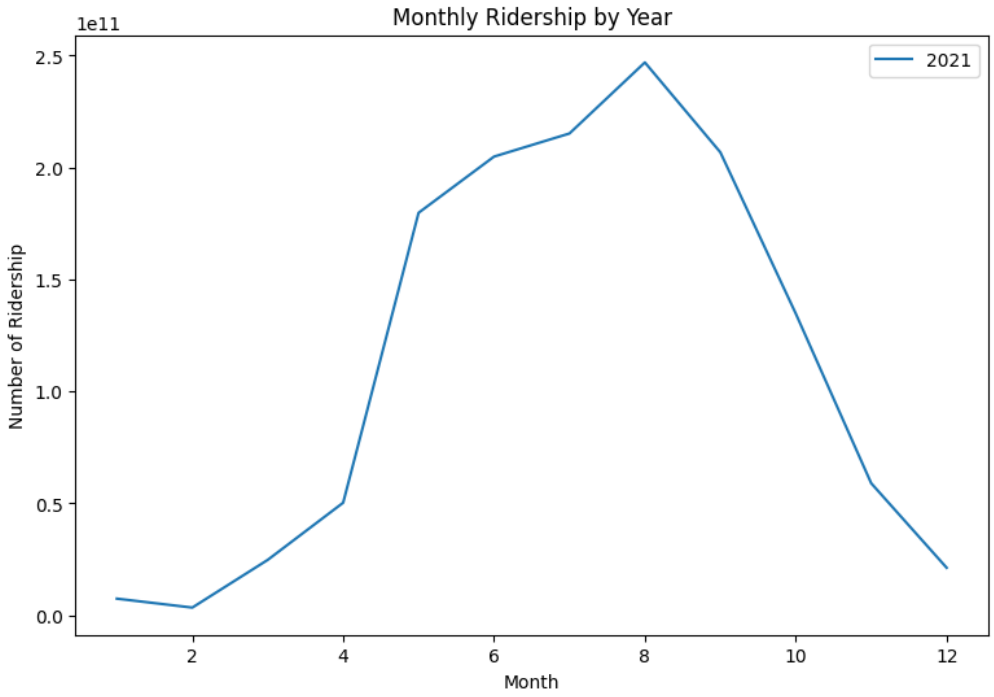


Figure 5.6: The total monthly ridership over a year.

- k is the number of components,
- $\bar{r}_{zi}$ is the average of the correlations between the components and the outside variable, and
- $\bar{r}_{ii}$ is the average inter-correlation between components.

5.5.5 Mutual Information (MI)

Since the correlation matrix can detect linear correlation, we need to apply another approach that identifies the non-linear correlation between features. In this paper, MI is used. MI is a statistical measure that quantifies the degree of dependence between two variables by assessing the amount of information gained about one variable when the other is known. In essence, it captures the extent to which knowledge of one variable reduces uncertainty about the other. The formula for MI, denoted as $I(X;Y)$ for two variables X and Y, involves calculating the sum over all possible outcomes of the joint probability distribution of X and Y, multiplied by the logarithm of the ratio of the joint probability to the product of the individual probabilities (Equation 5.2).

$$I(X;Y) = \sum_{X}\sum_{Y} p(x,y)\log\left(\frac{p(x,y)}{p(x)p(y)}\right) \tag{5.2}$$

- $p(x,y)$ is the probability of the two variables,
- $p(x), p(y)$ are the marginal probabilities.

5.5.6 Linear Regression

LR is a fundamental statistical method used for modeling the relationship between a dependent variable and one or more independent variables. The model assumes a linear connection between the variables, aiming to find the best-fitting line that minimizes the sum of squared differences between predicted and observed values. This approach is widely employed for its simplicity and interpretability, making it a valuable tool for predicting numeric outcomes (Equation 5.3).

$$y = \beta_0 + \beta_1 X + \varepsilon \tag{5.3}$$

where y is the predicted value, X is the independent variable, β_0 is the intercept, β_1 is the regression coefficient, and ε is the variation of the estimated coefficient.

5.5.7 Linear Poisson

Linear Poisson regression is an extension of the LR model designed specifically for count data, where the dependent variable represents the number of occurrences of an event. This method accommodates the inherent characteristics of count data, such as non-negativity and discrete values, by modeling the response variable with a Poisson distribution. It is particularly useful when the outcome variable involves counting events over time or space. Equation 5.4 shows the formula of Linear Poisson regression.

$$P(Y_i = y_i \mid X_i, \beta) = \frac{e^{-\exp\{X_i\beta\}}(\exp\{X_i\beta\})^{y_i}}{y_i!} \tag{5.4}$$

where Y is the dependent variable, X_i are the predictors, $\exp\{X_i\beta\}$ is the rate of Poisson distribution, and β is the regression coefficient.

5.5.8 Random Forest

RF is an ensemble learning method that leverages the power of multiple decision trees to improve predictive accuracy and robustness. Each tree is constructed independently, and the final prediction is determined by aggregating the individual tree predictions. RF is known for its ability to handle complex relationships in data, mitigate overfitting, and provide feature importance rankings, making it a versatile choice for various regression and classification tasks (Equation 5.5).

$$RFfi_i = \frac{\sum_{j \in \text{all trees}} \text{normfi}_{ij}}{T} \tag{5.5}$$

where $RFfi_i$ is the importance of feature i in the RF model, normfi_{ij} is the normalized value for feature importance of feature i in tree j, and T is the number of trees.

5.5.9 XGBoost Regression

XGBoost is a popular and powerful gradient boosting framework that excels in predictive modeling tasks. In the context of regression, XGBoost builds an ensemble of weak learners (typically decision trees) and sequentially refines them to optimize the predictive performance. It incorporates regularization techniques to prevent overfitting, handles missing data, and allows for customized loss functions, making it a robust and flexible regression tool (Equation 5.6).

$$y_i = \phi(x_i) = \sum_{k=1}^{K} f_k(x_i) \tag{5.6}$$

where y is the predicted value for each instance, x_i is the feature vector, K is the number of trees, and $f_k(x_i)$ is the output of the k-th tree for the i-th instance.

5.5.10 XGBoost Poisson

XGBoost Poisson regression is a variant of XGBoost specifically tailored for count data, following a Poisson distribution. Similar to its regression counterpart, it optimizes the model through boosting weak learners, but it adjusts its framework to accommodate the characteristics of count variables. This makes XGBoost Poisson well-suited for predicting counts or frequencies, commonly encountered in scenarios like traffic flow or user interactions. The formula for XGBoost Poisson is similar to Equation 5.6, with an exponential function added to it to ensure the predicted values are non-negative. This approach should be used whenever the target variable follows a Poisson distribution (Equation 5.7).

$$y = \frac{\lambda^k e^{-\lambda}}{k!} \tag{5.7}$$

where k is the number of events in a given interval, λ is the mean number of occurrences during the interval, and e is the base of the natural logarithm.

5.5.11 XGBoost Poisson and Exposure

XGBoost Poisson regression with exposure extends the traditional Poisson regression by incorporating an exposure term. The exposure term represents the time or space that each observation is exposed to the risk of the event, addressing situations where the data is not collected uniformly across all instances. This variant is particularly useful in scenarios where the exposure duration varies among observations, ensuring a more accurate modeling of the underlying process and improving the reliability of predictions.

5.6 Results and Discussion

The results of this paper reveal two main aspects: the selection of features and the performance of different methods. First, when choosing features, tools like scatter charts and correlation matrices

are used to understand how certain factors, like month number and trip duration, relate to bike ridership. The scatter chart worked well for numerical features. To include both numerical and categorical features, a correlation matrix is being used (Figures 5.7, 5.8). However, because the correlation matrix only considers linear relationships, the MI approach is employed for a more thorough analysis, as seen in Figure 5.9. MI highlighted key features for training the models, such as Bike ID, Trip Duration, Trip ID, Start and End Stations, Day-y, Day-x, Day, and Month Number and Name.

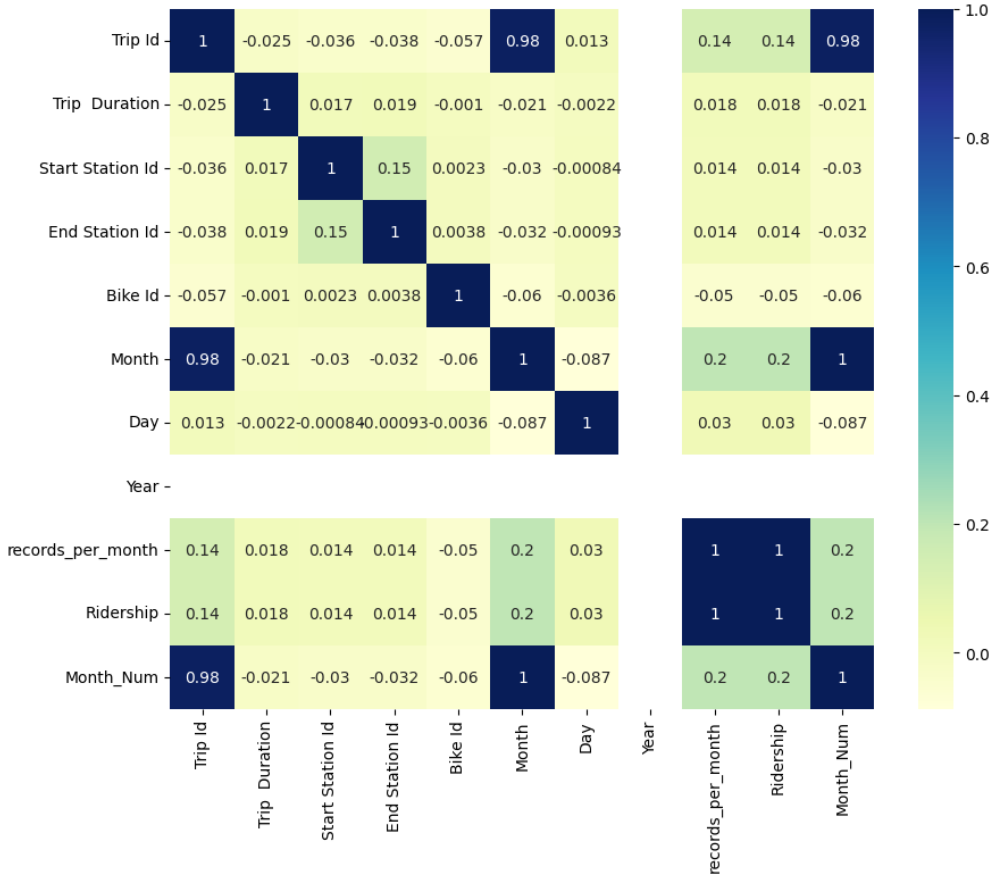


Figure 5.7: Correlation matrix for the features before one-hot encoding.

Another important point is the way time-related features are represented. To make sure there is a smooth transition between the end of one month and the start of the next, the ”day” feature is transformed into a 2D space. This shift ensures that the first day of each month is close to the 30th and 31st of the previous month. A similar trans-formation is explored for weeks, but it did

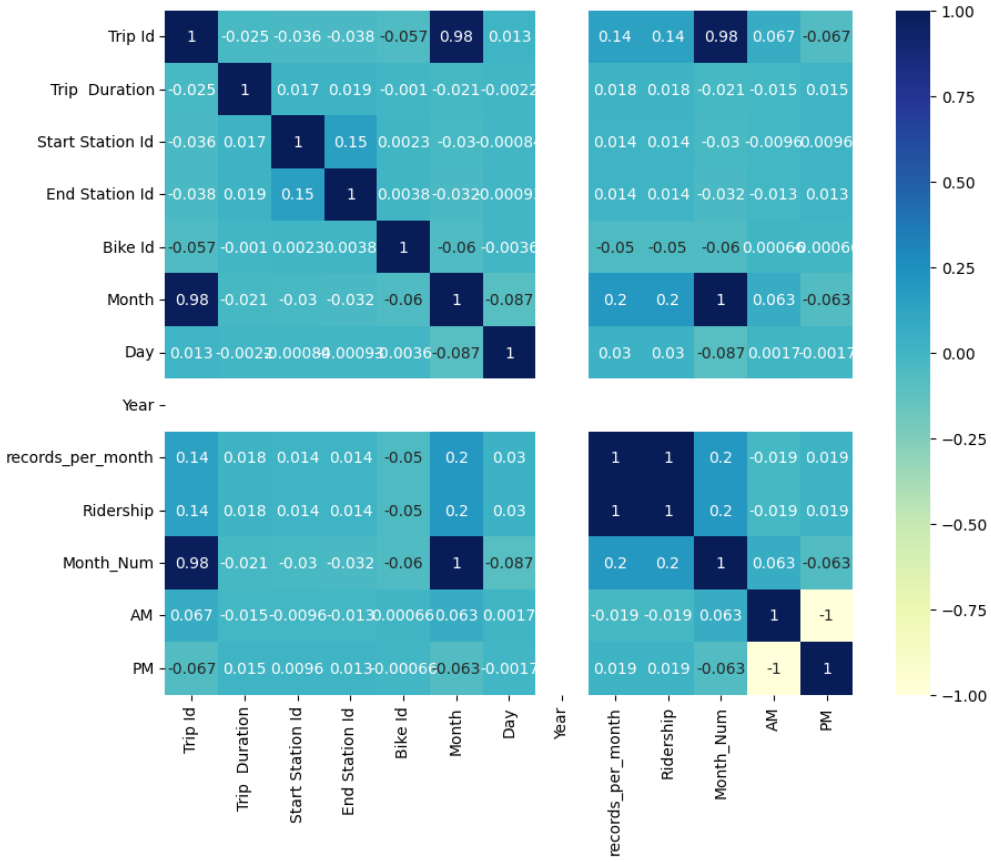


Figure 5.8: Correlation matrix for the features after one-hot encoding.

not significantly impact feature im-portance. The exploration of FS methods not only helped to under-stand how different factors relate to ridership but also identified the most important features for building accurate predictive models.

The results showed that one-hot encoding [202] works well on AM-PM features and types of users (Annual, and Casual). The adoption of one-hot encoding is pivotal for several reasons. It allows us to incorporate categorical information into our predictive models, which often rely on numerical input. By transforming categorical features into a binary format, we ensure that the model can effectively learn and respond to the unique characteristics associated with each category. This method prevents the model from assigning unintended ordinal relationships to categorical values and enhances its ability to capture nuanced patterns within the data.

After fine-tuning the parameters of the above-mentioned models to optimize predictive performance, the cross-validation techniques are employed, then model efficacy are iteratively assessed

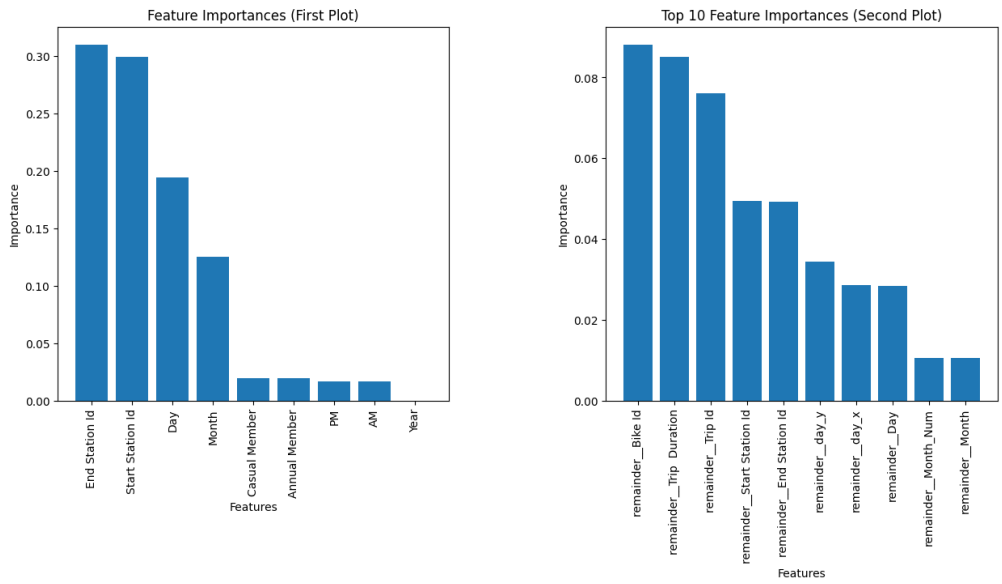


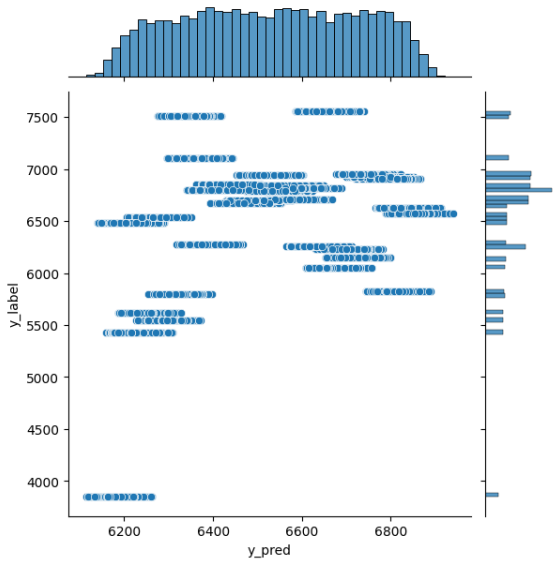
Figure 5.9: Results of MI approach for FS. The left image shows the importance of features before encoding categorical features, and the right image shows the importance of features after encoding categorical features.

and hyper parameters are refined to strike an optimal balance between predictive accuracy and generalizability.

