NOVA IMS Information Management School

MDSAA

Master Degree Program in

Data Science and Advanced Analytics

Quantum Computing in Machine Learning

Performance Evaluation and Comparative Analysis

Jaime Marques Simões

Master Thesis

presented as partial requirement for obtaining a Master's Degree in Data Science and Advanced Analytics

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

Quantum Computing in Machine Learning

Performance Evaluation and Comparative Analysis

by

Jaime Marques Simões

Master Thesis presented as partial requirement for obtaining the Master's degree in Data Science and Advanced Analytics, with a specialization in Data Science

Supervised by

Dr. Vítor Santos, Phd, NOVA Information Management School

May, 2025

STATEMENT OF INTEGRITY

I hereby declare having conducted this academic work with integrity. I confirm that I have not used plagiarism or any form of undue use of information or falsification of results along the process leading to its elaboration. I further declare that I have fully acknowledged the Rules of Conduct and Code of Honor from the NOVA Information Management School.

[Lisbon, May 2025]

ABSTRACT

This thesis explores the integration of quantum computing into Machine Learning (ML), aiming to evaluate the effectiveness and limitations of quantum-enhanced models compared to classical approaches. As quantum technologies continue to develop, they offer the potential for solving complex computational problems more efficiently, particularly in fields requiring high-dimensional data processing. A literature review is conducted to contextualize current advancements in Quantum Machine Learning, highlighting theoretical benefits and existing practical limitations. This work investigates whether quantum methods, specifically the Quantum Support Vector Classifier (QSVC), can provide measurable improvements over its classical counterpart. The QSVC, implemented using IBM's Qiskit ML library, replaces traditional kernel functions with quantum kernels computed via quantum circuits and feature maps, embedding classical data into a Hilbert space. The experimental portion of this thesis compares classical and quantum models on two distinct datasets: one from the healthcare domain and another from particle physics. Various parameters are tested in a constrained grid search due to the high computational demands of quantum circuits. Results indicate that while quantum models are significantly more resource-intensive, they can achieve comparable or even improved performance on data that has more complex relations between their variables. However, scalability remains a major challenge. This work concludes that although QSVC does not yet consistently outperform classical models, it offers advantages in specific contexts and lays the groundwork for identifying future use cases where quantum computing may offer a practical advantage in ML.

KEYWORDS

Quantum Machine Learning; Support Vector Classifier; Quantum Support Vector Classifier; Performance Evaluation; Comparative Analysis

Sustainable Development Goals (SDG)

SGD 9: Industry, Innovation and Infrastructure

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

AI	Artificial Intelligence
CNN	Convolutional Neural Network
ML	Machine Learning
MRI	Magnetic Resonance Imaging
PRISMA	Preferred Reporting Items for Systematic Reviews and Meta-Analyses
QCNN	Quantum Convolutional Neural Network
QKNN	Quantum K-Nearest Neighbors
QML	Quantum Machine Learning
QRAM	Quantum Random Access Memory
QSVC	Quantum Support Vector Classifier
QSVM	Quantum Support Vector Machine
QCNN	Quantum Convolutional Neural Network
RQ	Research Questions
SETI	Search for Extraterrestrial Intelligence
SLRQ	Systematic Literature Review Question
SVC	Support Vector Classifier
SVM	Support Vector Machine
VQC	Variational Quantum Classifier

1. INTRODUCTION

This chapter introduces the context, objectives, and relevance of exploring the intersection of quantum computing and machine learning.

1.1. CONTEXT AND PROBLEM IDENTIFICATION

Quantum computing represents a revolutionary leap in computational capabilities, utilizing the principles of quantum mechanics to perform operations that would be impossible for classical computers (Preskill, 2018). The field has already shown its usability in several areas, such as simulating molecular structures and reactions more accurately than classical computers, which can be applied in drug discovery and material science (Peruzzo et al., 2014). With advancements in quantum hardware and algorithms, the domain and accessibility of these resources are increasing (Qiskit | IBM Quantum Computing).

Machine Learning has become an integral part of many industries, with its predictive and descriptive abilities that enable more accurate decision-making and a deeper understanding of the dynamics within those industries (Jordan & Mitchell, 2015). Traditional machine learning (ML) models rely on classical computational resources, which face limitations in processing power, particularly when handling large datasets or complex models. Quantum computing offers a potential breakthrough in overcoming these limitations (Biamonte et al., 2017). One of the open-source resources that can be utilized is IBM Qiskit (Qiskit | IBM Quantum Computing). This Python-based software development kit provides tools for developing and running quantum algorithms.

Most of the research conducted on this topic appears to be theoretical, but there are few practical results. There is a need to evaluate the performance gains and practical applicability of these quantum-enhanced models across various machine-learning tasks. There is a noticeable gap in comprehensive studies that analyze the performance of quantum computing in different machine learning models compared to classical models. Most existing research focuses on theoretical frameworks with limited experimental results (Harrow & Montanaro, 2017). This gap underscores the need for empirical studies to gain a deeper understanding of the actual performance gains and limitations of quantum computing in various machine learning applications. This situation leads to the formulation of the following Research Questions (RQ):

RQ1: What kinds of ML problems benefit from the use of quantum computing?

By answering this question, it will be possible to identify the characteristics of the ML problems that will derive the most significant benefit from quantum computing. For example, the size of the dataset, the type of variables for each situation, and the complexity of the data. By categorizing these problems, we can determine quantum computing's unique capabilities and where they provide an advantage.

RQ2: Quantify the benefits relative to classical computation for these instances.

The answer to this question aims to measure the potential performance improvement in practical terms. The research will involve benchmarking quantum machine learning (ML) models against their corresponding classical versions. Metrics for evaluation will include speed and accuracy of the results. By providing quantitative evidence of the benefits, this research will help measure the advantages postulated in theoretical claims (Preskill, 2018).

1.2. OBJECTIVES

The goal of this research is to evaluate the performance of quantum computing models in various machine learning (ML) problems using IBM Qiskit and then compare their performance and usability against classical models to understand the practical applications of current quantum capabilities.

To achieve the primary goal, the following intermediate objectives were defined:

- Conduct a comprehensive review of existing research on quantum computing in machine learning, with a focus on the practical results obtained.
- Identify the specific problems that can benefit the most from quantum computing by defining the types of predictions and descriptive models that should be used, ensuring that the classical version has a corresponding quantum one that can be used for comparison.
- Study the Qiskit software in detail to effectively apply the models and algorithms.
- Create the models, both classical and quantum, until they are consistent and running.
- Define the evaluation metrics to use for comparison.
- Create a reference guide for the studied class of problems that identifies where quantum computing outperforms classical methods and the extent of its benefits.

1.3. STUDY RELEVANCE AND IMPORTANCE

By providing a comprehensive reference guide for the application of quantum computing in AI, the results of this thesis could make valuable contributions to various fields of study, including medicine, finance, and multiple engineering disciplines. This section examines the specific contributions that these initiatives could make.

Medicine: The integration of quantum computing into machine learning (ML) models could accelerate drug discovery by offering higher computational power and efficiency, optimizing treatment plans, and improving personalized medicine. For example, quantum computing can enhance the development of new drugs by simulating complex molecular structures more efficiently, thereby reducing the time and cost of these treatments (Cao et al., 2018). Additionally, quantum algorithms could aid in identifying patterns in genomic data, leading to a deeper understanding and improved treatment of these disorders (Kösoglu-Kind et al., 2023).

Finance: In this sector, quantum computing has the potential to improve risk management and predictive analytics. The higher speed of analysis could improve fraud detection from financial data, market forecasting, or portfolio optimization. For instance, quantum computing

can evaluate a large number of possible outcomes simultaneously, thereby reducing risk and increasing the return on investment portfolios (Orus et al., 2019). Furthermore, the accuracy of credit scoring models could be improved by analyzing a larger set of financial behaviors and patterns (Rebentrost et al., 2018).