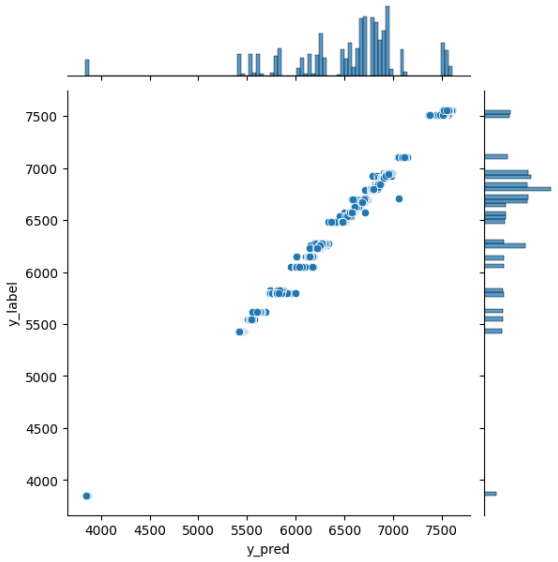
The fine-tuning process involved iterative adjustments to model parameters, with a focus on optimizing the learning rate, tree depth, and feature subsampling to achieve the best predictive performance. Cross-validation techniques ensured robust evaluation across multiple subsets of the data, guarding against overfitting and providing a reliable estimate of each model's performance. The final results, based on RMSE, highlight the superior predictive capabilities of the XGBoost Poisson+ Exposure model, followed by XGBoost Poisson, XGBoost Regression, RF, Linear Poisson, and LR (Table 5.1).

Table 5.1: Comparison between different methods.

Model	Mean RMSE
Linear Regression	598.873
Linear Poisson	437.88
Random Forest	54.567
XGBoost Regression	24.633
XGBoost Poisson	15.734
XGBoost Poisson+Exposure	11.687



(a) The results of LR with (RMSE=598.87334).



(b) The right one shows the results of XGBoost Poisson + Exposure with (RMSE = 11.687).

Figure 5.10: Comparison of results between LR and XGBoost Poisson + Exposure.

5.7 Conclusion

In conclusion, this paper investigated ridership prediction for optimal resource allocation and enhanced user experience in BSS urban transportation networks. Transitioning from visual inspections to numerical analyses, this paper emphasizes the significance of feature engineering in refining data for subsequent modeling stages. The fine-tuning of model parameters, guided by iterative adjustments and cross-validation, ensures an optimal balance between predictive accuracy and generalizability. The results, based on RMSE, rank models in order of effectiveness, with XGBoost Poisson+ Exposure leading the way, followed by XGBoost Poisson, XGBoost Regression, RF, Linear Poisson, and LR. These findings underscore the potential of predictive models offering valuable insights for sustainable transportation management. In the future work we will publish the results of our model by comparing different granularity influenced by factors such as weather conditions, weekdays, and special events.

Chapter 6

A Model Agnostic Method for Feature Selection on Quantum Computing

6.1 Preface

This chapter describes the fifth sub-objective, which is to introduce a novel approach leveraging QA to enhance FS processes. FS is a pivotal task in ML, aimed at identifying the most relevant variables for building predictive models. By reducing the dimensionality of data, we can improve model performance and reduce computational costs. The method proposed in this research uses the power of QA to navigate the complex search space of FS more effectively than traditional methods. The development of this QC-based FS method involved experimentation and validation across various datasets, including wine quality, insurance, and BSS data. The main contribution of this paper is to introduce a novel model agnostic algorithm for FS in regression models that, in most cases, outperforms current state-of-the-art approaches in QA. To validate the FS results, we compare R-squared and RMSE across three target models: 1) LR, 2) RF, and 3) XGBoost. The effectiveness of the proposed method is evaluated using various datasets, demonstrating its versatility and robustness. The results demonstrate the method's superiority in achieving higher predictive accuracy with fewer features, thus simplifying models and enhancing computational efficiency. The content

of this chapter is submitted for publication in Quantum Machine Intelligence Journal as follows: **A. Nourbakhsh, S. Sheikh Gargar, M. Jadidi**. "A Model Agnostic Method for Feature Selection on Quantum Computing." *Quantum Machine Intelligence Journal, Submitted May 2024*.

6.1.1 Copyright and Licensing

This will be added once the paper is published.

6.1.2 Contributions

The roles of the authors in this chapter are outlined as follows: Amirhossein Nourbakhshrezaei was responsible for the literature review, data collection and organization, methodological development, implementation for analysis and modeling, result validation and visualization, and the initial manuscript writing. Soroush Sheikh Gargar was responsible for the methodological development, result validation, and visualization. Mojgan Jadidi oversaw the research, provided the funding, and contributed to the methodology development, as well as the writing and editing of the manuscript.

6.2 Abstract

This study introduces a novel model-agnostic FS method leveraging Quantum Computing (QC) principles, specifically QA. FS is a crucial data pre-processing step in ML that aims to identify the most relevant features for predictive models, enhancing their performance and efficiency. The proposed method demonstrates its effectiveness by outperforming traditional MI methods and current state-of-the-art techniques in QC for FS. Using three datasets; Wine Quality, Insurance, and BSS, we evaluated the proposed method against three different target models: LR, RF, and XGBoost. Our results show that the proposed method consistently achieves higher (R^2) values and lower RMSE scores in most cases compared to traditional methods. The study highlights the potential of integrating QA for FS in enhancing predictive accuracy and computational efficiency. The proposed method's ability to achieve higher predictive accuracy with fewer features simplifies models and reduces computational costs, making it valuable for large-scale data mining and ML tasks. Future work will focus on refining the proposed method and exploring its applicability to real-time data processing scenarios.

6.3 Introduction

Feature selection is a critical data preprocessing strategy that tries to identify the most relevant features for predictive models. In many real-world scenarios, datasets contain a large number of features, only a few of which may be directly related to the target variable. The process of selecting relevant features can lead to significant improvements in data mining and ML [203]. Moreover, FS simplifies models by removing irrelevant or redundant features, leading to faster training and less memory usage.

In supervised learning, FS methods can be divided into three main categories: wrappers [204], filter-based methods [205], and embedded methods [206]. Wrapper methods, such as forward and backward selection, iteratively evaluate feature subsets by training and testing a predictive model [207]. Although they provide accurate results, they can be computationally expensive. Filter-based

methods select features based on their statistical relationship with the target variable without involving a predictive model. Examples include MI [208], Pearson similarity, and Spearman similarity, which assess the dependence or correlation between features and the target [209]. These methods are computationally efficient but may lack precision compared to wrappers. Embedded methods integrate FS within the training process of a predictive model. Least Absolute Shrinkage and Selection Operator (LASSO) [210], for example, uses L1 regularization to shrink less important feature coefficients to zero, effectively selecting relevant features.

In practice, solving FS through brute force approaches is computationally expensive, especially when handling high-dimensional data. Evaluating all possible feature combinations quickly becomes impractical due to the combinatorial explosion. As a result, solving them by heuristic approaches are more efficient.

One promising heuristic approach is QA [170], a well-known hybrid method in QC. QA leverages principles of quantum mechanics to approximate global optimization problems efficiently. This makes it particularly suitable for FS, as it can navigate the search space more effectively, balancing computational efficiency and solution quality.

The main contribution of this paper is to introduce a novel model agnostic algorithm for FS in regression models that, in most cases, outperforms current state-of-the-art approaches in QA. To validate the FS results, we compare R-squared and RMSE across three target models: 1) LR, 2) RF, and 3) XGBoost. The effectiveness of the proposed method is evaluated using various datasets, demonstrating its versatility and robustness.

The implementation of our algorithms and all associated code are available for public access at Google Colab Link.

The rest of this paper is structured as follows: Section 6.4 reviews the relevant literature and existing approaches to FS in QC. Section 6.5 presents the methodology behind our proposed algorithm. Section 6.6 outlines the experiments conducted to evaluate the algorithm, while Section 6.7 discusses the results obtained and compares them to the state-of-the-art methods. Finally, Section 6.8 concludes the paper and offers insights into potential future work.

6.4 Literature Review

Recent years have seen significant progress in research activity around FS and more specifically FS using QC, as evidenced by the growing number of publications. Figure 6.1 illustrates the increasing trend of papers focusing on FS both with and without using QC.

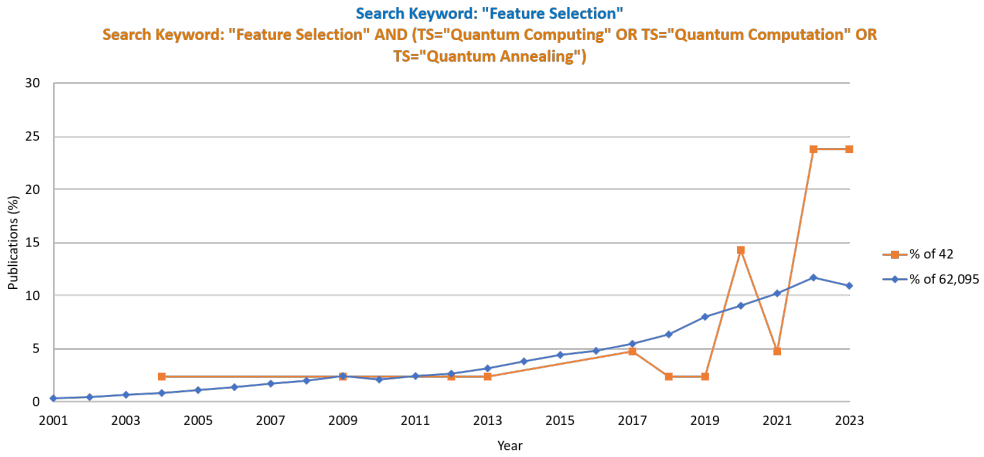


Figure 6.1: Results of the query in Web Of Science (WOS) that shows the percentage of publications per year for different keywords.

Many researchers have been working on FS using QC. However, the idea of using QA for FS first emerged in 2017 when Milne et al. [94] introduced a method based on Quadratic Unconstrained Binary Optimization (QUBO) to formulate the FS problem. They applied this method to credit scoring and classification. The QUBO model aimed to identify features that were both independent and influential, achieving a smaller feature subset without compromising accuracy compared to RFE. This highlights the potential of QA in high-dimensional data mining tasks. Nembrini et al.[95] investigated and designed a hybrid FS algorithm for recommender systems using QA. They represent FS as an optimization problem and solve it using D-Wave's quantum computer. Their results demonstrate that this approach is effective in identifying a limited set of critical features, showcasing the growing potential of QC in practical scientific applications. Hellstern et al. [96] explored the application of FS in QC for supervised learning. They formulated QUBO and solved it using both classical and quantum methods, including QA and QAOA. The authors compared QUBO-based FS to standard methods like RFE and LASSO. Mücke et al. [97] proposed a novel

FS algorithm based on QUBO. This algorithm identifies a specified number of features by optimizing both feature importance and redundancy. The authors solved this QUBO problem using classical computers, VQE, and QA. Experimental results on benchmark datasets show that their method is competitive with standard FS methods. Vlasic et al. [98] introduced a QFS algorithm leveraging QA. They used information-theoretic metrics; MI and Spearman correlation coefficients. When compared to classical FS methods such as the ANOVA F-test and Chi-squared statistics, QFS consistently delivered superior model stability and predictive performance across three datasets. Sabbar et al.[99] propose a QIGA to enhance AMC. The QIGA model leverages QC principles to improve FS and classification accuracy. By extracting enhanced cumulants from modulation signals and optimizing the search process, the QIGA algorithm achieves superior performance. This innovative approach demonstrates the potential of quantum-inspired optimization in improving AMC accuracy and efficiency. Other related works can be found here [100, 101, 102, 103, 104, 105].

6.5 Methodology

In this section, the model agnostic method and the way that it can be formulated to solve as a QUBO problem on a quantum computer are described. In FS, for each feature f belonging to the set of features F, and a target t, it is important to select features that have the strongest correlation with t. To evaluate the proposed method, six functions are investigated, and the evaluation method is applied using three target methods.

6.5.1 Quantum Annealing

To formulate the model, first QC model should be selected. Given the rapidly advancing quantum hardware technology, there is a strong incentive to analyze and tackle current NP-hard problems, which remain computationally challenging even for the most powerful classical computers [181]. Different QC models utilize distinct technologies, resulting in each model having a variety number of qubits and levels of noise due to the entanglement of many qubits. Among the existing QC mod-

els, QA has shown promising results in solving NP-hard problems, making it the chosen method in this research.

The Hamiltonian function of a QC device links specific states, known as eigenstates, to their respective energies [182]. When the system is in an eigenstate of the Hamiltonian, the energy is well-defined and referred to as the eigenenergy. All other states have uncertain energies. The eigenspectrum is the set of eigenstates with known eigenenergies, as shown in Eq. 6.1:

$$H_{Ising}(s) = -\frac{A(s)}{2}\sum_i \hat{\sigma_x}^{(i)} + \frac{B(s)}{2}\left(\sum_i h_i \hat{\sigma_x}^{(i)} + \sum_{i>j} J_{i,j}\hat{\sigma_z}^{(i)}\hat{\sigma_z}^{(j)}\right) \tag{6.1}$$

where $J_{i,j}$ represents the coupling strength between qubits i and j, h_i is the bias for qubit i, and $\hat{\sigma^{(i)}}$ are the Pauli matrices. The equation comprises two terms; the initial Hamiltonian (first term) and the final Hamiltonian (second term). The first term represents the lowest-energy state of the superposition of qubits (0, 1) and is known as the tunneling Hamiltonian. The second term represents the lowest-energy state of the final states and is the solution to the problem at hand. It incorporates the biases (σ) and coupling between qubits in the classical state and is known as the Hamiltonian problem.

$$f(x) = \sum_i Q_{i,i}x_i + \sum_{i<j} Q_{i,j}x_i x_j \tag{6.2}$$

where $Q_{i,i}$: Linear coefficients

$Q_{i,j}$: Quadratic coefficients, or non-zero off-diagonal terms

$$E_{Ising}(s) = \sum_{i=1} h_i s_i + \sum_{i=1}\sum_{j=i+1} J_{i,j}s_i s_j \tag{6.3}$$

where h_i: Linear coefficients

$J_{i,j}$: Quadratic coefficients, or non-zero off-diagonal terms

To execute the model on D-Wave QA computers, it needs to be reformulated as either a Constrained Quadratic Model (CQM), Binary Quadratic Model (BQM), or Discrete Quadratic Model (DQM). In this research, as a proof of concept and to implement our method, we formulated the problem using a hybrid approach that works with BQM, given that the variables are binary and ensures that each qubit is either selected or not selected to predict the target, with a constraint to set total number of features to a specific number. The model can be executed as a QUBO (Eq. 6.2) or an Ising model (Eq. 6.3). More specifically, the Ising model serves as input for the quantum hardware. However, due to the nature of QUBO (with parameters 0 and 1 instead of -1 and +1 like in the Ising model), formulating problems using QUBO is generally simpler. Nevertheless, both models can be converted into each other. To map the problem to either the Chimera graph or Pegasus graph, the model should first be formulated as a QUBO, and then the Ising model will be passed to the Quantum Processing Unit (QPU) in D-Wave.

6.5.2 Proposed Method

To define biases and couplers in Equation 6.2, we need to define our problem properly. If F features are available, for each target, F qubits should be initialized. If feature i is selected then $q_i = 1$, and $i = 1, ..., F$. In other words, the sum of all possibilities for features equals to$k$ (Eq. 6.4).

$$\sum_{i=1}^{F} x_i = k \tag{6.4}$$

where F = number of features,

x_i = qubit,

k = number of selected features.

To effectively select a subset of features, it is crucial to maximize the correlation between f_i and the t, while minimizing the correlation between the f_i and f_j to reduce redundancy and noise. Let

C represent a function that computes the correlations between variables. Specifically, Equation 6.5 illustrates the first term in the cost function that is related to the correlation between f_i and the t, whereas Equation 6.6 details the correlations among the features themselves. In our proposed method, we specify two different functions for C, one for $C_{f_i,t}$, and one for $C_{(f_i,f_j),t}$. The functions are defined in the next subsections.

$$\sum_{i=1}^{F} x_i.C_{i,t} \tag{6.5}$$

where $C_{i,t}$ = correlation between f_i and t

$$\sum_{i=1}^{F} \sum_{j=1,j\neq i}^{F} x_i x_j.C_{(i,j),t} \tag{6.6}$$

where $C_{(i,j),t}$ = correlation between f_i, f_j and t

Defining Linear Coefficient

In the proposed method, the linear coefficient $C_{i,t}$ within the cost function encapsulates both linear and non-linear associations between f_i and t. Additionally, it incorporates a weighting factor for each f_i, which may assume both negative and positive values. This weighting is designed to amplify the significance of each feature in the model, thereby enhancing the model's sensitivity to particularly influential features (Equation 6.7).

$$C_{f_i,t} = -\frac{MI_{f_i,t}}{R^2_{f_i,t}} \tag{6.7}$$

where $MI_{f_i,t}$ = Mutual Information between f_i and t,

$R^2_{f_i,t} = R^2$ for predicted t using feature i.