Engineering: In fields such as material science and structural engineering, quantum computing could produce more accurate simulations of physical systems, leading to the development of better materials. Quantum algorithms could simulate molecular and atomic interactions, which is critical for designing new materials with specific properties.

The outcome of this thesis could have broad implications in different areas of society. By contributing to advancements in medicine, quantum computing could contribute to public health, reducing healthcare costs and improving patient outcomes (Cao et al., 2018). Similarly, efficient material design could impact industries from construction to transportation (Bauer et al., 2020). In the tech industry, these advancements could promote innovations and the introduction of new products and services (Preskill, 2018).

2. LITERATURE REVIEW

This section presents a comprehensive review of the literature. First, it covers some of the key concepts in quantum computing, such as qubits and quantum gates, to establish a solid foundation for the topic. In Section 2.2, the Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) protocol is applied to systematically review and synthesize the scientific work on quantum computing and Artificial Intelligence (AI). This way, the state of the art can be determined, and research gaps can be identified.

2.1. QUANTUM COMPUTING

The following sections provide an overview of the core concepts in quantum computing, offering a general introduction to the field. This foundation is essential to understanding how quantum systems function and how they differ from classical computing.

2.1.1. Overview

Quantum computing leverages the principles of quantum mechanics to perform computations that surpass the capabilities of a classical computer. It is essential to review some of the key concepts of quantum mechanics to understand this technology.

Superposition states that a quantum system can exist in multiple states simultaneously. For example, an electron in an atom does not reside in a single position but rather exists in a probability cloud, occupying all possible positions (Preskill, 2018). Because the positions of particles are not deterministic, they are represented by a wave function. When a measurement is made, the wave function collapses to a single outcome, and the superposition of states is lost. The result is determined by the probabilities encoded in the wave function (Nielsen & Chuang, 2010). Quantum interference occurs when the probability amplitudes or different quantum states combine, leading to constructive or destructive interference. This property can be used to amplify correct answers and cancel incorrect ones (Preskill, 2018). Entanglement is a phenomenon in which two particles become linked in such a way that the state of one particle instantaneously influences the state of the other, regardless of the distance between them (Ying, 2010).

Because qubits can exist in a superposition of multiple states, a quantum computation system can process numerous possibilities simultaneously, known as quantum parallelism (Preskill, 2018).

Quantum computation is done with quantum circuits, which are structured sequences of quantum gates. The process begins with an initial state consisting of qubits in a well-defined state. Quantum gates are then applied in sequence to manipulate these qubits, transforming their states in a controlled manner (Nielsen & Chuang, 2010). The outcome of the computation is obtained by measuring the final state of the qubits when the quantum superposition collapses into a classical result (Ying, 2010).

2.1.2. Qubits and Quantum Registers

Qubits are the new version of classical bits. Classical bits can only exist as 0 or 1, while qubits use the principle of superposition to exist in multiple states simultaneously. A qubit can be represented as $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, with α and β being two complex numbers that satisfy the condition $|\alpha|^2 + |\beta|^2 = 1$ (Ying, 2010). So, the fundamental nature of quantum information is described by a vector in a two-dimensional Hilbert space. Qubits also benefit from entanglement, a quantum phenomenon in which the states of multiple qubits become interconnected and cannot be described independently of one another. For example, an entangled state might be represented as $|\psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$, which is known as a Bell state. This means that the system of two qubits simultaneously exists in the two states $|00\rangle$ and $|11\rangle$ with and even probability of 0,5 for each state (Nielsen & Chuang, 2010).

Quantum registers are groups of qubits that collectively form quantum information. Like classical registers, they allow for the simultaneous processing of different states. For a classical computer, n bits enable 2^n possible states, whereas a quantum register can exist in a superposition of these 2^n states, allowing the processing of many states in parallel (Ying, 2010). Mathematically, the state of a quantum register can be represented as a tensor product of individual qubit states. For example, a quantum register with two qubits in states $|\psi_1\rangle = \alpha_1|0\rangle + \beta_1|1\rangle$ and $|\psi_2\rangle = \alpha_2|0\rangle + \beta_2|1\rangle$ can be written as $|\psi_1\rangle \otimes |\psi_2\rangle = (\alpha_1|0\rangle + \beta_1|1\rangle) \otimes (\alpha_2|0\rangle + \beta_2|1\rangle)$, switching to a four-dimensional state space (Ying, 2010).

2.1.3. Quantum Gates

Quantum gates are the fundamental operations that enable quantum computation analogous to classical logic gates in conventional computing. Classical gates operate on binary bits, while quantum gates operate on qubits, allowing quantum gates to perform complex transformations on qubits.

A quantum gate describes a discrete time step of the evolution of a closed quantum system, how the state of the qubits changes over time as different gates are applied. The primary principle of quantum gates is unitarity, which ensures that the total probability of the quantum state is conserved, thereby preserving quantum information throughout the computation. A matrix represents each quantum gate. The matrix U is unitary if it satisfies the condition $U^\dagger U = UU^\dagger = I$, with I being the identity matrix and $U^\dagger$ the Hermitian conjugate (Ying, 2010). This element contrasts with classical gates, which are often irreversible.

One of the most useful gates is the Hadamard gate. Either for a single qubit or for a quantum register, it transforms the defined state into a superposition of all the possible combinations of qubit values. For a single qubit, $H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ with the Hadamard gate being defined in expression 1 (Ying, 2010):

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

In case of it being applied to a quantum register composed of only two qubits, the result is $H \otimes H|00\rangle = H|0\rangle \otimes H|0\rangle = (\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)) \otimes (\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle))$. This operation

results in a superposition of all possible states of the two-qubit system, creating a quantum parallelism that is essential to quantum computing.

2.1.4. Quantum Measurement

Quantum measurement is the process of collapsing the superposition of multiple states into a single defined state after the application of quantum gates to the system. In quantum mechanics, the act of measurement fundamentally changes the state of the system. The qubit takes a definite value, either $|0\rangle$ or $|1\rangle$, each with the corresponding probability determined by the amplitudes of the wavefunction in the superposition state (Nielsen & Chuang, 2010) mathematically, if a qubit is in the state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, where α and β are complex numbers representing the probability amplitudes. The probability of measuring the qubit in the state $|0\rangle$ is $|\alpha^2|$ and for the state $|1\rangle$ is $|\beta^2|$. After the measurement, the information about the original superposition is lost (probabilities of the system being in different states) (Ying, 2010), which limits the amount of information that can be extracted from a quantum system. Techniques such as quantum error correction have been developed to mitigate this issue, aiming to preserve quantum information during the computation process (Preskill, 2018).

2.1.5. Tools

Qiskit is an open-source quantum computing framework developed by IBM (Qiskit | IBM Quantum Computing). It enables the user to design and apply quantum algorithms.

Qiskit Aer (*Exact and Noisy Simulation with Qiskit Aer Primitives*) is the component of the framework dedicated to quantum circuit simulation. It utilizes a classical computing environment that simulates the behavior of a quantum circuit without requiring access to physical quantum hardware. Because of this, these simulations provide a noise-free, controlled environment that is best suited for initial development and testing. It is also a way to be resource efficient, with faster experimentation times. Simulations handle a larger number of qubits than current quantum hardware, making them better for scaling quantum algorithms.

IBM Quantum Platform (*IBM Quantum*) is a cloud-based platform that gives access to IBM's quantum hardware. The qubits used in IBM's quantum computers are made from superconducting materials, which allows for the creation of quantum states at temperatures near absolute zero. These qubits are manipulated using light pulses that control their states, acting as quantum gates (Jurcevic et al., 2021). The current processor in use is called "Osprey," with 433 qubits (IBM, 2022). When a quantum circuit is executed, the processor physically manipulates the qubits according to the sequence of quantum gates in the circuit. Measuring the final state by collapsing the superposition into a classical binary output gives the result of the computation. The inherent noise in quantum systems is still a significant challenge, affecting the results of the computations. Tools like Qiskit Ignis can help manage these issues with correction methods.

While Qiskit is the primary tool used in this project, some other prominent quantum computing 2.1platforms are worth mentioning. Google Cirq (*Cirq*) is designed to run on

Google's quantum computers. Cirq is stronger at performing tasks on quantum algorithms optimized for Noisy Intermediate Scale Quantum devices, having less range than Qiskit on different quantum computing tasks. Microsoft's platform (Microsoft Quantum Development Kit), with the programming language Q#, is focused on the development of quantum algorithms and applications. It is, however, highly integrated with classical cloud computing services, while Qiskit has a deep connection with IBM's quantum hardware, which makes it more advantageous for this research.

2.2. Systematic Literature Review on Quantum Computing in AI

To analyse the current state of research on practical results obtained from using quantum computing on AI and ML problems, a systematic review of the literature needs to be conducted. The goal is to identify existing research on this topic to understand the advancements being made in the field and to determine where a contribution can be made. This review prioritizes studies that present empirical findings rather than theoretical discussions.

PRISMA provides a standardized protocol for conducting systematic reviews. It ensures that the process is reproducible and of high quality (Moher et al., 2009) by considering all relevant studies based on predetermined criteria.

2.2.1. PRISMA Protocol

The PRISMA protocol was developed to standardize the conduct of systematic reviews by researchers, ensure consistency in the process, and minimize bias across different fields of study (Liberati et al., 2009). The protocol is based on a 27-item checklist and a four-phase flow diagram. The main idea is to identify the most appropriate studies that answer specific RQ.

The protocol follows a structured process, beginning with the identification of relevant research studies. This step involves conducting a thorough search across various scientific databases. This search was performed using selected keywords and other strategies to identify an initial pool of potentially relevant studies (Liberati et al., 2009).

Following this, the screening process involves filtering the initial pool of studies based on predetermined inclusion and exclusion criteria, such as the date and language of publication. Studies that do not meet the criteria are excluded from the review. This step ensures that only the most pertinent studies are selected for further consideration (Liberati et al., 2009).

The next stage is eligibility, where the remaining studies are evaluated in more detail. Here, entire sections of the articles are assessed to confirm their relevance and quality. This deeper analysis ensures that all included studies make a valuable contribution (Liberati et al., 2009).

Finally, the inclusion phase involves selecting the studies that will form the core of the systematic review. These studies were analyzed in depth and synthesized to draw conclusions and identify trends in the existing body of research. This final selection serves as the foundation for assessing the current state of knowledge in the field (Liberati et al., 2009).

2.2.2. PRISMA Execution

The goal of this review is to provide a comprehensive analysis of the existing research on quantum computing in AI, with a primary focus on practical outcomes. This systematic research begins by setting the research question (RQ) (Table 2.1), which should be answered by the end of the protocol, ensuring that it addresses the most critical aspects of quantum computing in AI.

Table 2.1 – Systematic Review's RQ

SLRQ1	What is the status of research in this area?
SLRQ2	What kind of AI techniques have been tested with quantum computing?
SLRQ3	What are the limits of the applicability of quantum computing in AI?
SLRQ4	What are the advantages and disadvantages of applying quantum computing to AI?

To answer the proposed RQ, a set of keywords was selected based on the most relevant concepts. These keywords were then used to conduct the searches in the selected databases, ensuring that the review included the most appropriate studies. These keywords (Table 2.2) reflect a focus on techniques, algorithms, and tools relevant to the field.

Table 2.2 – Systematic Review's Keywords

	Quantum Computing	**Artificial Intelligence**
Keywords	Quantum Computing	Artificial Intelligence
	Quantum Algorithms	Deep Learning
	Quantum Machine Learning	Machine Learning
		Artificial Neural Networks

A set of scientific databases were chosen for their broad coverage of literature in fields relevant to this research. They not only have a lot of content but also provide advanced search functionalities, which enable better-targeted searches based on predetermined criteria. To obtain the initial set of studies for this systematic review, a search string was created that looks for specific terms and phrases in the titles, abstracts, and keywords of the scientific papers present in these databases. The search string used was:

("Quantum Computing" OR "Quantum Algorithms" OR "Quantum Machine Learning") AND ("Artificial Intelligence" OR "Deep Learning" OR "Machine Learning" OR "Artificial Neural Networks") AND ("Applications" OR "Challenges" OR "Advantages" OR "Limitations" OR "Results").

Using the operator AND forces the search to find studies that have at least one term for every one of the three substrings, capturing both the core topics of quantum computing and AI, as well as studies that address their practical applications. Another constraint applied to this search was the date of publication for these studies. Only papers published in 2020 or later were considered. This way, the information retrieved is up to date with the current state of the art.

The search was conducted in September 2024 on the scientific information resource databases presented in Table 2.3.

Table 2.3 – Systematic Review's Resource Databases

Resource Database	Resource URL
Scopus	https://www.scopus.com/home.uri
Science Direct	https://www.sciencedirect.com/
IEEE	https://ieeexplore.ieee.org/

With the search string mentioned above for these three sources, the total amount of papers and articles obtained was (n = 9400). Following the steps of the PRISMA protocol, the exclusion criteria serve as a guide to iteratively filter all the scientific studies obtained from the initial search string. The first of these to be applied were the duplicate papers. After using this filter, the number of studies left was (n = 8538). The following criterion was applied: removing papers that have no direct connection to their corresponding file. This culling process means that when exporting database resources, the column corresponding to a link or URL directing to the paper is left empty. This procedure resulted in having 172 studies removed, moving to the next step with (n = 8366) studies. To ensure that the retained papers have had some visibility in the literature, studies published before 2024 and with no citations were removed in the next step. Papers published in 2024 were removed from this process because those without citations might be due to the short time that specific study has been publicly available rather than a lack of relevant content. After this step, the amount of papers remaining was (n = 7346). Because the number of studies in the pool remains vast, it was decided to change the lower boundary for the publication date from 2020 to 2021 rather than setting a lower boundary. Thus, the remaining studies are more up-to-date with these topics, and we can reduce the number of papers in the current pool. After that, the number of remaining papers was filtered down to (n = 6668). The final step for this section was to filter studies based on the type of publication. Papers published in books, consulting articles, or websites were removed as they do not qualify as academic or scientific publications. After this final step, the number of studies remaining was (n = 6149).

Table 2.4 – Systematic Review's Inclusion and Exclusion Criteria

Inclusion Criteria	Exclusion Criteria
Any paper that shows evidence of the use of quantum computing in AI models	Duplicate papers and not in the English language: (Exclusion #1)
Papers published in journals and presented at conferences	Papers with no direct connection to their corresponding file: (Exclusion #2)
Paper will be published between 2021 and 2024	Papers published before 2024 that have no citations: (Exclusion #3)
Papers published before 2024 need to have at least one citation	Papers published before 2021: (Exclusion #4)
	Non-academic or non-scientific papers (e.g., websites, newspapers, consulting articles, books, citations): (Exclusion #5)
	Papers with titles outside the scope of this research: (Exclusion #6)
Papers that, on a deeper analysis, do not make significant contributions to answering the SLRQ introduced	Abstracts not relevant to this research: (Exclusion #7)
	Papers with content that help answer the initial SLRQ: (Exclusion #8)

The following steps are the most significant and pinpoint the most relevant articles best suited to answer the SLRQ proposed. A filtering method was applied to all the titles, where the corresponding studies were removed if none of the subsequent keywords were present:

'Quantum Machine Learning', 'Quantum Computing', 'Quantum Neural Networks', 'Quantum Convolutional Neural Networks', 'Quantum Support Vector', 'Quantum K-NN', 'Quantum K-Nearest Neighbors', 'Quantum Clustering', 'Quantum vs Classical', 'Hybrid', 'Image Classification', 'Time Series Prediction', 'Optimization', 'Comparison', 'Comparative', 'Results', 'Practical Results', 'Empirical', 'Analysis', 'Quantum Algorithms', 'Quantum AI', 'Quantum Artificial Intelligence', 'Quantum-enhanced', 'Quantum Deep Learning', 'Quantum Models', 'Quantum Support Vector Machine', 'Performance Gains' 'Quantum Networks', 'Performance Comparison', 'Empirical Study', 'Practical Study', 'Benchmarking', 'Evaluation', 'Quantum Computing Models', 'Quantum Algorithm Development', 'Quantum Optimization'.