Defining $MI_{f_i,t}$ as Equation 6.8, and $R^2_{i,t}$ as Equation 6.9 the $C_{f_i,t}$ will be defined as Equation 6.10.

$$MI_{f_i,t} = -\sum_{f_i \in \mathcal{X}} p(f_i)\log p(f_i) - \sum_{t \in \mathcal{Y}} p(t)\log p(t) - \sum_{f_i \in \mathcal{X}}\sum_{t \in \mathcal{Y}} p(f_i,t)\log p(f_i,t) \tag{6.8}$$

where $p(f_i)$ = probability that f_i equals a specific value from X,

$p(t)$ = probability that the target variable t equals a specific value from Y.

$p(f_i,t)$ =joint probability of f_i and t.

$$R^2(f_i,t) = 1 - \frac{\sum_{s=1}^{S}(t_s - \hat{t}_s)^2}{\sum_{s=1}^{S}(t_s - \bar{t})^2} \tag{6.9}$$

where S = Total number of samples,

t_s = the actual values of the target for sample s,

$\hat{t}_s$ = the predicted values of the target using the feature f_i,

$\bar{t}$ = the mean of the actual values of the target

$$C_{f_i,t} = \frac{[\sum_{f_i \in \mathcal{X}} p(f_i)\log p(f_i) + \sum_{t \in \mathcal{Y}} p(t)\log p(t) - \sum_{f_i \in \mathcal{X}}\sum_{t \in \mathcal{Y}} p(f_i,t)\log p(f_i,t)][\sum_{s=1}^{S}(t_s-\bar{t})^2]}{\sum_{s=1}^{S}(t_s-\bar{t})^2 - \sum_{s=1}^{S}(t_s-\hat{t}_s)^2} \tag{6.10}$$

R^2 ranges from -1 to +1. In the context of the formula specified in Equation 6.7, a negative R^2 value indicates that the feature f_i provides poor predictive capability for the target variable. This is because a negative R^2 suggests that the model's predictions are worse than a simple baseline model that predicts the mean of the target. Consequently, within the framework where the objective is to minimize $C_{f_i,t}$, a negative R^2 value contributes positively to the computation of $C_{f_i,t}$, resulting in a higher positive value. This higher value of $C_{f_i,t}$ leads to the non-selection of the feature due to its detrimental impact on the model's performance. Conversely, a positive R^2 value, indicating better predictive performance, results in a lower value of $C_{f_i,t}$ in our minimization objective, leads to the selection of the feature.

Defining Quadratic Coefficient

For the second part of the cost function which is a quadratic term, we proposed another model-agnostic method that is defined based R^2 and *RMSE* in two different scenarios (Equation 6.11).

$$C_{(f_i,f_j),t} = \begin{cases} \frac{R^2_{(ij),t} - R^2_{i,t}}{RMSE_{(ij),t}} & \text{if } (R^2_{(ij),t} - R^2_{i,t}) \leq 0, \\ \\ (R^2_{(ij),t} - R^2_{i,t}).RMSE_{(ij),t} & \text{otherwise.} \end{cases} \tag{6.11}$$

where $R^2_{(ij),t} = R^2$ of features i, j predicting t ,

$R^2_{i,t} = R^2$ of feature i predicting t,

$RMSE_{(ij),t}$ $RMSE$ of features i, j predicting t.

For calculating R^2 and $RMSE$, LR, RF, and XGBoost are used. Once the linear coefficients ($C_{f_i,t}$) and quadratic coefficients ($C_{(f_i,j),t}$) are defined, we can construct the Q matrix in Equation 6.2.

6.6 Experiments

To rigorously evaluate the proposed method against established state-of-the-art techniques, the identical problem was addressed using both traditional MI on classical computers and QC models. The metrics R^2 and RMSE were computed utilizing three distinct predictive models: LR, RF, and XGBoost. These models were also employed to assess the efficacy of the selected features in predicting the target variable. Comprehensive investigations of all possible combinations of C function and target models were conducted and presented in Figure 6.2.

6.6.1 Datasets

The proposed model has been tested on three different datasets.

Wine Quality Dataset

The wine quality dataset [211] is a popular resource derived from the UCI ML Repository, specifically curated to facilitate the predictive modeling of wine quality based on physicochemical tests. It consists of 1599 observations and 11 features.

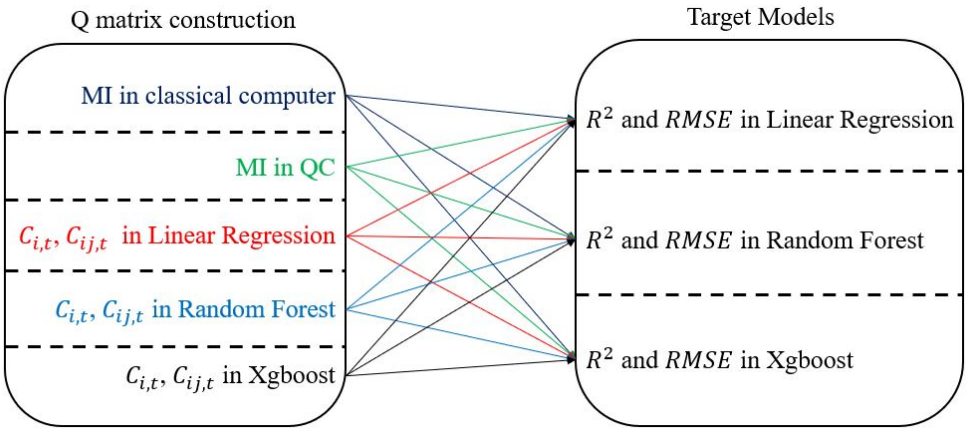


Figure 6.2: Left side: list of correlation/similarity methods. Right side: list of target models to evaluate the results.

Insurance Dataset

The insurance dataset [212] encompasses various attributes of insurance beneficiaries and is particularly designed for projects that involve predictive modeling of medical costs. The primary objective when analyzing this dataset is to develop predictive models that can accurately estimate individual medical costs billed by health insurance. This involves exploring regression tasks, assessing the influence of demographic and behavioral factors on health costs, and potentially guiding insurance cost strategies.

Bike Sharing System Dataset

This dataset [213] pertains to the BSS in Toronto and captures comprehensive ridership data over a two-month period. Initially containing 352,911 records and 10 features, the dataset provides detailed insights into the usage patterns of the city's BSS. To augment the dataset's utility and depth, it has been enriched with additional socio-demographic data extracted from the Toronto Census dataset [214]. This enrichment involved integrating geospatial characteristics of the bike stations based on their locations within specific regions of Toronto. The census data, sourced from the Toronto ward profiles, provides extensive demographic and statistical information for each ward where bike stations are located. After the process of feature engineering and the inclusion of census

data, the enriched dataset now comprises 283,919 records and 2,689 features. This significant increase in features reflects the integration of detailed socio-demographic variables, which offer a richer, multidimensional view of the dataset. These enhancements allow for more granular analyses and the ability to explore complex interactions between BSS usage patterns and socio-economic factors.

6.6.2 Configuration of QUBO

The quantum solver function is central to implementing the QUBO. In this research, D-Wave's QA solver is used to solve the proposed method. The function takes three parameters; Binary Quadratic Model (BQM); binary variables, the list of features from the dataset; and the number of features to be selected. Initially, the function interacts with D-Wave's Quantum Processing Unit (QPU) through the *DWaveSampler* to access the quantum device. It converts the QPU's architecture into a graph representation using networkx functions. This graph assists in finding an appropriate clique embedding for the variables of BQM. The embedding maps the high-dimensional problem into the physical layout of the quantum processor. Once the embedding is determined, a fixed embedding composite is created, which facilitates the direct use of the QPU by integrating the embedding within the sampling process. The function then prints the maximum chain length used in the embedding, providing an insight into the complexity and depth of the minor embedding required for the problem. The quantum sampling is designed to optimize hybrid solutions that balance both classical and quantum computations. The function enters a loop, iterating over the desired number of features to select, adjusting and updating the BQM with combination constraints. This ensures that the exact number of features is selected. The QPU read is set to 100, and the maximum iteration is set to 3.

6.7 Results and Discussion

Figure 6.3 shows the R^2 values by selected features for the wine dataset using the traditional MI method. The x-axis represents the number of selected features, and the y-axis represents the R^2 values that is calculated using different target methods; LR, XGBoost, and RF. In the traditional method, the best R^2 was achieved using 9 features, and it equals 0.501251 ($RMSE = 0.5709$). Figure 6.4 depicts the R^2 values by selected features for the wine dataset using the MI in QC in the current state of the art. The best R^2 for MI in QC was achieved using 9 features where $R^2 = 0.5137$ and $RMSE = 0.563691$.

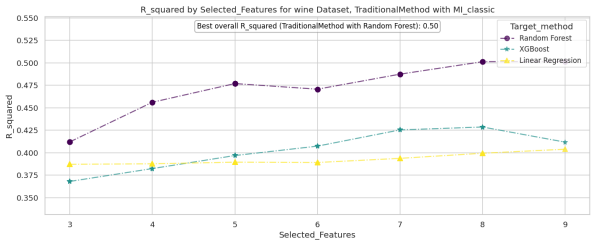


Figure 6.3: Results of FS, using traditional MI for wine dataset. R^2 is calculated using different target methods.

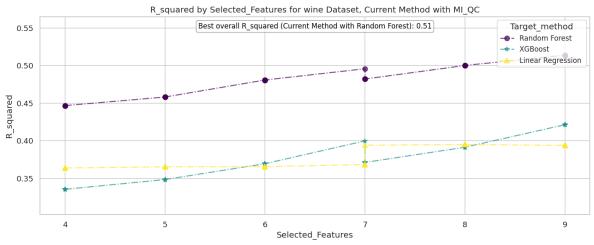


Figure 6.4: Results of FS, using existing MI in QC for wine dataset. R^2 is calculated using different target methods.

Figure 6.5 shows the results of the proposed method where both R^2 and $RMSE$ are calculated using LR. In this scenario, $R^2 = 0.50201$ and $RMSE = 0.570469$ for 9 selected features was the best solution. In another scenario where to calculate R^2 and $RMSE$ for the proposed method, RF has been used, the results are shown in Figure 6.6. In this case, $R^2 = 0.51377$ and $RMSE = 0.56369$ for 7 selected features.

Figure 6.7 shows the results of the proposed method where both R^2 and $RMSE$ are calculated using XGBoost. The best values obtained are $R^2 = 0.50889$ and $RMSE = 0.56651$ for 7 selected features. Figure 6.8 represents a comparison between different similarity methods for FS. Using 7 features, the proposed method outperformed both traditional MI and MI in QC. With 9 features, the proposed method outperformed traditional MI and showed a slight difference under MI in QC.

It is important to note that the goal of FS is to choose a subset of features that can achieve acceptable accuracy with fewer features. Models that maintain high accuracy with a smaller number of features

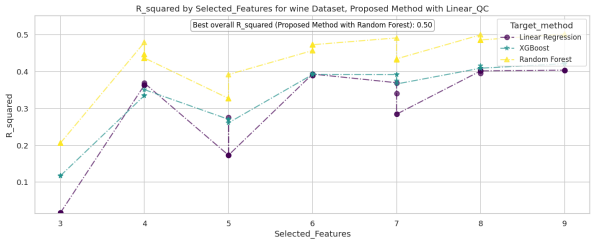


Figure 6.5: Results of FS, using the proposed method. R^2 and *RMSE* are calculated using LR for the wine dataset. The final R^2 is calculated using different target methods.

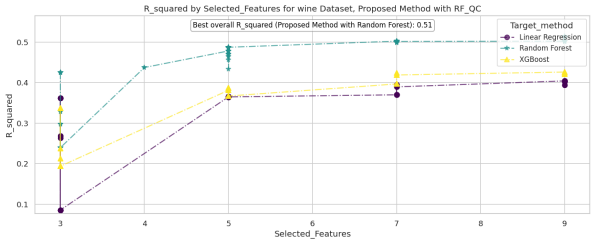


Figure 6.6: Results of FS, using the proposed method. R^2 and *RMSE* are calculated using RF for the wine dataset. The final R^2 is calculated using different target methods.

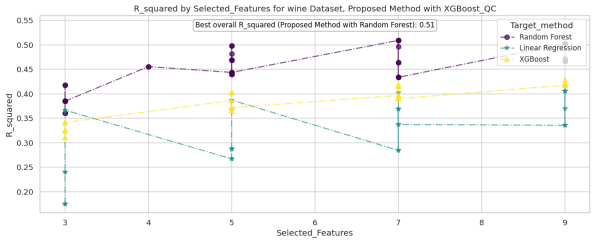


Figure 6.7: Results of FS, using the proposed method. R^2 and *RMSE* are calculated using XGBoost for the wine dataset. The final R^2 is calculated using different target methods.

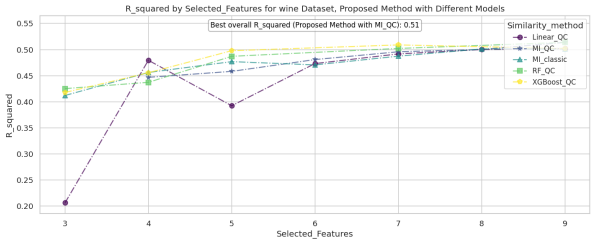


Figure 6.8: Comparing the results of FS with different methods. The R^2 is grouped by similarity function for each number of features.

are preferred. The trend lines in the figures indicate that the proposed method achieves a higher R^2 value with fewer selected features. This demonstrates the efficiency of the proposed method in identifying the most relevant features, thus reducing the complexity and potentially improving the model's performance.

The table 6.1 presents the R^2 values for FS using traditional and proposed methods. It compares the performance of different methods across varying numbers of selected features. The traditional MI method, and MI in QC, show a gradual improvement in R^2 as the number of features increases, with *MI_classic* starting at 0.4119 for 3 features and reaching 0.5013 for 9 features, while *MI_QC* starts at 0.4467 for 4 features and reaches 0.5138 for 9 features. The proposed methods, generally exhibit higher R^2 values. Notably, the proposed method achieves the highest R^2 of 0.5089 for 7 features. Moreover, in most cases it demonstrates competitive performance, particularly with 5 and 7 features, indicating the effectiveness of the proposed methods in enhancing model performance through FS. The results highlight that the proposed methods, especially, are capable of achieving

better model accuracy compared to traditional methods.

Table 6.1: R^2 of the FS for traditional and proposed methods. Empty values indicate that the QC was unable to find an embedding for the problem.

			Proposed Method		
Features	MI_classic	MI_QC	Linear_QC	RF_QC	XGBoost_QC
3	0.4119	-	0.2062	**0.4251**	0.4177
4	0.4560	0.4467	**0.4794**	0.4372	0.4555
5	0.4768	0.4582	0.3922	0.4871	**0.4978**
6	0.4706	**0.4808**	0.4731	**0.4881**	0.4505
7	0.4873	0.4958	0.4916	**0.5021**	**0.5089**
8	0.5012	0.5001	0.5000	**0.5157**	-
9	0.5013	**0.5138**	0.5020	**0.5138**	0.5020

Table 6.2 displays the *RMSE* values for FS across various methods and different numbers of selected features. Consistent with the R^2 results, the traditional MI method (*MI_classic*) and MI in QC (*MI_QC*) show a reduction in *RMSE* as the number of features increases. Specifically, *MI_classic* starts at 0.6199 for 3 features and decreases to 0.5709 for 9 features, while *MI_QC* begins at 0.6013 for 4 features and drops to 0.5637 for 9 features. The proposed methods demonstrate generally lower *RMSE* values, with the lowest *RMSE* of 0.5665 achieved for 7 features.

Table 6.2: *RMSE* of the FS for traditional and proposed methods. Empty values indicate that the QC was unable to find an embedding for the problem.

			Proposed Method		
Features	MI_classic	MI_QC	Linear_QC	RF_QC	XGBoost_QC
3	0.6199	-	0.7202	**0.6130**	0.6169
4	0.5962	0.6013	**0.5833**	0.6065	0.5965
5	0.5847	0.5951	0.6302	0.5790	**0.5729**
6	0.5882	0.5825	0.5868	**0.5784**	0.5993
7	0.5788	0.5740	0.5764	0.5704	**0.5665**
8	0.5709	0.5715	0.5716	**0.5626**	-
9	0.5709	**0.5637**	0.5705	**0.5637**	0.5705

Figure 6.9 illustrates the R^2 values for the insurance dataset using the traditional MI method with different FS scenarios. The x-axis shows the number of selected features (ranging from 3 to 6), while the y-axis represents the R^2 values. The highest R^2 value is 0.8836416 and the best *RMSE*

is 4250.233107 for 6 selected features. Moreover, Figure 6.10 represents the results of FS using MI in QC. The best overall score based on the target models are as follows; $R^2 = 0.8836416$ and $RMSE = 4250.233107$ for 6 selected features, which are the same as traditional MI method.