This action removed 4612 studies from the pool, which still leaves (n = 1537) papers. After this, the abstracts were analyzed, and those that did not show relevant contributions or that were out of the scope of this research were removed, leaving (n = 133) studies remaining.

From this final set, a deeper analysis of the contents of each paper was conducted to obtain the final set that will answer our SLRQ. From this, we harvested (n = 18) papers.

The entire process of selecting the articles that present the state of the art for the proposed RQ is represented in the following flowchart:

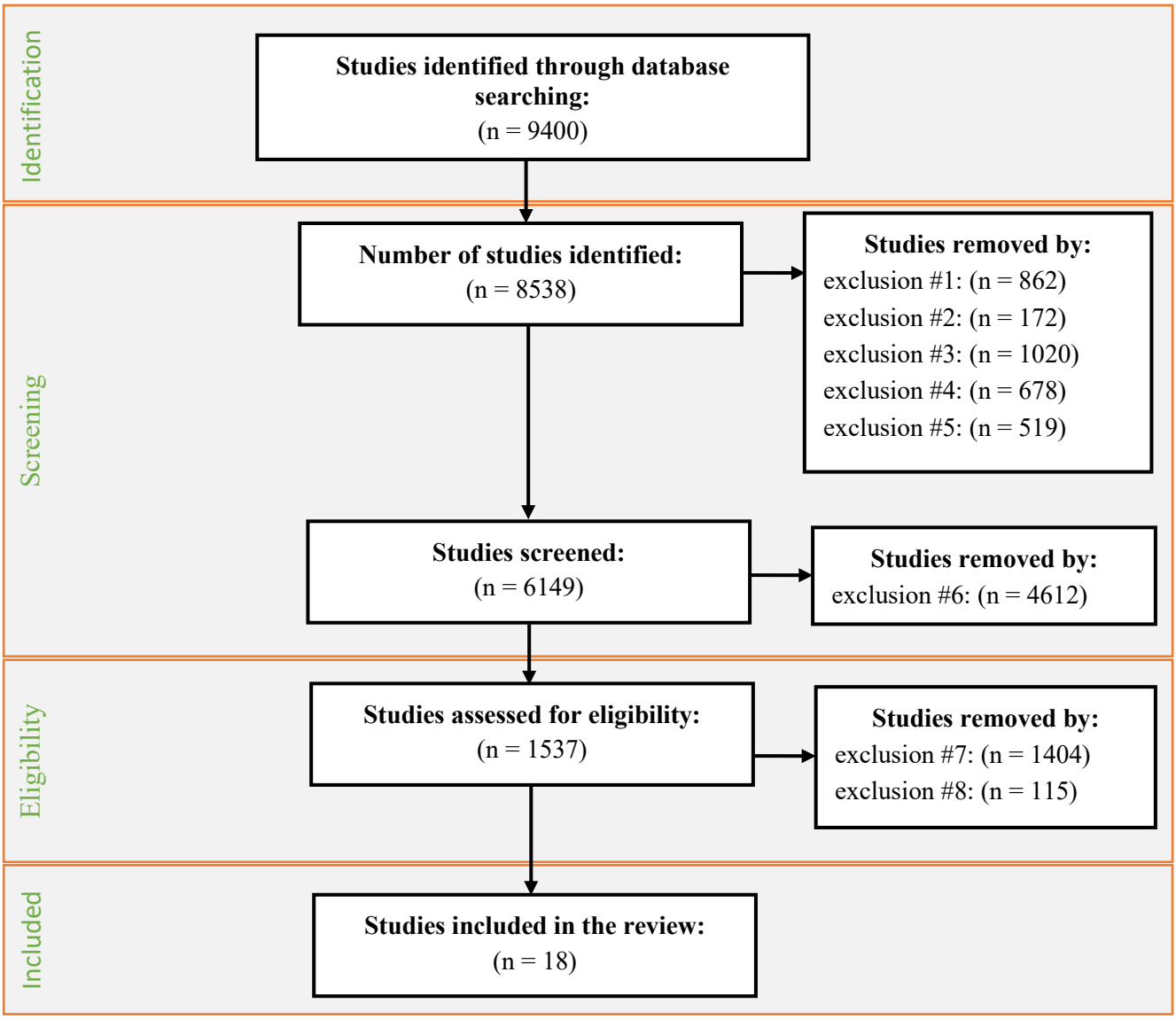

Figure 2.1 - PRISMA Execution Flow Chart (adapted from (Moher et al., 2009))

From an initial pool of studies retrieved through a systematic search of multiple databases, 18 studies were identified as making a direct contribution to the focus of this research. These studies offer valuable insights into current developments and barriers in the field.

The table below summarizes the main findings and methods of each of the selected papers. This review served to identify existing gaps in the literature.

Table 2.5 – Contributions of each of the selected studies

#	Authors	Article	Contribution	Publication Type	Uses Qiskit
#1	(Alharbi et al., 2021)	Image Classification Based on Quantum Machine Learning	Utilizes combinations of quantum and classical algorithms, as well as Neural Networks and Convolutional Neural Networks, to make predictions on the MNIST dataset (image classification) with improved results compared to purely classical methods. Steps are defined to classify the CIFAR-10 dataset, but no results are obtained due to technological constraints.	Conference Paper	✓
#2	(Kashif & Al-Kuwari, 2022)	Demonstrating Quantum Advantage in Hybrid Quantum Neural Networks for Model Capacity	Performs comparative analysis between hybrid Quantum Neural Networks (QNN) and classical ones in multiclass classification problems. Both are constructed with approximately the same number of trainable parameters. For the Hybrid QNN, two different encoding techniques are applied: amplitude encoding and angle encoding. The results give the hybrid models an advantage in all instances tested.	Conference Paper	✗
#3	(Tantawi et al., 2023)	Exploring the Power of Quantum Convolutional Neural Networks for Brain MRI Image Classification	Proposes a Quantum Convolutional Neural Network (QCNN) model for brain image classification, combining classical and quantum computing techniques. Using image resizing, data normalization, and augmentation, the traditional CNN extracts features, and the quantum component handles classification. Results from a publicly available real brain Magnetic Resonance Imaging (MRI) dataset show an improvement over the traditional CNN counterpart.	Conference Paper	✓

#	Authors	Article	Contribution	Publication Type	Uses Qiskit
#4	(Ceschini et al., 2022)	Hybrid Quantum-Classical Recurrent Neural Networks for Time Series Prediction	Construction of a hybrid Quantum Recurrent Neural Network that combines two Short-Term Memory layers with a Variational Quantum layer for predicting parameters on a problem related to renewable energy (Time Series Problem). This model outperforms classical methods by exhibiting better generalization (i.e., being more resistant to noise and unstable conditions) and faster convergence.	Conference Paper	✓
#5	(Ghosh et al., 2024)	Quantum vs. Classical: A Rigorous Comparative Study on Neural Networks for Advanced Satellite Image Classification	Compares a classical convolution neural network model with a hybrid CNN model that trains on data converted to the quantum format via a quantum feature map on the RDI-CB256 dataset (Satellite Images). It only utilizes the simulator provided by Qiskit rather than quantum hardware. Despite achieving similar results in terms of accuracy, the hybrid model exhibits stability much sooner and exhibits better generalization.	Conference Paper	✓
#6	(Özpolat & Karabatak, 2023)	Exploring the Potential of Quantum-Based Machine Learning: A Comparative Study of QSVM and Classical Machine Learning Algorithms	Takes a diabetes dataset from the UC Irvine ML Repository and applies both a Quantum Support Vector Machine model and different classical models like Random Forest and K-Nearest Neighbours. It's a binary classification problem. Principal Component Analysis is applied across the various models to reduce the dimensionality of the problem. The accuracy obtained by QSVM is lower than classical methods. The authors argue this is due to the hardware limitations encountered in this study.	Conference Paper	✓