Figure 6.9: Results of FS, using traditional MI for the insurance dataset. R^2 is calculated using different target methods.

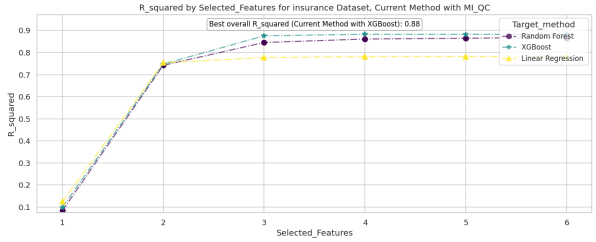


Figure 6.10: Results of FS, using existing MI in QC for the insurance dataset. R^2 is calculated using different target methods.

Figure 6.11 however, shows the results of proposed method for the insurance dataset using LR to calculate R^2 and $RMSE$ that are used in the proposed method. The results of this scenario show an early convergence similar to MI in QC with only 4 selected features. The R^2 equals 0.88286 and $RMSE$ is 4264.417877. Figure 6.12 displays another scenario for the proposed method where R^2 and $RMSE$ are calculated using RF. As shown, the trend converges earlier compared to the traditional MI method, achieving similar performance with only 4 selected features. In this scenario, $R^2 = 0.88321$ and $RMSE = 4258.088$.

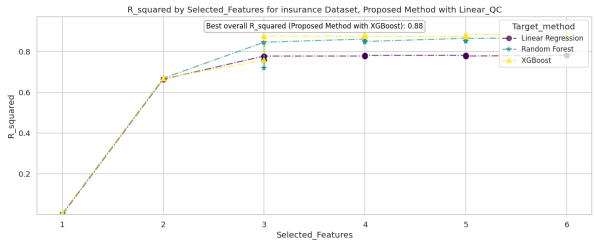


Figure 6.11: Results of FS, using the proposed method. R^2 and $RMSE$ are calculated using LR for the insurance dataset. The final R^2 is calculated using different target methods.

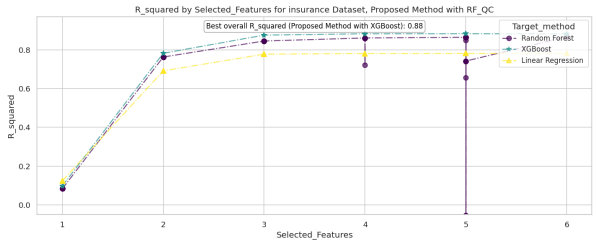


Figure 6.12: Results of FS, using the proposed method. R^2 and $RMSE$ are calculated using RF for the insurance dataset. The final R^2 is calculated using different target methods.

Analysing the results of the proposed method where the variables are calculated using XGBoost, shows the same performance. As Figure 6.13 shows, the R^2 are converged after 3 selected features. There is a displacement in the chart for 3 features which shows the impact of alpha. The charts for

showing the impact of alpha are given in the Appendix section. Figure 6.14 visualizes a comparison between different methods used for FS with both proposed method, traditional MI, and MI in QC.

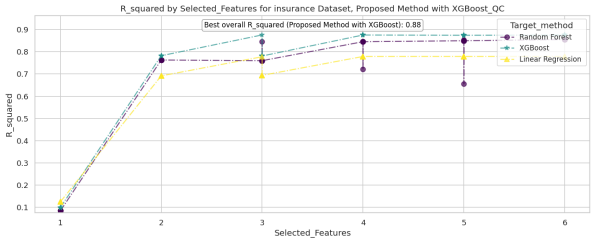


Figure 6.13: Results of FS, using the proposed method. R^2 and *RMSE* are calculated using XGBoost for the insurance dataset. The final R^2 is calculated using different target methods.

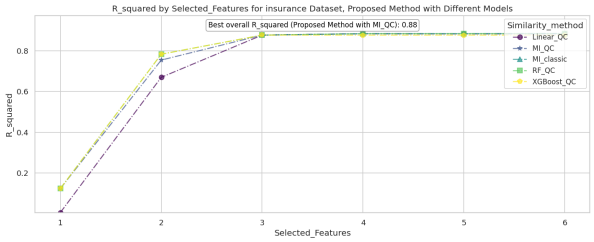


Figure 6.14: Comparing the results of FS with different methods on the insurance dataset. The R^2 is grouped by similarity function for each number of features.

Figures [6.15,6.16,6.17,6.18,6.19,6.20,] illustrate the performance of different similarity methods in terms of R^2 values. The traditional MI method struggles to achieve satisfactory results, with the best solution obtained yielding an R^2 value of only 0.22942. This indicates that the traditional MI method is less effective in capturing the relationship between the features and the target variable. On the other hand, QC approaches demonstrate promising results, achieving an R^2 value of 0.55686. In the proposed methods and also in MI in QC, results outperform traditional methods. This indicates that both linear and non-linear correlation between the features and the target variable are identified. Moreover, using non-linear models in the proposed method like RF and XGBoost can capture more effectively than LR. However, it is noteworthy that using LR to determine the required parameters in the proposed method still outperforms the traditional MI method.

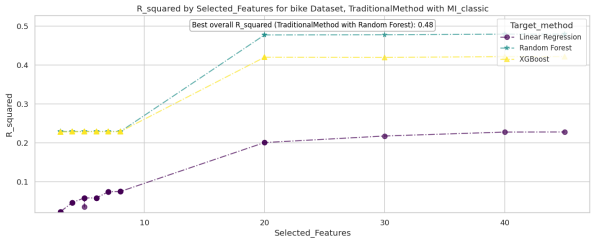


Figure 6.15: Results of FS, using traditional MI for the bike dataset. R^2 is calculated using different target methods.

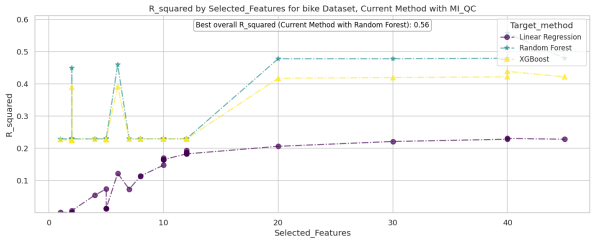


Figure 6.16: Results of FS, using existing MI in QC for the bike dataset. R^2 is calculated using different target methods.

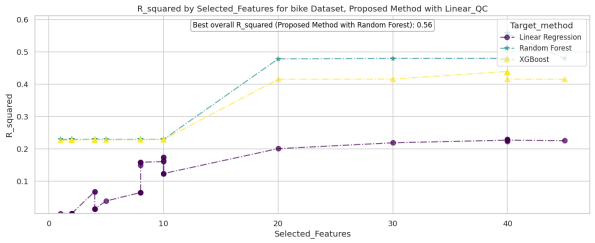


Figure 6.17: Results of FS, using the proposed method. R^2 and *RMSE* are calculated using LR for the bike dataset. The final R^2 is calculated using different target methods.

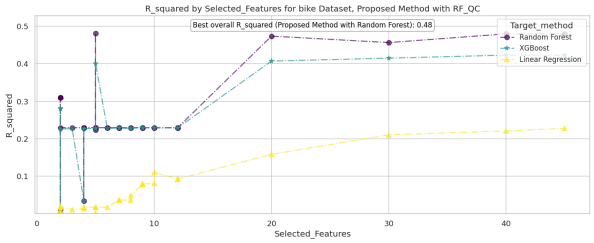


Figure 6.18: Results of FS, using the proposed method. R^2 and *RMSE* are calculated using RF for the bike dataset. The final R^2 is calculated using different target methods.

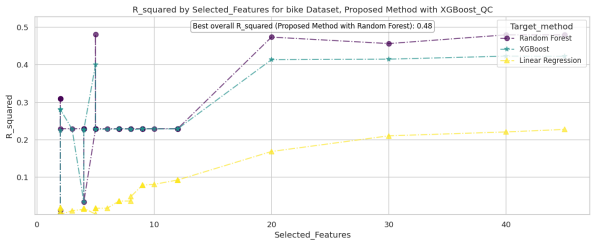


Figure 6.19: Results of FS, using the proposed method. R^2 and *RMSE* are calculated using XGBoost for the bike dataset. The final R^2 is calculated using different target methods.

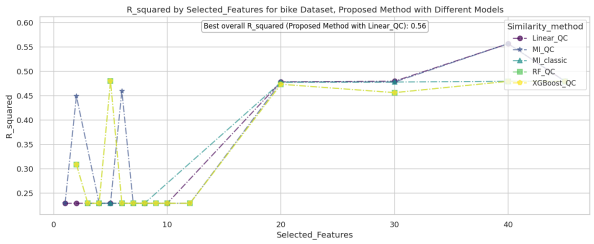


Figure 6.20: Comparing the results of FS with different methods on the bike dataset. The R^2 is grouped by similarity function for each number of features.

6.8 Conclusion

In this study, we presented a novel model-agnostic FS method leveraging QC principles, specifically QA, to enhance the predictive performance of regression models. Our approach was tested against three datasets; Wine Quality, Insurance, and BSS using three different target models: LR, RF, and XGBoost. The results demonstrate the efficacy of our proposed method. Compared to traditional MI methods, our approach consistently achieved higher R^2 values and lower RMSE scores. For instance, in the BSS, while the traditional MI method struggled with few number of features, our proposed method achieved promising results, indicating a significant improvement. This suggests that QC approaches, especially those incorporating non-linear models like RF and XGBoost, can better capture the intricate relationships between features and the target variable. Moreover, the results show competitive performance compared to the MI in QC methods in the state of the art. The ability of our proposed method to achieve higher predictive accuracy with fewer features

is particularly noteworthy. This not only simplifies the models but also enhances computational efficiency, making it a valuable method for large-scale data mining and ML tasks. In conclusion, the integration of QA for FS proves to be a robust and effective approach, outperforming traditional methods. This research opens avenues for further exploration of QC applications in ML, with potential expansions into other types of predictive models and more complex models. As future work, the focus will be on refining the proposed method and exploring its applicability to real-time data processing scenarios, thereby advancing the frontier of QML.

Chapter 7

Conclusion

7.1 Summary of Research

This research tackles several key challenges in transportation management using QC. Most transportation issues can be formulated as discrete models that are complex, involving various interrelated components and stakeholders. These models must function in dynamic environments with fluctuating demand, evolving infrastructure, and external factors like weather and geography. This study explores and investigates solutions to these problems by examining different aspects of transportation management using QC. The following sections detail the problem, proposed solutions, and results for each objective 1.4 of this research.

7.2 Final Remarks and Contributions

7.2.1 Explore The Potential Of QC In Transportation Applications

The first objective1.4.1 involves formulating the risk of exposure to viruses, such as COVID-19, during city journeys by optimizing the VRP. By transforming this into QUBO problem and solving it with QA, the research aims to minimize virus exposure for road users, leveraging data from Beijing's taxi services. The results demonstrate that the QPU is significantly faster for small problems,

although it has limitations due to its architecture. Hybrid QC emerges as a reliable and scalable solution, capable of finding better solutions (lower energy) and handling more complex problems by dividing them into smaller batches.

7.2.2 Develop a QC-based Approach For Dynamic Rebalancing BSS

The second objective1.4.2 focuses on optimizing the rebalancing of BSS to ensure bike availability and meet user demand. By framing the problem as a QUBO problem and utilizing QC, specifically the Advantage system 4.1 on the QPU, the research demonstrates a significant reduction in the occupancy index, indicating improved bike distribution efficiency in Toronto's BSS.

7.2.3 Enhance BSS Management By Clustering Stations Using QC

To achieve the third objective1.4.3, the potential of QC in ML is explored, particularly in clustering BSS stations. The hybrid model developed addresses clustering as a CSP on a QA computer. Using real-time data from Toronto's BSS, the research evaluates various clustering methods, demonstrating the efficiency of QC in solving complex clustering problems and enhancing the management of BSS.

7.2.4 Predict BSS Ridership To Enhance System Management

To achieve the fourth objective1.4.4, different modeling approaches to predict BSS ridership are investigated with geospatial features, focusing on Toronto BSS as a case study. Accurate predictions are essential for effective management, resource optimization, and user satisfaction. Techniques such as LR, RF, and XGBoost are analyzed, with findings indicating that XGBoost Poisson regression with exposure provides the most accurate predictions.

7.2.5 Improve The Accuracy Of The Predictive Models

Finally, a model-agnostic FS method leveraging QC principles is proposed1.4.5. This method improves predictive model performance by identifying relevant features, outperforming traditional and state-of-the-art methods in terms of predictive accuracy and computational efficiency. Evaluations using datasets like Wine Quality, Insurance, and BSS show the proposed method's superiority, highlighting its potential for large-scale data mining and ML tasks.

7.3 Limitations

Although we have demonstrated the impact and promising potential of QC in various transportation problems, several limitations remain.

1. The quantum adiabatic theorem states that if one starts from the ground state of the initial Hamiltonian and the transformation is done slowly enough, then the final state of the system is the ground state of the final Hamiltonian. One limitation is that the minimal required time needed to perform the transformation scales is as 1.9.

2. It is concluded that QPU is faster than every other solver. However, QPU is only applicable for small problems, and this limitation is caused by QPU architectures.

3. Hybrid QC requires transferring data between classical and quantum units which affects the overall computational speed.

4. QA is very sensitive to noises and environmental factors which may lead to inaccurate solutions, which is why different iterations are required to validate the solution.

5. The effectiveness of quantum models is dependent on the architecture of the quantum device and its alignment with the problem's topology. If the problem's structure does not match the device's architecture, the required circuit depth increases, leading to performance degradation even with low noise rates.

6. Embedding fully connected graphs, commonly encountered in transportation applications, results in a substantial space overhead. This quadratic space overhead increases the time to solution because it necessitates managing a larger number of variables in the embedded problem.

7. The constraints in this research are defined as squared terms. These terms create fully connected interactions among the qubits, posing difficulties due to the sparse connectivity of the chimera graph. This challenge needs alternative methods to solve these problems efficiently.

8. The restricted dynamic range of coupling strengths and pre-factors in quantum annealers makes it difficult to apply large penalty terms required for certain constraint satisfaction problems, resulting in higher error rates and distorted ground state distributions.

7.4 Future Works

The field of QC is rapidly evolving, with significant advancements continually pushing the boundaries of what is possible. Based on the experience we have gained from this research, we have identified several critical areas for future work that are essential to improving the field of QC in transportation and broader spatial graph-based applications. These include improving error correction methods in QA, optimizing chain strength in minor embedding, developing compilers tailored for QC architectures, exploring gate-based QC models, and creating user-friendly interfaces for interaction with proposed models. Each of these areas addresses fundamental challenges and opportunities that will shape the future of QC, enhancing its reliability, efficiency, and accessibility.

7.4.1 Error Correction in QA

Error correction is a paramount concern in QC, particularly in QA where decoherence and operational errors can significantly affect performance. Future work is required to focus on developing robust error correction techniques specifically tailored for QA processes. This involves design-

ing algorithms that can detect and correct errors in real-time, ensuring the integrity of information throughout the annealing process. By mitigating errors, we can enhance the accuracy and reliability of QA, making them more practical for solving complex optimization problems.

7.4.2 Chain Strength in Minor Embedding

We have used minor embedding in QA to map a problem onto the hardware's physical qubits. The chain strength, which determines the coupling between qubits in a chain, plays a crucial role in the effectiveness of this mapping. In this research, a constant value is used as chain strength. Future research is needed that should aim to optimize the selection of chain strengths to minimize errors and improve solution quality.

7.4.3 Developing a Compiler for QC Architecture

In this research, the Pegasus graph and the Chimera graph are used that have their specific topology. Not all hardware accepts this topology. Migrating from one topology to another topology requires more pre-processing. Designing and implementing compilers that can efficiently map quantum models onto a variety of quantum hardware platforms is required. These compilers will need to account for the unique constraints and capabilities of different quantum devices, enabling more efficient and effective utilization of quantum resources.

7.4.4 Exploring The Gate-based QC Models

In this research, the potential of QC and mostly QA are investigated for transportation applications. However, it is important to explore the potential of other QC models, such as gate-based models. Gate-based models provide precise control over individual qubits and quantum gates, facilitating the implementation of complex quantum operations and error correction protocols. This capability makes them comparable to QA, warranting further investigation to fully understand their advantages and applications.

7.4.5 An Interface For User Interaction

Another important aspect of this research is the accessibility of the proposed methods. Ideally, users should be able to test and run their datasets on these methods and evaluate the results. Therefore, developing an intuitive and user-friendly interface is important. Future work can focus on developing an interface that facilitates interaction with the models, making it easier for users to design, run, and analyze quantum algorithms not only in transportation applications but any other spatial graphs-based applications. This will simplify the complexities of QC, thereby broadening its accessibility to researchers, developers, and industry professionals.