#	Authors	Article	Contribution	Publication Type	Uses Qiskit
#7	(Reka et al., 2024)	Exploring Quantum Machine Learning for Enhanced Skin Lesion Classification: A Comparative Study of Implementation Methods	This study explores QML in multiclass skin disease classification using the HAM10000 dataset. It combines quantum computing with classical ML by implementing various combinations of qubit rotation encoding and decoding using Pauli X, Y, and Z gates. The research involves data augmentation techniques to improve model performance. A Quantum Convolutional Neural Network is developed, achieving a higher classification accuracy compared to classical pre-trained models such as ResNet50 and DenseNet. A Quantum Support Vector Classifier was also tested; however, the results were not as favorable.	Journal Article	✓
#8	(Zeguendry et al., 2022)	Quantum Machine Learning: Practical Cases	Presents a comparative analysis of classical with quantum models by implementing QCNN and a Variational Quantum Classifier on a quantum computer. The QNN is applied to data for handwritten digit recognition using a quantized convolutional layer, outperforming its classical counterpart in terms of accuracy and efficiency. The VQC is implemented on the Iris dataset with amplitude encoding. It also demonstrates better accuracy compared to classical classifiers.	Conference Paper	✗

#	Authors	Article	Contribution	Publication Type	Uses Qiskit
#9	(Upama et al., 2024)	A Comparative Study of Classical and Quantum Algorithms for Heart Disease Prediction Using Patients' Vital Signs	The authors investigate the comparative effectiveness of classical and quantum algorithms for predicting heart disease (a binary problem). They utilize a public dataset, the "Heart Disease Health Indicators Dataset," to train and evaluate the models. For SVM, the quantum version outperforms the classical one; however, for NN, this is not the case. Accuracy values increase with the increase of the sample size.	Conference Paper	✓
#10	(Wiedmann et al., 2023)	An Empirical Comparison of Optimizers for Quantum Machine Learning with SPSA-based Gradients	This study proposes an optimization approach for Variational Quantum Circuits by combining Simultaneous Perturbation Stochastic Approximation (SPSA)- based gradient estimates with optimizers such as Adam, AMSGrad, and RMSProp. The results show an improvement in convergence speed and error reduction compared to traditional methods. SPSA-AMSGrad consistently performs better, particularly in noisy environments.	Conference Paper	✗
#11	(Yadav et al., 2023)	Hybrid Quantum-Classical Convolutional Neural Network for Allen Telescope SETI Image Classification	A Hybrid Quantum-Classical CNN for classifying the telescope images is presented. The data, presented as 2D spectrograms, passes through a QCNN used to extract quantum features. These features are then processed in a classical CNN for classification. The quantum layer includes techniques such as angle embeddings and decoding with Pauli-Z gates. The model is tested on a small portion of a publicly available dataset and demonstrates slightly better results than classical models for binary classification tasks.	Conference Paper	✗

#	Authors	Article	Contribution	Publication Type	Uses Qiskit
#12	(Matic et al., 2022)	Quantum-classical convolutional neural networks in radiological image classification	Applies Hybrid CNN to both 2D and 3D medical images (datasets “Breast Ultrasound” and “LIDC-IDRI”). The methodology includes a quantum convolutional layer that replaces a classical one, using encoding schemes. The Hybrid CNN achieves similar results to those of a classical CNN (in a binary classification problem) while using fewer trainable parameters.	Conference Paper	✗
#13	(Jadhav et al., 2023)	Quantum Machine Learning: Scope for real-world problems	The authors explore the application of several quantum algorithms and compare them to their classical counterparts, including K-Nearest Neighbors and Quantum K-Nearest Neighbors. The process involves encoding the dataset into quantum states and using quantum measurements to determine the nearest neighbors. The authors conclude that QML is more effective for text and image classification problems, especially for smaller datasets.	Conference Paper	✓

#	Authors	Article	Contribution	Publication Type	Uses Qiskit
#14	(Poggiali et al., 2024)	Quantum Clustering with k-Means: A Hybrid Approach	The study introduces three versions of a quantum k-means model (hybrid approach). The first one performs a single Euclidean distance computation, the second version computes the distance to all centroids in parallel, and the last one calculates the distance to all records simultaneously. It also introduces the Inverse Stereographic Projection (ISP) as a preprocessing step to deal with issues related to distance computation. Classical data is encoded into a quantum format via the FF-QRAM algorithm. Similar clustering outcomes are obtained for the quantum and classical versions, but the current quantum hardware available imposes firm limits on the performance that can be achieved.	Journal Article	✓
#15	(Maldonado-Romo et al., 2024)	Quantum K-Nearest Neighbors: Utilizing QRAM and SWAP-Test Techniques for Enhanced Performance	This study introduces a QKNN algorithm that utilizes Quantum Random Access Memory, which enables simultaneous access to multiple memory cells. Grover's algorithm is within the QKNN to increase classification accuracy. The SWAP-Test determines the similarity between the quantum states that represent each data, serving as a metric. The complete algorithm is tested on the Iris and MNIST datasets, yielding similar accuracy results for various QRAM sizes and encoding methods.	Journal Article	✓
#16	(Senokosov et al., 2024)	Quantum Machine Learning for image classification	Builds a Hybrid Quantum Neural Network, where each quantum layer's output leads to a classical layer. The model outperformed its classical counterpart on the MNIST dataset and obtained similar results on the Medical MNIST and CIFAR-10 datasets while using fewer trainable parameters.	Journal Article	✗

#	Authors	Article	Contribution	Publication Type	Uses Qiskit
#17	(Kobayashi et al., 2022)	Overfitting in quantum Machine Learning and entangling dropout	This study introduces the entangling dropout method, which randomly removes entangling gates in a quantum circuit during training, given by two parameters: the layer dropout ratio and the gate dropout ratio. These parameters need to be optimized. The effect is comparable to traditional L1 and L2 regularization techniques, requiring fewer training iterations.	Journal Article	✗
#18	(Chen et al., 2020)	Quantum Convolutional Neural Networks for High Energy Physics Data Analysis	The authors construct a Hybrid QNN to be tested on simulated data from High-Energy Physics experiments. Particle images and paths are simulated that present different features used for a classification problem. For muons and protons, the Hybrid CNN exhibits higher accuracy and faster convergence times, whereas for other particles, it performs similarly to the classical models.	Journal Article	✗

2.2.3. PRISMA Results Analysis

With the selection process and reviews for the corresponding studies concluded, we can now analyze the contributions made by each of these studies and how they address the proposed SLRQ.

SLRQ1 - What is the status of research in this area?

The research on quantum computing in AI is at a promising stage of development. The field is primarily characterized by proof-of-concept studies that aim to showcase potential performance improvements offered by quantum algorithms, though practical implementations remain limited.

Of these practical implementations presented in the literature, most focus on hybrid models, which combine quantum and classical computing. For example, a study by Ghosh et al. (2024) conducted a comparative analysis between classical and hybrid quantum-classical CNNs for satellite image classification. They found that while the hybrid model exhibited similar accuracy to the classical model, it demonstrated faster convergence and greater stability. In another study involving image classification tasks, Tantawi et al. (2023) explore the use of

QCNN for brain MRI image classification. By combining classical preprocessing methods, such as data augmentation and normalization, with quantum processing for the classification task, the model introduced showed an improvement in results over its classical counterpart. Both studies and others of a similar nature highlight a potential advantage of quantum computing in specific image-processing tasks. However, they also note that these experiments relied heavily on simulated environments rather than real quantum hardware, which is a limiting factor.