Bibliography

[1] A. Nourbakhsh, M. Jadidi, and K. Shahriari, "Exploring quantum computing potentials in solving a combinatorial optimization problem to minimize exposure to covid-19 during a city journey," *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 43, pp. 419–425, 2022.

[2] T. Litman and D. Burwell, "Issues in sustainable transportation," *International Journal of Global Environmental Issues*, vol. 6, no. 4, pp. 331–347, 2006.

[3] M. Zou, X. M. Chen, H. Yu, Y. Tong, Z. Huang, M. Li, and H. Zou, "dynamic transportation planning and operations: concept, framework and applications in china," *Procedia-Social and Behavioral Sciences*, vol. 96, pp. 2332–2343, 2013.

[4] Y. Zhou, G. Ma, and H. Zhang, "Analysis on the relationship between transportation and economic development," in *Proceedings of the 4th International Conference on Economic Management and Model Engineering, ICEMME 2022, November 18-20, 2022, Nanjing, China*, 2023.

[5] K. T. Geurs, L. La Paix, and S. Van Weperen, "A multi-modal network approach to model public transport accessibility impacts of bicycle-train integration policies," *European transport research review*, vol. 8, pp. 1–15, 2016.

[6] F.-E. MOHTICH, S. BENSIALI, A. IDRI, and M. EL-AYACHI, "Geospatial data integration for intelligent transportation systems: How to leverage geospatial data to improve travel effi-

ciency and convenience through more accurate traffic forecasting in intelligent transportation systems.," *African Journal on Land Policy & Geospatial Sciences*, vol. 6, no. 4, 2023.

[7] J.-G. Lee and M. Kang, "Geospatial big data: challenges and opportunities," *Big Data Research*, vol. 2, no. 2, pp. 74–81, 2015.

[8] S. Y. He, Y.-H. Kuo, and K. K. Sun, "The spatial planning of public electric vehicle charging infrastructure in a high-density city using a contextualised location-allocation model," *Transportation Research Part A: Policy and Practice*, vol. 160, pp. 21–44, 2022.

[9] H. J. Miller and S.-L. Shaw, "Geographic information systems for transportation in the 21st century," *Geography Compass*, vol. 9, no. 4, pp. 180–189, 2015.

[10] C. McAndrews and J. Marcus, "The politics of collective public participation in transportation decision-making," *Transportation Research Part A: Policy and Practice*, vol. 78, pp. 537–550, 2015.

[11] P. Jones and K. Lucas, "The social consequences of transport decision-making: clarifying concepts, synthesising knowledge and assessing implications," *Journal of transport geography*, vol. 21, pp. 4–16, 2012.

[12] A. M. De Souza, C. A. Brennand, R. S. Yokoyama, E. A. Donato, E. R. Madeira, and L. A. Villas, "Traffic management systems: A classification, review, challenges, and future perspectives," *International Journal of Distributed Sensor Networks*, vol. 13, no. 4, p. 1550147716683612, 2017.

[13] M. Ogryzek, D. Adamska-Kmieć, and A. Klimach, "Sustainable transport: an efficient transportation network—case study," *Sustainability*, vol. 12, no. 19, p. 8274, 2020.

[14] A. Ignatov, V. Baskov, T. Ablyazov, A. Aleksandrov, and N. Zhilkina, "Algorithm for optimizing urban routes in traffic congestion," in *Technological Transformation: A New Role For Human, Machines And Management: TT-2020*, pp. 23–38, Springer, 2021.

[15] S. Damart and B. Roy, "The uses of cost–benefit analysis in public transportation decision-making in france," *Transport Policy*, vol. 16, no. 4, pp. 200–212, 2009.

[16] B. Van Wee, K. Geurs, and C. Chorus, "Information, communication, travel behavior and accessibility," *Journal of Transport and Land Use*, vol. 6, no. 3, pp. 1–16, 2013.

[17] A. Sadavare and R. Kulkarni, "A review of application of graph theory for network," *International Journal of Computer Science and Information Technologies*, vol. 3, no. 6, pp. 5296–5300, 2012.

[18] S. Scellato, L. Fortuna, M. Frasca, J. Gómez-Gardenes, and V. Latora, "Traffic optimization in transport networks based on local routing," *The European Physical Journal B*, vol. 73, pp. 303–308, 2010.

[19] H. Bast, D. Delling, A. Goldberg, M. Müller-Hannemann, T. Pajor, P. Sanders, D. Wagner, and R. F. Werneck, "Route planning in transportation networks," *Algorithm engineering: Selected results and surveys*, pp. 19–80, 2016.

[20] Y. Nie, H. Zhang, and D.-H. Lee, "Models and algorithms for the traffic assignment problem with link capacity constraints," *Transportation Research Part B: Methodological*, vol. 38, no. 4, pp. 285–312, 2004.

[21] H. Shang, Y. Liu, H. Huang, R. Guo, *et al.*, "Vehicle scheduling optimization considering the passenger waiting cost," *Journal of Advanced Transportation*, vol. 2019, 2019.

[22] G. C. Cricsan, L. B. Iantovics, and E. Nechita, "Computational intelligence for solving difficult transportation problems," *Procedia Computer Science*, vol. 159, pp. 172–181, 2019.

[23] G. J. Woeginger, "Exact algorithms for np-hard problems: A survey," in *Combinatorial Optimization—Eureka, You Shrink! Papers Dedicated to Jack Edmonds 5th International Workshop Aussois, France, March 5–9, 2001 Revised Papers*, pp. 185–207, Springer, 2003.

[24] F.-T. Lin, C.-Y. Kao, and C.-C. Hsu, "Applying the genetic approach to simulated annealing in solving some np-hard problems," *IEEE Transactions on systems, man, and cybernetics*, vol. 23, no. 6, pp. 1752–1767, 1993.

[25] R. L. Karg and G. L. Thompson, "A heuristic approach to solving travelling salesman problems," *Management science*, vol. 10, no. 2, pp. 225–248, 1964.

[26] L. Fan and C. L. Mumford, "A metaheuristic approach to the urban transit routing problem," *Journal of Heuristics*, vol. 16, pp. 353–372, 2010.

[27] P. Benioff, "The computer as a physical system: A microscopic quantum mechanical hamiltonian model of computers as represented by turing machines," *Journal of statistical physics*, vol. 22, no. 5, pp. 563–591, 1980.

[28] G. Brassard, "Brief history of quantum cryptography: A personal perspective," in *IEEE Information Theory Workshop on Theory and Practice in Information-Theoretic Security, 2005.*, pp. 19–23, IEEE, 2005.

[29] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, 1976.

[30] C. H. Bennett, G. Brassard, C. Crépeau, R. Jozsa, A. Peres, and W. K. Wootters, "Teleporting an unknown quantum state via dual classical and einstein-podolsky-rosen channels," *Physical review letters*, vol. 70, no. 13, p. 1895, 1993.

[31] C. H. Bennett, G. Brassard, S. Popescu, B. Schumacher, J. A. Smolin, and W. K. Wootters, "Purification of noisy entanglement and faithful teleportation via noisy channels," *Physical review letters*, vol. 76, no. 5, p. 722, 1996.

[32] C. H. Bennett, E. Bernstein, G. Brassard, and U. Vazirani, "Strengths and weaknesses of quantum computing," *SIAM journal on Computing*, vol. 26, no. 5, pp. 1510–1523, 1997.

[33] C. H. Bennett, G. Brassard, and A. K. Ekert, "Quantum cryptography," *Scientific American*, vol. 267, no. 4, pp. 50–57, 1992.

[34] R. P. Feynman, "Simulating physics with computers, international journal of theoretical physics," *International journal of theoretical physics*, 1982.

[35] D. Deutsch, "Quantum theory, the church–turing principle and the universal quantum computer," *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, vol. 400, no. 1818, pp. 97–117, 1985.

[36] J. Preskill, "Quantum computing and the entanglement frontier," *arXiv preprint arXiv:1203.5813*, 2012.

[37] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. Brandao, D. A. Buell, *et al.*, "Quantum supremacy using a programmable superconducting processor," *Nature*, vol. 574, no. 7779, pp. 505–510, 2019.

[38] e. a. Edwin Pednault, "Quantum supremacy." IBM website, 2019. URL https://www.ibm.com/blogs/research/2019/10/on-quantum-supremacy.html.

[39] Y. Lin, P. Wang, and M. Ma, "Intelligent transportation system (its): Concept, challenge and opportunity," in *2017 ieee 3rd international conference on big data security on cloud (bigdatasecurity), ieee international conference on high performance and smart computing (hpsc), and ieee international conference on intelligent data and security (ids)*, pp. 167–172, IEEE, 2017.

[40] S. Jayasooriya and Y. Bandara, "Measuring the economic costs of traffic congestion," in *2017 Moratuwa Engineering Research Conference (MERCon)*, pp. 141–146, IEEE, 2017.

[41] G. Titos, H. Lyamani, L. Drinovec, F. Olmo, G. Movcnik, and L. Alados Arboledas, "Evaluation of the impact of transportation changes on air quality," *Atmospheric Environment*, vol. 114, pp. 19–31, 2015.

[42] D. Banister, K. Anderton, D. Bonilla, M. Givoni, and T. Schwanen, "Transportation and the environment," *Annual review of environment and resources*, vol. 36, pp. 247–270, 2011.

[43] M. Natvig and H. Westerheim, "National multimodal travel information–a strategy based on stakeholder involvement and intelligent transportation system architecture," *IET Intelligent Transport Systems*, vol. 1, no. 2, pp. 102–109, 2007.

[44] Z. Tian, J. Zhou, and M. Wang, "Dynamic evolution of demand fluctuation in bike-sharing systems for green travel," *Journal of cleaner production*, vol. 231, pp. 1364–1374, 2019.

[45] H. J. Miller, "Potential contributions of spatial analysis to geographic information systems for transportation (gis-t)," *Geographical Analysis*, vol. 31, no. 4, pp. 373–399, 1999.

[46] A. Nourbakhshrezaei, M. Jadidi, and G. Sohn, "Improving cyclists' safety using intelligent situational awareness system," *Sustainability*, vol. 15, no. 4, p. 2866, 2023.

[47] A. Nourbakhshrezaei, M. Jadidi, M. Delavar, and B. Moshiri, "A novel context-aware system to improve driver's field of view in urban traffic networks," *Journal of Intelligent Transportation Systems*, pp. 1–16, 2022.

[48] M. J. Koetse and P. Rietveld, "The impact of climate change and weather on transport: An overview of empirical findings," *Transportation Research Part D: Transport and Environment*, vol. 14, no. 3, pp. 205–221, 2009.

[49] B. Chen and H. H. Cheng, "A review of the applications of agent technology in traffic and transportation systems," *IEEE Transactions on intelligent transportation systems*, vol. 11, no. 2, pp. 485–497, 2010.

[50] A. Gooze, K. E. Watkins, and A. Borning, "Benefits of real-time transit information and impacts of data accuracy on rider experience," *Transportation Research Record*, vol. 2351, no. 1, pp. 95–103, 2013.

[51] Q. Wang and C. Tang, "Deep reinforcement learning for transportation network combinatorial optimization: A survey," *Knowledge-Based Systems*, vol. 233, p. 107526, 2021.

[52] J. Barceló, H. Grzybowska, and S. Pardo, "Vehicle routing and scheduling models, simulation and city logistics," *Dynamic Fleet Management: Concepts, Systems, Algorithms & Case Studies*, pp. 163–195, 2007.

[53] T. E. Tzoreff, D. Granot, F. Granot, and G. Sovisic, "The vehicle routing problem with pickups and deliveries on some special graphs," *Discrete Applied Mathematics*, vol. 116, no. 3, pp. 193–229, 2002.

[54] A. R. Barron, A. Cohen, W. Dahmen, and R. A. DeVore, "Approximation and learning by greedy algorithms," *Ann. Statist. 36 (1) 64 - 94*, 2008.

[55] J. B. Orlin, A. P. Punnen, and A. S. Schulz, "Approximate local search in combinatorial optimization," *SIAM Journal on Computing*, vol. 33, no. 5, pp. 1201–1214, 2004.

[56] G. Panchal and D. Panchal, "Solving np hard problems using genetic algorithm," *Transportation*, vol. 106, pp. 6–2, 2015.

[57] E. Aarts, J. Korst, and W. Michiels, "Simulated annealing," *Search methodologies: introductory tutorials in optimization and decision support techniques*, pp. 187–210, 2005.

[58] D. Ahr and G. Reinelt, "A tabu search algorithm for the min–max k-chinese postman problem," *Computers & operations research*, vol. 33, no. 12, pp. 3403–3422, 2006.

[59] H. N. Djidjev, G. Chapuis, G. Hahn, and G. Rizk, "Efficient combinatorial optimization using quantum annealing," *arXiv preprint arXiv:1801.08653*, 2018.

[60] A. Ajagekar and F. You, "Quantum computing for energy systems optimization: Challenges and opportunities," *Energy*, vol. 179, pp. 76–89, 2019.

[61] D. Zouache, F. Nouioua, and A. Moussaoui, "Quantum-inspired firefly algorithm with particle swarm optimization for discrete optimization problems," *Soft Computing*, vol. 20, pp. 2781–2799, 2016.

[62] Z. Bian, F. Chudak, R. Israel, B. Lackey, W. G. Macready, and A. Roy, "Discrete optimization using quantum annealing on sparse ising models," *Frontiers in Physics*, vol. 2, p. 56, 2014.

[63] E. Zahedinejad and A. Zaribafiyan, "Combinatorial optimization on gate model quantum computers: A survey," *arXiv preprint arXiv:1708.05294*, 2017.

[64] J. West, "The quantum computer," *Retrieved December*, vol. 1, p. 2002, 2000.

[65] T. Kato, "On the adiabatic theorem of quantum mechanics," *Journal of the Physical Society of Japan*, vol. 5, no. 6, pp. 435–439, 1950.

[66] J. Roland and N. J. Cerf, "Quantum search by local adiabatic evolution," *Physical Review A*, vol. 65, no. 4, p. 042308, 2002.

[67] L. K. Grover, "A fast quantum mechanical algorithm for database search," *arXiv preprint quant-ph/9605043*, 1996.

[68] T. Kadowaki and H. Nishimori, "Quantum annealing in the transverse ising model," *Physical Review E*, vol. 58, no. 5, p. 5355, 1998.

[69] R. Harris, M. Johnson, T. Lanting, A. Berkley, J. Johansson, P. Bunyk, E. Tolkacheva, E. Ladizinsky, T. Oh, F. Cioata, *et al.*, "Experimental demonstration of a robust and scalable flux qubit," *Physical Review B*, vol. 81, no. 13, p. 134510, 2010.

[70] A. Lucas, "Ising formulations of many np problems," *Frontiers in Physics*, vol. 2, p. 5, 2014.

[71] R. Raussendorf and H. J. Briegel, "A one-way quantum computer," *Physical Review Letters*, vol. 86, no. 22, p. 5188, 2001.

[72] W. Pfaff, T. H. Taminiau, L. Robledo, H. Bernien, M. Markham, D. J. Twitchen, and R. Hanson, "Demonstration of entanglement-by-measurement of solid-state qubits," *Nature Physics*, vol. 9, no. 1, pp. 29–33, 2013.

[73] D. Gross, S. T. Flammia, and J. Eisert, "Most quantum states are too entangled to be useful as computational resources," *Physical Review Letters*, vol. 102, no. 19, p. 190501, 2009.

[74] T. Morimae, "Hardness of classically simulating the one-clean-qubit model," *Physical Review A*, vol. 96, no. 4, p. 040302, 2017.

[75] T. Morimae, "Blind quantum computing can be efficiently simulated by classical computers," *arXiv preprint arXiv:1407.0530*, 2014.

[76] M. V. Larsen, Q. Guo, C. R. Breum, J. S. Neergaard-Nielsen, and U. L. Andersen, "Deterministic generation of a two-dimensional cluster state," *Science*, vol. 366, no. 6463, pp. 369–372, 2019.

[77] A. Broadbent, J. Fitzsimons, and E. Kashefi, "Universal blind quantum computation," *2009 50th Annual IEEE Symposium on Foundations of Computer Science*, pp. 517–526, 2009.

[78] J. Fitzsimons, "Private quantum computation: an introduction to blind quantum computing and related protocols," *npj Quantum Information*, vol. 3, no. 1, p. 23, 2017.

[79] T.-C. Wei, "Measurement-based quantum computation," *arXiv preprint arXiv:2103.12360*, 2021.

[80] H. J. Briegel, D. E. Browne, W. Dür, R. Raussendorf, and M. Van den Nest, "Measurement-based quantum computation," *Nature Physics*, vol. 5, no. 1, pp. 19–26, 2009.

[81] D. E. Browne and T. Rudolph, "One-way quantum computing," *Nature Physics*, vol. 2, no. 1, pp. 124–129, 2016.

[82] T. Karzig, C. Knapp, R. M. Lutchyn, P. Bonderson, M. B. Hastings, C. Nayak, J. Alicea, K. Flensberg, S. Plugge, Y. Oreg, *et al.*, "Topological quantum computation with near-term devices," *Nature Reviews Physics*, vol. 3, no. 5, pp. 241–256, 2021.