Another potential benefit is reducing the computational load in AI applications, as exemplified by the study by Matic et al. (2022). This study employs a hybrid quantum-classical CNN for medical image classification, specifically in breast cancer ultrasound applications. The results demonstrate that quantum-enhanced models can match the performance of classical models while using fewer parameters, indicating improved efficiency. The paper (Jadhav et al., 2023) employs several quantum algorithms and compares them to their classical counterparts in various types of problems, providing a general overview of the field. Besides the potential of applying quantum and hybrid models to image classification problems, it also mentions text classification as a context best suited for the field.

SLRQ2 - What kind of AI techniques have been tested with quantum computing?

Quantum computing has been applied across various AI techniques, with a significant focus on supervised learning models, optimization, clustering, and classification. The primary approach in the literature has been to adapt classical AI algorithms into hybrid quantum-classical ones, aiming to leverage the properties of quantum mechanics to enhance the capabilities of the models.

A significant area of focus has been on quantum-enhanced neural networks, particularly for image classification tasks. In the paper by Chen et al. (2020), a hybrid CNN is developed to classify high-energy physics data, simulating particle collision images from experimental datasets. By integrating quantum convolutional layers into a classical neural network, the model outperformed its classical counterpart in specific categories of a classification problem, demonstrating the advantages of quantum convolutional architectures. Quantum computing has also been explored in clustering techniques such as quantum k-means. In the paper by Maldonado-Romo et al. (2024), a classifier that utilizes QRAM and angle encoding is introduced to manage data more efficiently in a quantum context. The SWAP-Test was used for similarity comparison, achieving performance improvements over previous QKNN implementations.

Additionally, time-series prediction models have been explored with quantum enhancements, primarily through quantum recurrent neural networks (QRNNs). In the paper by Ceschini et al. (2022), the authors employed a hybrid QRNN model for time-series prediction in renewable energy forecasting, incorporating quantum layers with classical short-term memory layers. This model showed better generalization and higher resistance to noisy data. Quantum computing has also been applied in optimization algorithms, which are very important across various domains. In the paper by Wiedmann et al. (2023), a quantum optimization technique utilizing variational quantum circuits is developed to optimize neural network weights. This study demonstrated faster convergence rates and reduced error compared to classical

optimization algorithms, such as gradient descent. These results are promising, particularly for ML tasks that require high computational power for parameter tuning.

SLRQ3 - What are the limits of applicability of quantum computing in AI?

One of the key limitations is the state of quantum hardware, which restricts the scalability and efficiency of quantum algorithms in AI. Current quantum devices, often in the Noisy Intermediate-Scale Quantum (NISQ) stage, are limited in qubit count and face significant stability issues. Poggiali et al. (2024) present three variations of a hybrid quantum k-means clustering algorithm that leverages quantum parallelism to enhance the cluster assignment step. While the models demonstrate the theoretical advantage of quantum parallelism in computing distances between records and centroids, the study also highlights that current quantum hardware lacks the robustness necessary for efficient clustering on larger datasets. Only the simplest algorithm, when implemented, was able to produce reliable results on a real quantum device, and even this required significant computational resources.

Resource accessibility also poses a considerable barrier to quantum AI. Quantum computing infrastructure remains expensive and broadly accessible only to well-funded institutions, limiting widespread experimentation. Paper (Zeguendry et al., 2022) explores the use of QCNN and VQC in handwritten digit recognition and compares them to classical CNN. The study found that QNN and VQC demonstrated strong classification accuracy and efficiency; however, it notes that this research required substantial computational resources, especially since real quantum hardware was not always available, necessitating simulations. This reliance on high-resource environments restricts the ability to explore quantum AI.

Quantum noise and instability are also significant limitations. Quantum systems are susceptible to external interference, resulting in high rates of computational errors. In their paper, Ceschini et al. (2022) observed that while quantum layers enabled improved model generalization in volatile conditions, the noise and instability inherent in quantum systems limited the model's performance. Without effective error correction techniques, which are still under development and require additional qubits, quantum models struggle with reliability.

Scalability issues also present a constraint on quantum AI. Quantum models have a limited capacity to handle the large datasets typically required in machine learning (ML), making it challenging to apply quantum models to data-intensive applications. In the Paper by Ghosh et al. (2024), despite demonstrating the superiority of the quantum model, they faced limitations when attempting to scale to larger, more complex datasets. This constraint restricts the scalability of quantum-enhanced models.

SLRQ4 - What are the advantages and disadvantages of applying quantum computing to AI?

Quantum computing offers several promising advantages for machine learning (ML) models, including increased data processing speed, enhanced model optimization, and the ability to handle high-dimensional problems. However, practical limitations remain, primarily due to hardware constraints, noise, and scalability issues.

One primary advantage of quantum computing is its potential for enhanced model capacity and increased data processing speed. Quantum-influenced models can process high-dimensional data more efficiently and explore larger solution spaces than classical models. The paper by Kashif & Al-Kuwari, 2022 explores hybrid quantum neural networks that utilize quantum features, such as amplitude and angle encoding, to enhance model capacity in multiclass classification. Their findings show that Hybrid QNN consistently outperforms classical neural networks with similar parameters, demonstrating a superior ability of the quantum models to manage complex data. Another advantage is the potential of quantum computing for better optimization and generalization. The paper authored by Jadhav et al. (2023) analyzes several quantum algorithms, including QKNN, and compares their performance to that of their classical counterparts in various classification tasks. By encoding datasets into quantum states and utilizing quantum measurements to identify the nearest neighbors, the study reveals that quantum models exhibit advantages, most notably in image and text classification tasks, particularly for smaller datasets, where quantum models can achieve high accuracy with fewer data points.

Despite these advantages, quantum computing also faces several disadvantages that limit its application. One of the primary challenges is current hardware limitations that restrict the performance and speed of quantum algorithms in some instances. In the paper written by Özpolat & Karabatak (2023), its performance still falls behind that of classical models, although the QSVM model achieves competitive accuracy. This factor is primarily attributed to limitations in quantum hardware, including restricted qubit availability and reliance on simulators rather than actual quantum devices. The study highlights that quantum models may become more competitive once hardware limitations are addressed. However, until then, these restrictions constrain the speed and consistency of quantum algorithms compared to classical methods. Another significant limitation is scalability. Current quantum systems are unable to handle the large datasets commonly required for these applications. Paper (Upama et al., 2024) explores the use of quantum-enhanced models for heart disease prediction, revealing that while quantum models performed well on smaller datasets, they struggled to scale up effectively as the dataset size increased. This constraint limits the application of quantum computing to small, controlled datasets, preventing its use in large-scale, real-world applications.

3. METHODOLOGY

The methodology for this study is designed to explore and assess the potential of quantum computing in machine learning (ML). By dividing it into four phases (Exploration, Conceptual, Testing, and Conclusive), the study progressively builds from existing research to practical testing and analysis. Each phase is designed to address specific objectives that contribute to an evaluation of both classical and quantum-based ML models.

The initial phases focus on setting a strong foundation by analyzing relevant literature, identifying appropriate algorithms and datasets, as well as developing model architectures. Then, the application is extended to practical implementations, where the constructed models are tested and evaluated in controlled experiments. The final phase analyzes the results, with a special focus on comparing the effectiveness and potential applicability of quantum models.

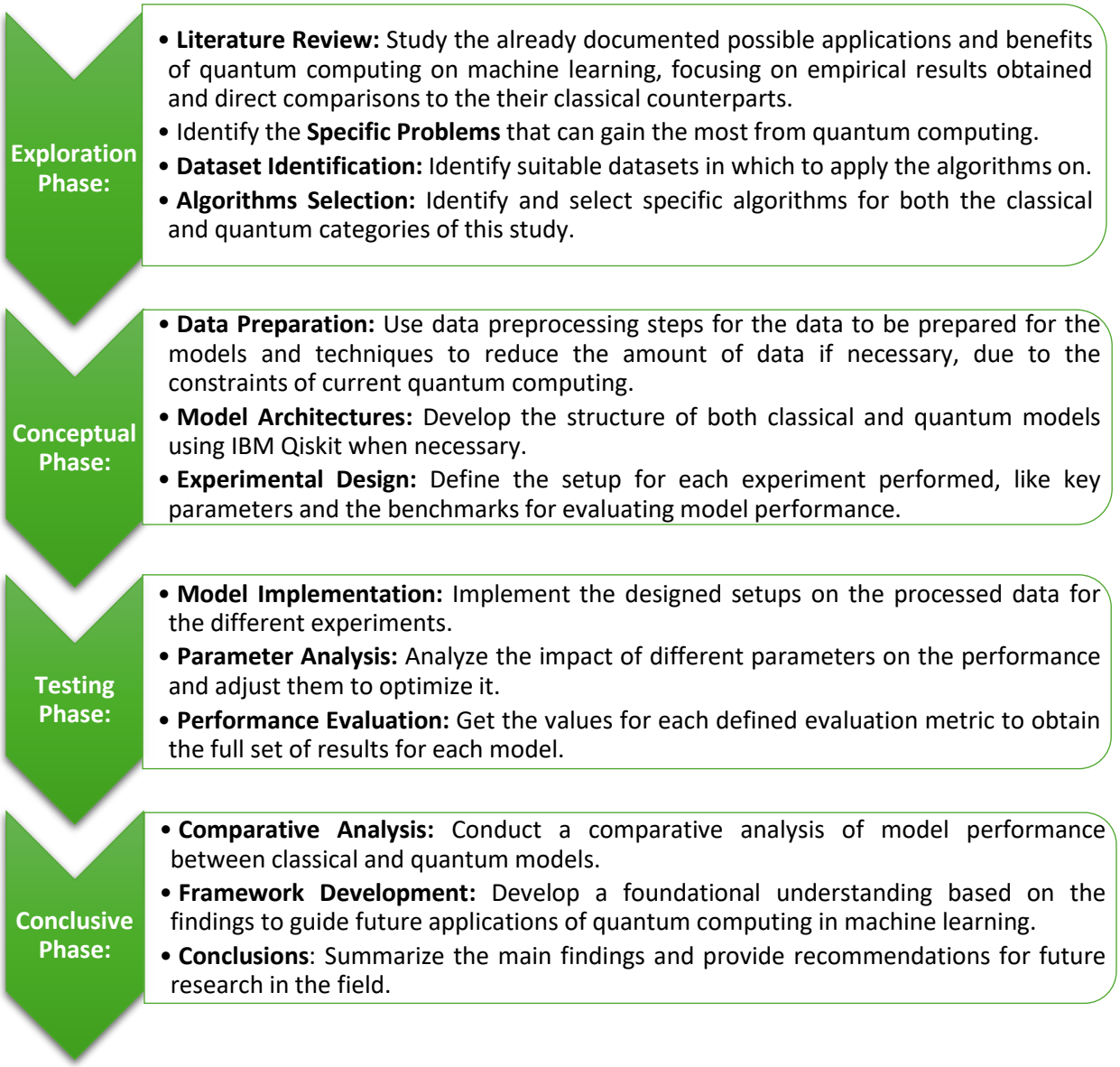

Figure 3.1 - Phases of the Methodologies

3.1. RESEARCH PHASES

In the following subsections, each of the methodology phases is explained in more detail, highlighting the specific steps required. Each phase plays a distinct role in this study, culminating in a comparative evaluation that aims to inform future applications in the field.

3.1.1 Exploration Phase

The primary goal of this phase is to establish a foundation for the study by identifying models, datasets, and insights present in the state-of-the-art field from existing literature on quantum computing in ML. The literature review focuses on current research regarding the potential benefits of applying quantum computing to machine learning models and specific areas where these benefits can be most noticeable, both for purely quantum models and hybrid quantum-classical models. Based on this literature review, the most pertinent algorithms are selected, both classical and their closest quantum counterparts, to facilitate a direct comparison of the results. Alongside the chosen models, suitable datasets must be selected for application, creating a robust testing environment that will evaluate the strengths and weaknesses of quantum versus classical approaches.

3.1.2. Conceptual Phase

In this phase, the focus shifts to preparing the data and designing the models that will be evaluated later. Data preparation involves preprocessing steps to ensure the data is ready for analysis by both classical and quantum models, including handling missing values, normalizing features, and applying dimension reduction techniques. Reducing the dimensionality of the data is especially important, as current quantum capabilities have significant constraints on the amount of information that can be processed due to limitations on the number of qubits. Model architecture development involves constructing the models. The classical models are designed using standard machine learning libraries, while the quantum-enhanced ones are created using IBM Qiskit when necessary. This phase also involves setting key parameters and defining benchmark metrics used in the comparative section of the study.

3.1.3. Testing Phase

First, both the classical and quantum models are implemented on the selected, processed datasets chosen in the experimental setup, with their defined evaluation metrics. For quantum models, IBM Qiskit is used to run the experiments using a simulator. The classical counterparts are run in parallel to provide a direct performance baseline. Once the models are functional, parameter analysis is conducted to understand the impact of different hyperparameters on performance, allowing them to be optimized. After the hyperparameters are optimized, the results from each model are collected with the specified metrics. These results are the data used in the next phase for the comparative analysis between them. By

carefully testing each model variant, this phase ensures a reliable reflection of how classical and quantum models perform in real-world situations.

3.1.4. Conclusive Phase

The collected performance results are analyzed in a comparative context, highlighting the strengths and weaknesses of both classical and quantum models. This analysis aims to determine whether quantum models, despite current technological constraints, can offer practical advantages over classical approaches in certain specific scenarios. Following this analysis, the conclusions drawn serve to provide a direction for future implementations of quantum models in ML tasks. Finally, a summary of the main findings of this study, with an emphasis on the potential and limitations of applying quantum computation in machine learning, including possible improvements and areas for future research.

4. EMPIRICAL STUDY

The empirical phase of this thesis transitions from theoretical exploration to practical implementation, focusing on preparing the collected data and building the various models that will be evaluated using specified metrics. This chapter is designed to outline the steps involved in setting the domain of the work at hand, selecting suitable data accordingly, and constructing experimental pipelines for all models developed, including both classical and quantum domains.

4.1. RESEARCH PROBLEM UNDERSTANDING

This section begins with the selection of a relevant dataset that reflects the challenges highlighted in the literature review. The medical field is prioritized in this search due to its prevalence in the papers studied and the availability of well-structured datasets related to this field, such as those on diagnostic images and disease prediction. It also needs to be compatible with both classical and quantum models, taking into account the limitations of current quantum capabilities.

Once a dataset is identified, its content will be thoroughly analyzed to ensure that it aligns with the requirements and capabilities of classical and quantum machine learning algorithms. The exploration phase will involve examining the various types of features present and taking the necessary steps to process them, such as encoding categorical ones or applying normalization and scaling techniques to numerical features. If the dataset involves images, it requires additional processing steps, such as resizing and edge detection. Another critical step in this section is how noisy and missing data are handled, such as missing entries in important features or outliers that impact the dataset's statistics. For the quantum domain, specific processing steps need to be taken, such as amplitude or angle coding. These encodings require specific data formats, which are created in this phase. Additionally, Subsampling or dimensionality reduction may be necessary due to quantum computational constraints.

After all the data processing is complete, the dataset will be split into training, validation, and test sets. For quantum experiments, smaller sets may be necessary due to computational constraints, with an emphasis on cross-validation techniques to maximize the amount of data available. The focus then shifts to building the models. The classical ones serve as the baseline, constructed using standard libraries like scikit-learn or TensorFlow. For the quantum models, IBM Qiskit is utilized to design algorithms. Two potential quantum approaches can be explored: fully quantum models, where the entire computational pipeline utilizes quantum gates, and hybrid quantum-classical models, which use a combination of classical processing followed by quantum computation.