[83] A. Stern, "Non-abelian anyons: theory and experiment," *Nature*, vol. 464, no. 7286, pp. 187–193, 2010.

[84] F. Wilczek, "Majorana returns," *Nature Physics*, vol. 5, no. 9, pp. 614–618, 2009.

[85] H. Bartolomei, A. Marguerite, J. M. Berroir, E. Bocquillon, B. Placais, A. Cavanna, Q. Dong, U. Gennser, Y. Jin, and G. Feve, "Fractional statistics in anyon collisions," *Science*, vol. 368, no. 6487, pp. 173–177, 2020.

[86] J. Nakamura, S. Liang, G. Gardner, and M. Manfra, "Direct observation of anyonic braiding statistics at the $\nu = 1/3$ fractional quantum hall state," *Nature Physics*, vol. 16, no. 9, pp. 931–936, 2020.

[87] R. M. Lutchyn, E. P. Bakkers, L. P. Kouwenhoven, P. Krogstrup, C. M. Marcus, and Y. Oreg, "Majorana zero modes in superconductor–semiconductor heterostructures," *Nature Reviews Materials*, vol. 3, no. 5, pp. 52–68, 2018.

[88] Y. Oreg, G. Refael, and F. von Oppen, "Helical liquids and majorana bound states in quantum wires," *Physical review letters*, vol. 105, no. 17, p. 177002, 2010.

[89] T. Karzig, C. Knapp, R. M. Lutchyn, P. Bonderson, M. B. Hastings, C. Nayak, J. Alicea, K. Flensberg, S. Plugge, Y. Oreg, *et al.*, "Scalable designs for quasiparticle-poisoning-protected topological quantum computation with majorana zero modes," *Physical Review B*, vol. 95, no. 23, p. 235305, 2017.

[90] A. Y. Kitaev, "Fault-tolerant quantum computation by anyons," *Annals of Physics*, vol. 303, no. 1, pp. 2–30, 2003.

[91] S. Feng, H. Chen, C. Du, J. Li, and N. Jing, "A hierarchical demand prediction method with station clustering for bike sharing system," in *2018 IEEE Third International Conference on Data Science in Cyberspace (DSC)*, pp. 829–836, IEEE, 2018.

[92] C. Kloimüllner and G. R. Raidl, "Hierarchical clustering and multilevel refinement for the bike-sharing station planning problem," in *Learning and Intelligent Optimization: 11th International Conference, LION 11, Nizhny Novgorod, Russia, June 19-21, 2017, Revised Selected Papers 11*, pp. 150–165, Springer, 2017.

[93] D. Birant and A. Kut, "St-dbscan: An algorithm for clustering spatial–temporal data," *Data & knowledge engineering*, vol. 60, no. 1, pp. 208–221, 2007.

[94] A. Milne, M. Rounds, and P. Goddard, "Optimal feature selection in credit scoring and classification using a quantum annealer," *White Paper 1Qbit*, 2017.

[95] R. Nembrini, M. Ferrari Dacrema, and P. Cremonesi, "Feature selection for recommender systems with quantum computing," *Entropy*, vol. 23, no. 8, p. 970, 2021.

[96] G. Hellstern, V. Dehn, and M. Zaefferer, "Quantum computer based feature selection in machine learning," *IET Quantum Communication*, 2023.

[97] S. Mücke, R. Heese, S. Müller, M. Wolter, and N. Piatkowski, "Feature selection on quantum computers," *Quantum Machine Intelligence*, vol. 5, no. 1, p. 11, 2023.

[98] A. Vlasic, H. Grant, and S. Certo, "An advantage using feature selection with a quantum annealer," *arXiv preprint arXiv:2211.09756*, 2022.

[99] B. M. Sabbar and H. A. Rasool, "Quantum inspired genetic algorithm model based automatic modulation classification," *Management*, 2021.

[100] Y. Li, R.-G. Zhou, R. Xu, J. Luo, W. Hu, and P. Fan, "Implementing graph-theoretic feature selection by quantum approximate optimization algorithm," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[101] R. Bhagawati and T. Subramanian, "Quantum-aided feature selection model–a quantum machine learning approach," *Journal of Discrete Mathematical Sciences and Cryptography*, 2023.

[102] F. Hu, B.-N. Wang, N. Wang, and C. Wang, "Quantum machine learning with d-wave quantum computer," *Quantum Engineering*, vol. 1, no. 2, p. e12, 2019.

[103] L. Wang, Z.-Y. Chen, F.-Y. Le, Z.-Q. Yu, C. Xue, X.-N. Zhuang, Q. Yan, Y. Yang, Y.-C. Wu, and G.-P. Guo, "A quantum feature selection framework via ground state preparation," *Physica Scripta*, vol. 98, no. 11, p. 115121, 2023.

[104] M. Ferrari Dacrema, F. Moroni, R. Nembrini, N. Ferro, G. Faggioli, and P. Cremonesi, "Towards feature selection for ranking and classification exploiting quantum annealers," in *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 2814–2824, 2022.

[105] G. Turati, M. F. Dacrema, and P. Cremonesi, "Feature selection for classification with qaoa," in *2022 IEEE International Conference on Quantum Computing and Engineering (QCE)*, pp. 782–785, IEEE, 2022.

[106] WHO, "Who coronavirus (covid-19)," *World Health Organization*, 2021.

[107] WHO, "Coronavirus disease 2019 (covid-19): situation report, 72," *World Health Organization*, 2020.

[108] A. Carten, L. Di Francesco, and M. Martino, "The role of transport accessibility within the spread of the coronavirus pandemic in italy," *Safety science*, vol. 133, p. 104999, 2021.

[109] P. Shi, Y. Dong, H. Yan, C. Zhao, X. Li, W. Liu, M. He, S. Tang, and S. Xi, "Impact of temperature on the dynamics of the covid-19 outbreak in china," *Science of The Total Environment*, vol. 728, p. 138890, 2020.

[110] R. Tosepu, J. Gunawan, D. S. Effendy, L. O. A. I. Ahmad, H. Lestari, H. Bahar, and P. Asfian, "Correlation between weather and covid-19 pandemic in jakarta, indonesia," *Science of The Total Environment*, vol. 725, p. 138436, 2020.

[111] J. Xie and Y. Zhu, "Association between ambient temperature and covid-19 infection in 122 cities from china," *Science of The Total Environment*, vol. 724, p. 138201, 2020.

[112] D. Chen, S. Pan, Q. Chen, and J. Liu, "Vehicle routing problem of contactless joint distribution service during covid-19 pandemic," *Transportation Research Interdisciplinary Perspectives*, vol. 8, p. 100233, 2020.

[113] J. K. Lenstra and A. R. Kan, "Complexity of vehicle routing and scheduling problems," *Networks*, vol. 11, no. 2, pp. 221–227, 1981.

[114] R. D. Somma, D. Nagaj, and M. Kieferová, "Quantum speedup by quantum annealing," *Physical review letters*, vol. 109, no. 5, p. 050501, 2012.

[115] E. Gibney, "Hello quantum world google publishes landmark quantum supremacy claim," *Nature*, vol. 574, no. 7779, pp. 461–463, 2019.

[116] F. Neukart, G. Compostella, C. Seidel, D. Von Dollen, S. Yarkoni, and B. Parney, "Traffic flow optimization using a quantum annealer," *Frontiers in ICT*, vol. 4, p. 29, 2017.

[117] K. Michielsen, M. Nocon, D. Willsch, F. Jin, T. Lippert, and H. De Raedt, "Benchmarking gate-based quantum computers," *Computer Physics Communications*, vol. 220, pp. 44–55, 2017.

[118] T. Albash and D. A. Lidar, "Adiabatic quantum computation," *Reviews of Modern Physics*, vol. 90, no. 1, p. 015002, 2018.

[119] A. Roy and D. P. DiVincenzo, "Topological quantum computing," *arXiv preprint arXiv:1701.05052*, 2017.

[120] D. Nagaj, "Universal two-body-hamiltonian quantum computing," *Physical Review A*, vol. 85, no. 3, p. 032330, 2012.

[121] G. Kochenberger, J.-K. Hao, F. Glover, M. Lewis, Z. Lü, H. Wang, and Y. Wang, "The unconstrained binary quadratic programming problem: a survey," *Journal of combinatorial optimization*, vol. 28, no. 1, pp. 58–81, 2014.

[122] P. O. Scokaert and J. B. Rawlings, "Constrained linear quadratic regulation," *IEEE Transactions on automatic control*, vol. 43, no. 8, pp. 1163–1169, 1998.

[123] T. Hogg, "Adiabatic quantum computing for random satisfiability problems," *Physical Review A*, vol. 67, no. 2, p. 022314, 2003.

[124] J. Liu, L. Sun, W. Chen, and H. Xiong, "Rebalancing bike sharing systems: A multi-source data smart optimization," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1005–1014, 2016.

[125] T. Raviv and O. Kolka, "Optimal inventory management of a bike-sharing station," *Iie Transactions*, vol. 45, no. 10, pp. 1077–1093, 2013.

[126] M. Rainer-Harbach, P. Papazek, B. Hu, and G. R. Raidl, "Balancing bicycle sharing systems: A variable neighborhood search approach," in *European conference on evolutionary computation in combinatorial optimization*, pp. 121–132, Springer, 2013.

[127] C. Contardo, C. Morency, and L.-M. Rousseau, *Balancing a dynamic public bike-sharing system*, vol. 4. Cirrelt Montreal, 2012.

[128] J. Schuijbroek, R. C. Hampshire, and W.-J. Van Hoeve, "Inventory rebalancing and vehicle routing in bike sharing systems," *European Journal of Operational Research*, vol. 257, no. 3, pp. 992–1004, 2017.

[129] S. Ghosh and P. Varakantham, "Incentivizing the use of bike trailers for dynamic repositioning in bike sharing systems," in *Proceedings of the International Conference on Automated Planning and Scheduling*, vol. 27, pp. 373–381, 2017.

[130] S. C. Ho and W. Szeto, "Solving a static repositioning problem in bike-sharing systems using iterated tabu search," *Transportation Research Part E: Logistics and Transportation Review*, vol. 69, pp. 180–198, 2014.

[131] F. Chiariotti, C. Pielli, A. Zanella, and M. Zorzi, "A dynamic approach to rebalancing bike-sharing systems," *Sensors*, vol. 18, no. 2, p. 512, 2018.

[132] L. Caggiani and M. Ottomanelli, "A dynamic simulation based model for optimal fleet repositioning in bike-sharing systems," *Procedia-Social and Behavioral Sciences*, vol. 87, pp. 203–210, 2013.

[133] L. Pan, Q. Cai, Z. Fang, P. Tang, and L. Huang, "A deep reinforcement learning framework for rebalancing dockless bike sharing systems," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, pp. 1393–1400, 2019.

[134] G. Laporte and Y. Nobert, "Exact algorithms for the vehicle routing problem," in *North-Holland Mathematics Studies*, vol. 132, pp. 147–184, Elsevier, 1987.

[135] Y. Lu, U. Benlic, and Q. Wu, "An effective memetic algorithm for the generalized bike-sharing rebalancing problem," *Engineering Applications of Artificial Intelligence*, vol. 95, p. 103890, 2020.

[136] A. Singla, M. Santoni, G. Bartók, P. Mukerji, M. Meenen, and A. Krause, "Incentivizing users for balancing bike sharing systems," in *Twenty-Ninth AAAI conference on artificial intelligence*, 2015.

[137] E. Osaba, E. Villar-Rodriguez, and I. Oregi, "A systematic literature review of quantum computing for routing problems," *IEEE Access*, 2022.

[138] A. Singh, C.-Y. Lin, C.-I. Huang, and F.-P. Lin, "Quantum annealing approach for the optimal real-time traffic control using qubo," in *2021 IEEE/ACIS 22nd International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*, pp. 74–81, IEEE, 2021.

[139] C. H. Cooper, "Exploring potential applications of quantum computing in transportation modelling," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 9, pp. 14712–14720, 2021.

[140] W. Y. Suen, C. Y. Lee, and H. C. Lau, "Quantum-inspired algorithm for vehicle sharing problem," in *2021 IEEE International Conference on Quantum Computing and Engineering (QCE)*, pp. 17–23, IEEE, 2021.

[141] X. You, X. Miao, and S. Liu, "Quantum computing-based ant colony optimization algorithm for tsp," in *2009 2nd International Conference on Power Electronics and Intelligent Transportation System (PEITS)*, vol. 3, pp. 359–362, IEEE, 2009.

[142] X. Yan, N. Lv, Z. Liu, and K. Xu, "Quantum-inspired evolutionary algorithm for transportation network design optimization," in *2008 Second International Conference on Genetic and Evolutionary Computing*, pp. 189–192, IEEE, 2008.

[143] S. Wang, Z. Pei, C. Wang, and J. Wu, "Shaping the future of the application of quantum computing in intelligent transportation system," *Intelligent and Converged Networks*, vol. 2, no. 4, pp. 259–276, 2021.

[144] R. Harikrishnakumar, S. E. Borujeni, A. Dand, and S. Nannapaneni, "A quantum bayesian approach for bike sharing demand prediction," in *2020 IEEE International Conference on Big Data (Big Data)*, pp. 2401–2409, IEEE, 2020.

[145] R. Harikrishnakumar, S. E. Borujeni, S. F. Ahmad, and S. Nannapaneni, "Rebalancing bike sharing systems under uncertainty using quantum bayesian networks," in *2021 IEEE Inter-*

national Conference on Quantum Computing and Engineering (QCE), pp. 461–462, IEEE, 2021.

[146] R. Harikrishnakumar and S. Nannapaneni, "Smart rebalancing for bike sharing systems using quantum approximate optimization algorithm," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pp. 2257–2263, IEEE, 2021.

[147] V. S. Denchev, S. Boixo, S. V. Isakov, N. Ding, R. Babbush, V. Smelyanskiy, J. Martinis, and H. Neven, "What is the computational value of finite-range tunneling?," *Physical Review X*, vol. 6, no. 3, p. 031015, 2016.

[148] D. Venturelli, D. Marchand, and G. Rojo, "Job shop scheduling solver based on quantum annealing," in *Proc. of ICAPS-16 Workshop on Constraint Satisfaction Techniques for Planning and Scheduling (COPLAS)*, pp. 25–34, 2016.

[149] P. Martin and D. B. Shmoys, "A new approach to computing optimal schedules for the job-shop scheduling problem," in *Integer Programming and Combinatorial Optimization* (W. H. Cunningham, S. T. McCormick, and M. Queyranne, eds.), (Berlin, Heidelberg), pp. 389–403, Springer Berlin Heidelberg, 1996.

[150] P. DeMaio, "Bike-sharing: History, impacts, models of provision, and future," *Journal of public transportation*, vol. 12, no. 4, pp. 41–56, 2009.

[151] S. Bi, R. Xu, A. Liu, L. Wang, and L. Wan, "Mining taxi pick-up hotspots based on grid information entropy clustering algorithm," *Journal of Advanced Transportation*, vol. 2021, pp. 1–25, 2021.

[152] M. Wang and X. Zhou, "Bike-sharing systems and congestion: Evidence from us cities," *Journal of transport geography*, vol. 65, pp. 147–154, 2017.

[153] B. Cheng, J. Li, H. Su, K. Lu, H. Chen, and J. Huang, "Life cycle assessment of greenhouse gas emission reduction through bike-sharing for sustainable cities," *Sustainable Energy Technologies and Assessments*, vol. 53, p. 102789, 2022.

[154] L. Caggiani and R. Camporeale, "Toward sustainability: Bike-sharing systems design, simulation and management," 2021.

[155] P. Vogel, T. Greiser, and D. C. Mattfeld, "Understanding bike-sharing systems using data mining: Exploring activity patterns," *Procedia-Social and Behavioral Sciences*, vol. 20, pp. 514–523, 2011.

[156] R. P. Feynman, "Quantum mechanical computers," in *Conference on Lasers and Electro-Optics*, p. TUAA2, Optica Publishing Group, 1984.

[157] J. Preskill, "Quantum computing 40 years later," in *Feynman Lectures on Computation*, pp. 193–244, CRC Press, 2023.

[158] I. L. Chuang, N. Gershenfeld, and M. Kubinec, "Experimental implementation of fast quantum searching," *Physical review letters*, vol. 80, no. 15, p. 3408, 1998.

[159] I. L. Chuang, L. M. Vandersypen, X. Zhou, D. W. Leung, and S. Lloyd, "Experimental realization of a quantum algorithm," *Nature*, vol. 393, no. 6681, pp. 143–146, 1998.

[160] T. D. Ladd, F. Jelezko, R. Laflamme, Y. Nakamura, C. Monroe, and J. L. O'Brien, "Quantum computers," *nature*, vol. 464, no. 7285, pp. 45–53, 2010.

[161] V. M. Kendon, K. Nemoto, and W. J. Munro, "Quantum analogue computing," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 368, no. 1924, pp. 3609–3620, 2010.

[162] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, "Quantum machine learning," *Nature*, vol. 549, no. 7671, pp. 195–202, 2017.

[163] Z. Yang, M. Zolanvari, and R. Jain, "A survey of important issues in quantum computing and communications," *IEEE Communications Surveys & Tutorials*, 2023.

[164] J. Yang, B. Guo, Z. Wang, and Y. Ma, "Hierarchical prediction based on network-representation-learning-enhanced clustering for bike-sharing system in smart city," *IEEE Internet of Things Journal*, vol. 8, no. 8, pp. 6416–6424, 2020.

[165] B. Wang, Y. Tan, and W. Jia, "Tl-fcm: A hierarchical prediction model based on two-level fuzzy c-means clustering for bike-sharing system," *Applied Intelligence*, pp. 1–18, 2022.

[166] Z. Tong, Y. Zhu, Z. Zhang, R. An, Y. Liu, and M. Zheng, "Unravel the spatio-temporal patterns and their nonlinear relationship with correlates of dockless shared bikes near metro stations," *Geo-Spatial Information Science*, vol. 26, no. 3, pp. 577–598, 2023.

[167] N. P. Shangaranarayanee, V. Aakashbabu, M. Balamurugan, and R. Gokulraj, "Machine learning driven smart transportation sharing," *Journal of ISMAC*, 2024.

[168] M. Galvani, A. Torti, A. Menafoglio, and S. Vantini, "A novel spatio-temporal clustering technique to study the bike sharing system in lyon.," in *EDBT/ICDT Workshops*, pp. volume 1–2578, 2020.

[169] Z. Wang, *Topological quantum computation*. American Mathematical Soc., 2010.

[170] L. P. Yulianti and K. Surendro, "Implementation of quantum annealing: A systematic review," *IEEE Access*, 2022.

[171] S. Lloyd, "Hybrid quantum computing," in *Quantum information with continuous variables*, pp. 37–45, Springer, 2003.

[172] E. Knill, "Quantum computing with realistically noisy devices," *Nature*, vol. 434, no. 7029, pp. 39–44, 2005.

[173] J. R. McClean, J. Romero, R. Babbush, and A. Aspuru-Guzik, "The theory of variational hybrid quantum-classical algorithms," *New Journal of Physics*, vol. 18, no. 2, p. 023023, 2016.

[174] J. Tilly, H. Chen, S. Cao, D. Picozzi, K. Setia, Y. Li, E. Grant, L. Wossnig, I. Rungger, G. H. Booth, *et al.*, "The variational quantum eigensolver: a review of methods and best practices," *Physics Reports*, vol. 986, pp. 1–128, 2022.

[175] E. H. Houssein, Z. Abohashima, M. Elhoseny, and W. M. Mohamed, "Machine learning in the quantum realm: The state-of-the-art, challenges, and future vision," *Expert Systems with Applications*, vol. 194, p. 116512, 2022.

[176] M. H. Amin, E. Andriyash, J. Rolfe, B. Kulchytskyy, and R. Melko, "Quantum boltzmann machine," *Physical Review X*, vol. 8, no. 2, p. 021050, 2018.

[177] M. Kieferova and N. Wiebe, "Tomography and generative data modeling via quantum boltzmann training," *arXiv preprint arXiv:1612.05204*, 2016.

[178] P. Rebentrost, M. Mohseni, and S. Lloyd, "Quantum support vector machine for big data classification," *Physical review letters*, vol. 113, no. 13, p. 130503, 2014.

[179] S. Lloyd, M. Mohseni, and P. Rebentrost, "Quantum principal component analysis," *Nature Physics*, vol. 10, no. 9, pp. 631–633, 2014.

[180] V. Dunjko, J. M. Taylor, and H. J. Briegel, "Quantum-enhanced machine learning," *Physical review letters*, vol. 117, no. 13, p. 130501, 2016.

[181] W. M. Kaminsky and S. Lloyd, "Scalable architecture for adiabatic quantum computing of np-hard problems," *Quantum computing and quantum bits in mesoscopic systems*, pp. 229–236, 2004.

[182] C. R. Laumann, R. Moessner, A. Scardicchio, and S. L. Sondhi, "Quantum annealing: The fastest route to quantum computation?," *The European Physical Journal Special Topics*, vol. 224, no. 1, pp. 75–88, 2015.

[183] S. C. Brailsford, C. N. Potts, and B. M. Smith, "Constraint satisfaction problems: Algorithms and applications," *European journal of operational research*, vol. 119, no. 3, pp. 557–581, 1999.

[184] M. Syakur, B. K. Khotimah, E. Rochman, and B. D. Satoto, "Integration k-means clustering method and elbow method for identification of the best customer profile cluster," in *IOP conference series: materials science and engineering*, vol. 336, p. 012017, IOP Publishing, 2018.

[185] D.-T. Dinh, T. Fujinami, and V.-N. Huynh, "Estimating the optimal number of clusters in categorical data clustering by silhouette coefficient," in *Knowledge and Systems Sciences: 20th International Symposium, KSS 2019, Da Nang, Vietnam, November 29–December 1, 2019, Proceedings 20*, pp. 1–17, Springer, 2019.

[186] N. Bouhmala, "How good is the euclidean distance metric for the clustering problem," in *2016 5th IIAI international congress on advanced applied informatics (IIAI-AAI)*, pp. 312–315, IEEE, 2016.

[187] S. Kapil and M. Chawla, "Performance evaluation of k-means clustering algorithm with various distance metrics," in *2016 IEEE 1st international conference on power electronics, intelligent control and energy systems (ICPEICES)*, pp. 1–4, IEEE, 2016.

[188] M. R. Berthold and F. Höppner, "On clustering time series using euclidean distance and pearson correlation," *arXiv preprint arXiv:1601.02213*, 2016.

[189] E. Eren and V. E. Sathishkumar, "A review on bike-sharing: The factors affecting bike-sharing demand," *Sustainable Cities and Society*, 2020.

[190] S. V. E and J. Prince, "Using data mining techniques for bike sharing demand prediction in metropolitan city," *Computer Communications*, 2020.

[191] S. Leem and J. Oh, "Enhancing multistep-ahead bike-sharing demand prediction with a two-stage online learning-based time-series model: insight from seoul," *The Journal of Supercomputing*, 2024.

[192] S. Feng, H. Chen, C. Du, J. Li, and N. Jing, "A hierarchical demand prediction method with station clustering for bike sharing system," in *IEEE Third International Conference on Data Science in Cyberspace (DSC)*, 2018.

[193] Y. Guo and J. Zhang, "Identifying the factors affecting bike-sharing usage and degree of satisfaction in ningbo, china," *PLoS ONE*, vol. 12, no. 9, p. e0185100, 2017.

[194] K. Schimohr and P. D. Baer, "Prediction of bike-sharing trip counts: Comparing parametric spatial regression models to a geographically weighted xgboost algorithm," *Geographical analysis journal of theoretical geography*, 2022.

[195] V. Albuquerque and M. Silva, "Machine learning approaches to bike-sharing systems: A systematic literature review," *ISPRS Geo Information*, 2021.

[196] S. Sohrabi and R. P. Pel, "Real-time prediction of public bike sharing system demand using generalized extreme value count model," *Transportation Research Part A: Policy and Practice*, 2020.

[197] M. Hua and J. Chen, "Forecasting usage and bike distribution of dockless bike-sharing using journey data," *IET Intelligent Transport Systems*, 2020.

[198] Y. Yang and A. Hohl, "Using graph structural information about flows to enhance short-term demand prediction in bike-sharing systems," *Computers, Environment and Urban Systems*, 2020.

[199] W. Chen and X. Chen, "What factors influence ridership of station-based bike sharing and free-floating bike sharing at rail transit stations?," *International Journal of Sustainable Transportation*, 2022.

[200] Y.-H. Seo, S. Yoon-Keun, and Y. Kim, "Predicting demand for a bike-sharing system with station activity based on random forest," *Machine Learning and Artificial Intelligence in Municipal Engineering*, 2021.

[201] P. Lin, J. Weng, S. Hu, D. Alivanistos, X. Li, and B. Yin, "Revealing spatio-temporal patterns and influencing factors of dockless bike sharing demand," in *IEEE*, 2020.

[202] J. Li and Y. S. Soh, "Deep convolutional neural network based ecg classification system using information fusion and one-hot encoding techniques," *Mathematical problems in engineering*, 2018.

[203] J. Cai, J. Luo, S. Wang, and S. Yang, "Feature selection in machine learning: A new perspective," *Neurocomputing*, vol. 300, pp. 70–79, 2018.

[204] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artificial intelligence*, vol. 97, no. 1-2, pp. 273–324, 1997.

[205] D. R. Munirathinam and M. Ranganadhan, "A new improved filter-based feature selection model for high-dimensional data," *The Journal of Supercomputing*, vol. 76, no. 8, pp. 5745–5762, 2020.

[206] V. Varma and M. L. Engineer, "Embedded methods for feature selection in neural networks," *arXiv preprint arXiv:2010.05834*, 2020.

[207] A. Jović, K. Brkić, and N. Bogunović, "A review of feature selection methods with applications," in *2015 38th international convention on information and communication technology, electronics and microelectronics (MIPRO)*, pp. 1200–1205, Ieee, 2015.

[208] L. Batina, B. Gierlichs, E. Prouff, M. Rivain, F.-X. Standaert, and N. Veyrat-Charvillon, "Mutual information analysis: a comprehensive study," *Journal of Cryptology*, vol. 24, no. 2, pp. 269–291, 2011.

[209] R. L. Iman and W. Conover, "A measure of top–down correlation," *Technometrics*, vol. 29, no. 3, pp. 351–357, 1987.

[210] Y. Grandvalet, "Least absolute shrinkage is equivalent to quadratic penalization," in *ICANN 98: Proceedings of the 8th International Conference on Artificial Neural Networks, Skövde, Sweden, 2–4 September 1998 8*, pp. 201–206, Springer, 1998.

[211] C. A. A. F. M. T. Cortez Paulo and R. J., "Wine Quality." UCI Machine Learning Repository, 2009. DOI: https://doi.org/10.24432/C56S3T.

[212] B. Lantz, "Machine learning with r," 2013.

[213] City of Toronto, "Bike share toronto ridership data," 2024. Accessed: 05/04/2024.

[214] City of Toronto, "Ward profiles - 25 ward model," 2021. Accessed: 06/04/2024.

Chapter 8

Appendices

A Source Codes

1. Source Code for Chapter 2: Code, Readme
2. Source Code for Chapter 3: Code, Readme
3. Source Code for Chapter 4: Code, Readme
4. Source Code for Chapter 5: Code, Readme
5. Source Code for Chapter 6: Code, Readme

B Results

1. Results of the section 3 can be found at the end of each section in the notebook: Notebook
2. Results of the section 4 can be found at the end of each section in the notebook: Notebook
3. Results of the section 5 can be found at the end of each section in the notebook: Notebook
4. Results of the section 6 can be found at the end of each section in the notebook: Notebook

B.1 Additional Results for Chapter 6

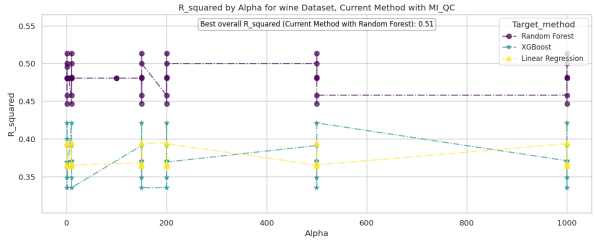


Figure 8.1: RMSE by Selected Features for wine Dataset, Proposed Method with Linear QC.

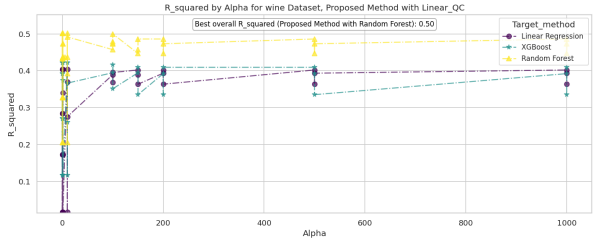


Figure 8.2: R^2 by Alpha for wine Dataset, Current Method with MI QC.

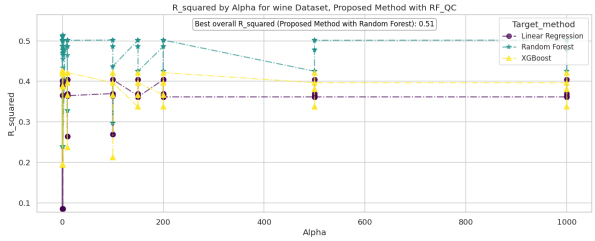


Figure 8.3: R^2 by Alpha for wine Dataset, Proposed Method with Linear QC.

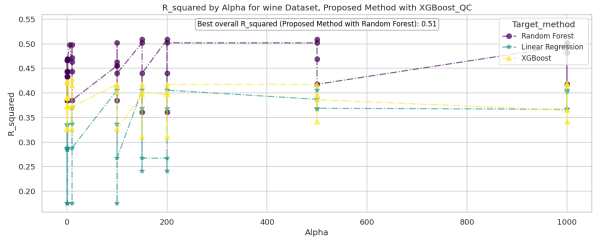


Figure 8.4: R^2 by Alpha for wine Dataset, Proposed Method with RF QC.

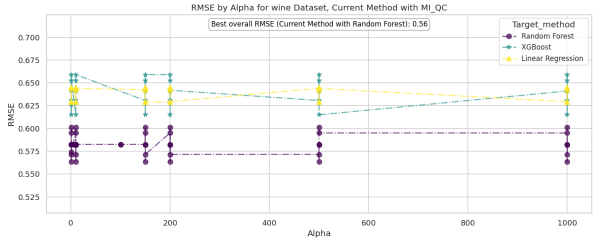


Figure 8.5: R^2 by Alpha for wine Dataset, Proposed Method with XGBoost QC.

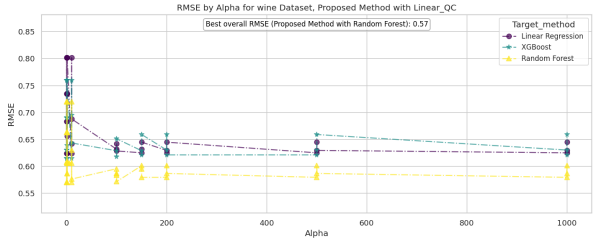


Figure 8.6: RMSE by Alpha for wine Dataset, Current Method with MI QC.

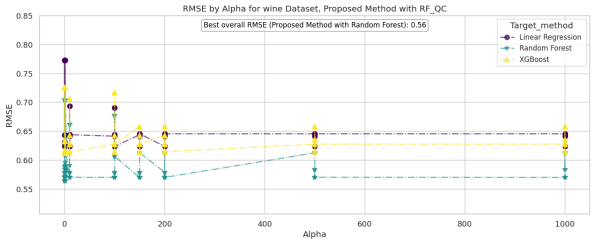


Figure 8.7: RMSE by Alpha for wine Dataset, Proposed Method with Linear QC.

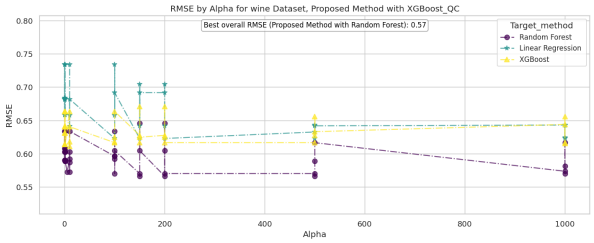


Figure 8.8: RMSE by Alpha for wine Dataset, Proposed Method with RF QC.

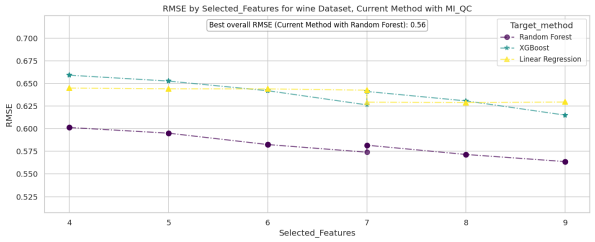


Figure 8.9: RMSE by Alpha for wine Dataset, Proposed Method with XGBoost QC.

Figure 8.10: RMSE by Selected Features for wine Dataset, Current Method with MI QC.

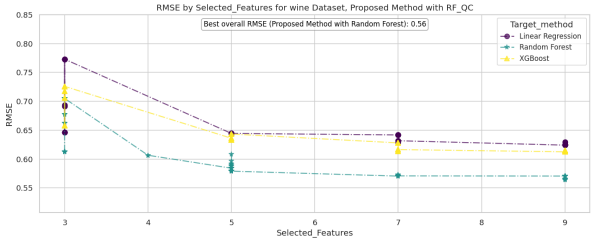


Figure 8.11: RMSE by Selected Features for wine Dataset, Proposed Method with RF QC

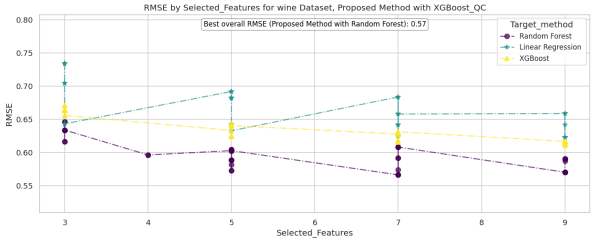


Figure 8.12: RMSE by Selected Features for wine Dataset, Proposed Method with XGBoost QC

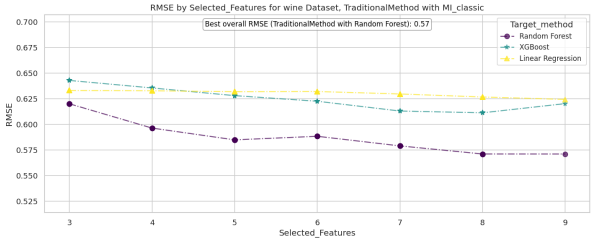


Figure 8.13: RMSE by Selected Features for wine Dataset, Traditional Method with MI classic

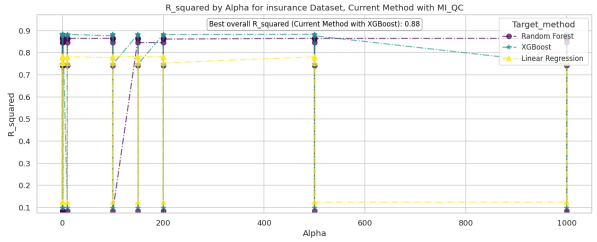


Figure 8.14: R^2 by Alpha for insurance Dataset, Current Method with MI_QC. Best overall R^2 (Current Method with XGBoost): 0.88

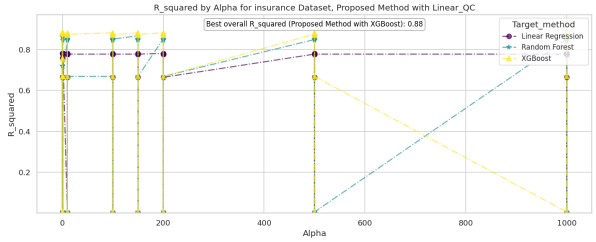


Figure 8.15: R^2 by Alpha for insurance Dataset, Proposed Method with Linear_QC. Best overall R^2 (Proposed Method with XGBoost): 0.88

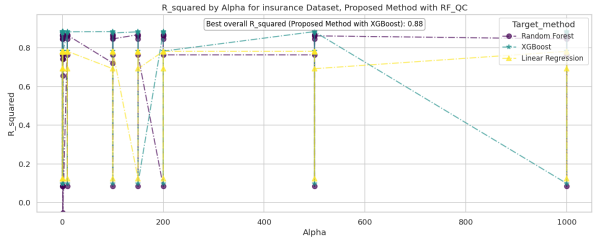


Figure 8.16: R^2 by Alpha for insurance Dataset, Proposed Method with RF_QC. Best overall R^2 (Proposed Method with XGBoost): 0.88

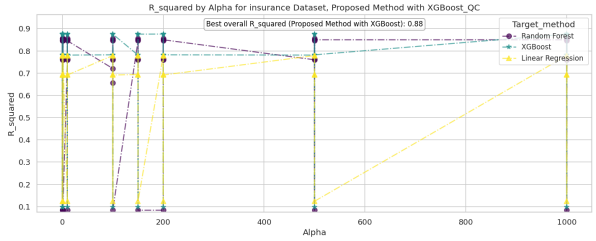


Figure 8.17: R^2 by Alpha for insurance Dataset, Proposed Method with XGBoost_QC. Best overall R^2 (Proposed Method with XGBoost): 0.88

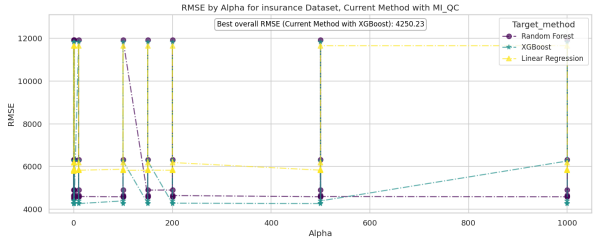


Figure 8.18: RMSE by Alpha for insurance Dataset, Current Method with MI_QC. Best overall RMSE (Current Method with XGBoost): 4250.23

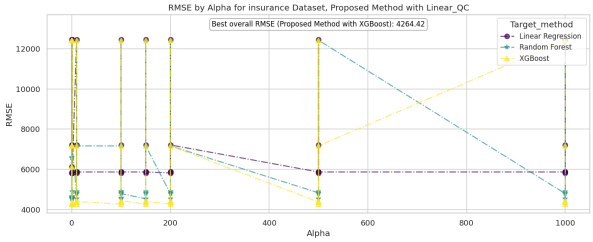


Figure 8.19: RMSE by Alpha for insurance Dataset, Proposed Method with Linear_QC. Best overall RMSE (Proposed Method with XGBoost): 4264.42

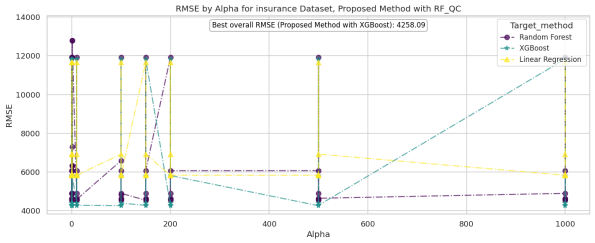


Figure 8.20: RMSE by Alpha for insurance Dataset, Proposed Method with RF_QC. Best overall RMSE (Proposed Method with XGBoost): 4258.09

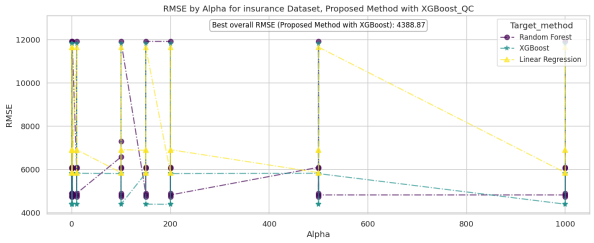


Figure 8.21: RMSE by Alpha for insurance Dataset, Proposed Method with XGBoost_QC. Best overall RMSE (Proposed Method with XGBoost): 4388.87

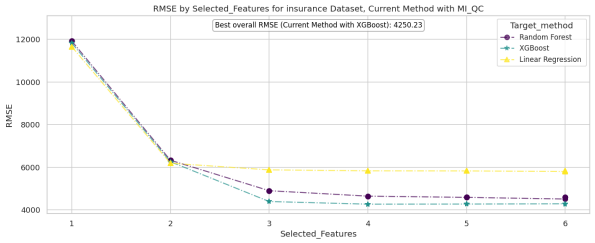


Figure 8.22: RMSE by Selected_Features for insurance Dataset, Current Method with MI_QC. Best overall RMSE (Current Method with XGBoost): 4250.23

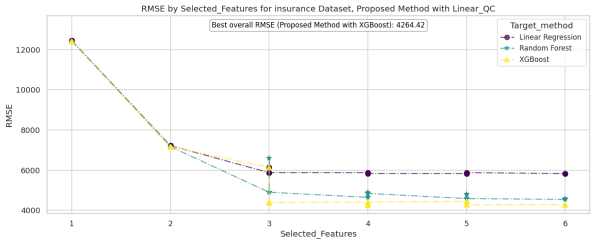


Figure 8.23: RMSE by Selected_Features for insurance Dataset, Proposed Method with Linear_QC. Best overall RMSE (Proposed Method with XGBoost): 4264.42

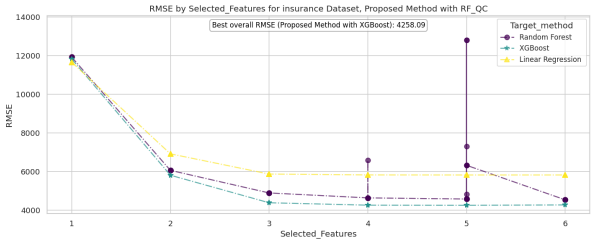


Figure 8.24: RMSE by Selected_Features for insurance Dataset, Proposed Method with RF_QC. Best overall RMSE (Proposed Method with XGBoost): 4258.09

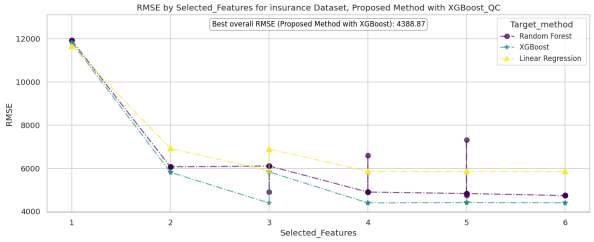


Figure 8.25: RMSE by Selected_Features for insurance Dataset, Proposed Method with XGBoost_QC. Best overall RMSE (Proposed Method with XGBoost): 4388.87

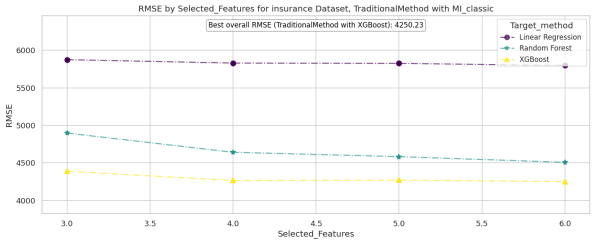


Figure 8.26: RMSE by Selected_Features for insurance Dataset, TraditionalMethod with MI_classic. Best overall RMSE (TraditionalMethod with XGBoost): 4250.23

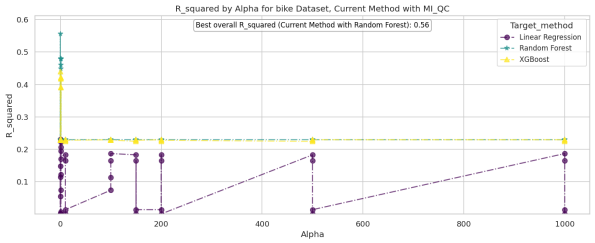


Figure 8.27: R^2 by Alpha for bike Dataset, Proposed Method with RF_QC

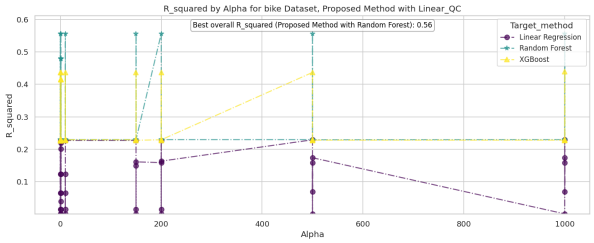


Figure 8.28: R^2 by Alpha for bike Dataset, Proposed Method with XGBoost_QC

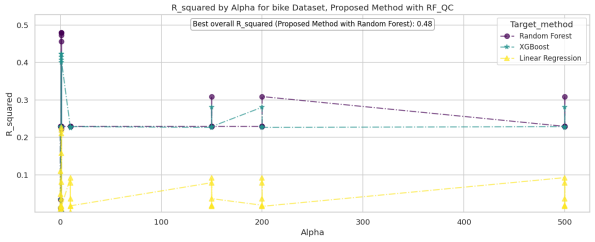


Figure 8.29: RMSE by Alpha for bike Dataset, Current Method with MI_QC

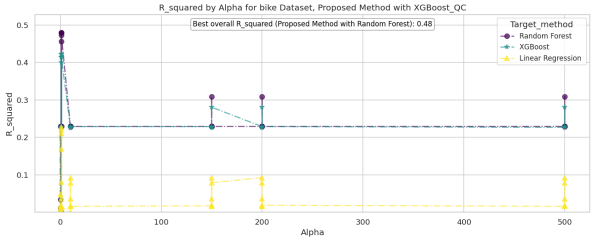


Figure 8.30: RMSE by Alpha for bike Dataset, Proposed Method with Linear_QC

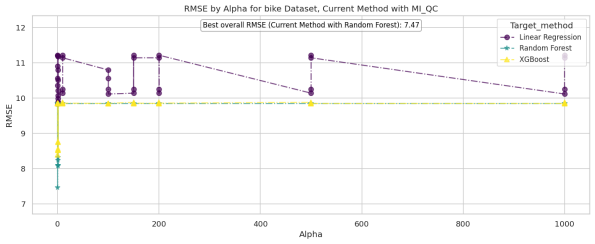


Figure 8.31: RMSE by Alpha for bike Dataset, Proposed Method with RF_QC

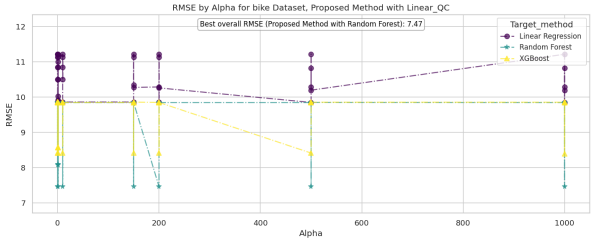


Figure 8.32: RMSE by Alpha for bike Dataset, Proposed Method with XGBoost_QC

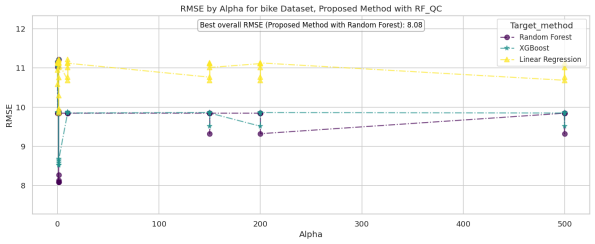


Figure 8.33: RMSE by Selected_Features for bike Dataset, Current Method with MI_QC

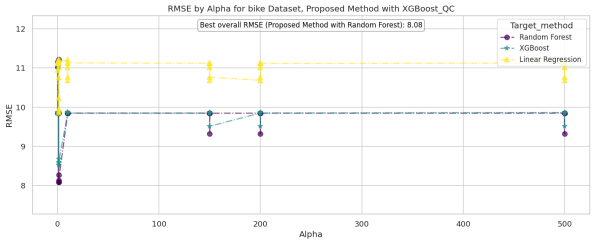


Figure 8.34: R^2 by Alpha for bike Dataset, Current Method with MI_QC

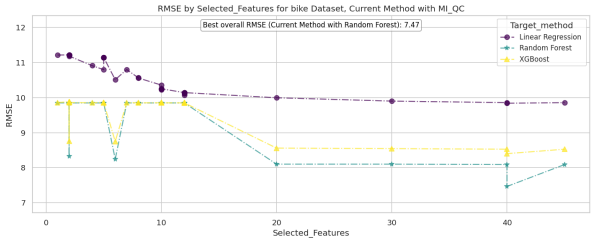


Figure 8.35: R^2 by Alpha for bike Dataset, Proposed Method with Linear_QC

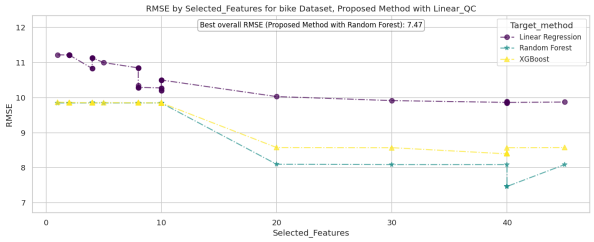


Figure 8.36: RMSE by Selected_Features for bike Dataset, Proposed Method with Linear_QC

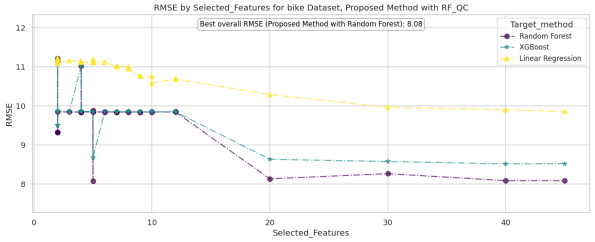


Figure 8.37: RMSE by Selected_Features for bike Dataset, Proposed Method with RF_QC

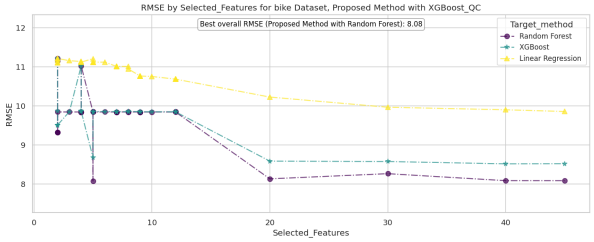


Figure 8.38: RMSE by Selected_Features for bike Dataset, Proposed Method with XGBoost_QC

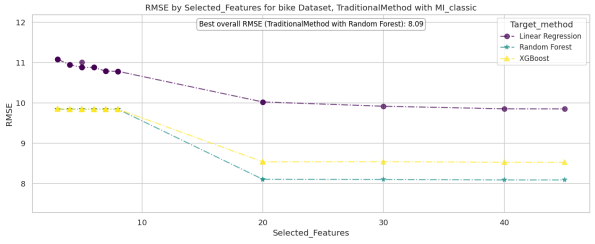


Figure 8.39: RMSE by Selected_Features for bike Dataset, TraditionalMethod with MI_classic