To evaluate the models, a set of predefined metrics are used, namely the F1 score and computational efficiency in the form of runtime comparisons.

4.2. Data Selection and Understanding

This section focuses on understanding the datasets selected for the empirical study. The goal is to ensure that the data is compatible with both classical and quantum-enhanced ML models, considering the limitations of current quantum simulators.

4.2.1. Experiment 1 - Heart Disease

The first dataset chosen is called “Heart Disease” (Janosi et al., 1989), which contains a subset of 13 features taken from the original dataset that had 76 attributes. Integrated into the Health and Medicine area, this is a multiclass classification problem with characteristics of different types, including categorical and numerical values. It contains 303 instances, so the results originating from this serve as a benchmark for what should be expected of smaller datasets. The target variable refers to the presence of heart disease in the patient, defined as an integer value from 0 (no presence) to 4. It can also be adapted to a binary classification scenario if we consider zero and then 1 for all the other options (1, 2, 3, and 4), meaning distinguishing only between the presence or absence of disease in the patient.

4.2.2. Experiment 2 - MiniBooNE

The second dataset, titled “MiniBooNE particle identification” (Roe, 2005a), is situated in the field of physics. It contains data taken from the MiniBooNE experiment (*Fermilab | Science at Fermilab | Experiments & Projects | Intensity Frontier | MiniBooNE*) at Fermilab. This experiment aimed to detect oscillations of muon anti-neutrinos into electron anti-neutrinos. For this dataset in particular, the content consists of 50 attributes for each instance of a particle detection that characterizes this interaction (the meaning of the features themselves is not specified), all of which are of the numerical type. The target variable is also of a classification type. Nonetheless, this time, only a binary situation is available, as the two possible values are the interaction being either an electron neutrino (signal) or a muon neutrino (background noise). In contrast to the first dataset, this dataset comprises 130065 instances, which, alongside 50 features instead of 13, makes it a significantly larger dataset with more dimensions and, consequently, more relations between the variables available. Therefore, the results obtained from this will serve as a benchmark for what can be expected from larger and more complex datasets.

4.3. Data Exploration and Processing

Before applying models to the data, the datasets undergo a series of preprocessing steps to ensure data quality, consistency, and compatibility with both classical and quantum approaches. This phase involves first understanding each of the variables, including their statistical distributions and general contributions to the dataset. Then, a more technical analysis is conducted, such as looking for outliers, handling missing values, normalizing numerical features, and encoding categorical variables. For the quantum-enhanced models, specific criteria for the data must be met and are done independently of the classical section.

Given the distinct characteristics of the two datasets, specific preprocessing steps are applied to each, as detailed in the following sections.

4.3.1. Experiment 1 - Heart Disease

As mentioned earlier, this dataset comprises 13 features and contains 303 entries. It involves a multiclass classification problem that can also be converted into a binary classification problem. Table A1 in the appendix displays the general statistics for each variable in the dataset.

The first feature is “age,” with no missing values and no outliers to note. The values for this attribute were normalized between 0 and 1 with the *MinMaxScaler* function (*MinMaxScaler*). Then we have the attribute “sex,” which has no missing values, where 0 represents female and 1 represents male. It is to be noted that the count of males is approximately double that of women. Feature “cp” indicates the type (if any) of the chest pain felt by the individual (Janosi et al., 1989). It doesn’t present missing values and was also normalized from its original classification format. This can be done because, despite presenting a categorical format, the order of the numbers does have meaning (the further apart two values are, the more distinct the real meaning those values represent). The next feature, “trestbps”, is a numerical value indicating the resting blood pressure in millimeters of mercury upon admission to the hospital (Janosi et al., 1989). Examining the boxplot created for this variable, we observe some outliers (Image A1 from Appendix A). To minimize the number of instances deleted, only the two highest values were defined as outliers and, therefore, deleted from the dataset. Although deleting instances is not the most accurate approach if we want a model that generalizes for all data points, a reminder is provided that the purpose of this study is to have comparable metrics between different types of models. Therefore, this does not affect how each of the models generally performs in the situation studied (in this case, a smaller dataset in the field of medicine or healthcare). In the end, it was also normalized to a value between 0 and 1. The next variable, “chol,” indicates the serum cholesterol in mg/dL (Janosi et al., 1989). Image A2 from Appendix A displays the boxplot for this feature. Using the same logic as explained in the previous variable, only the highest value was considered an outlier, and the corresponding data point was removed. The values are then normalized to a range of 0 to 1. “fbs” is a binary variable, where one indicates true and 0 indicates false, evaluating whether the patient has a fasting blood sugar level greater than 120 mg/dL (Janosi et al., 1989). With no missing values and already delimited between 0 and 1, no transformations were necessary. Continuing the description of all the present features, we have “restecg,” which refers to the resting electrocardiographic results. Unlike other previous features, despite being a categorical variable (with three possible values), their order has meaning for the measurement being made (Janosi et al., 1989): 0 signifies normal results, 1 indicates an abnormality in the ST-T wave, and 2 shows a probable or definite left ventricular hypertrophy. So, the higher the number the more serious the diagnosis is. Because of this and with no missing values, the variable was normalized with the same *MinMaxScaler* function used for all the others. “thalac” indicates the maximum heart rate achieved (Janosi et al., 1989). There are no missing values, and no outliers were assumed (Image A3 from Appendix A); therefore, the only transformation required for this feature is normalization. The attribute “exang” is another binary variable indicating whether it’s exercise-induced angina (1) or not (0) (Janosi et al., 1989). It does not have any missing values, so no transformations are necessary. The next feature is “oldpeak,”

a numerical variable indicating the ST depression induced by exercise relative to rest (Janosi et al., 1989). The distribution represented by a boxplot is shown in Image A4 from Appendix A, and the two highest values were considered outliers and removed from the dataset. The values were then normalized like before. Then we have “slope”, which is a categorical variable with three possible values describing the slope of the peak exercise ST segment: 1 for upsloping, 2 for flat and 3 for down sloping (Janosi et al., 1989). Considering again that the order of the values carries meaning the values were just normalized between 0 and 1. The next attribute does have missing values, four. “ca” indicates the number of major vessels (from 0 to 3) coloured by fluoroscopy. To fill these, a correlation is first ranking of all other features (expect the target variable) to “ca”. The top 6 of this ranking were then used in a K-Nearest-Neighbors approach. The number of Neighbors considered in this process was four. After this, the values were also normalized. The next variable “thal”, which is a categorical variable that classifies the defect by reversibility, also has missing values, two. The same method of the previous variable was applied with the normalization of the values after. The final variable is the target (“num”). It refers to the presence of heart disease in the patient. From 0 (no presence) to 4. From this feature two are created. One, “target multiclass”, retains the original and will be used for the multiclass problem of this dataset. And the other, “target binary”, keeps the 0 instances as 0 and joins all the other cases in the value 1. And with this, the description and processing regarding the dataset from experiment 1 is concluded.

Another vital detail to notice is the distribution of the target variable. In the multiclass scenario, the approximate percentages for each unique value are as follows: (0 with 54,7%; 1 with 18,1%; 2 with 12,1%; 3 with 10,7%; 4 with 4,4%). In the binary scenario: (0 with 54,7%; 1 with 45,3%). This aspect means that, especially in the multiclass problem, there is a high imbalance between the classes being evaluated.

4.3.2. Experiment 2 - MiniBooNE

For this larger dataset, there are 50 attributes and approximately 130 thousand data points. The physical meaning of each of these attributes associated with the real-life experiment is not provided by the source of the data. As a result, the processing that can be applied, as well as the interpretability of the data, is reduced.

Performing a general statistical analysis on the data reveals that there are no missing values, and all variables are numerical. Similar to the first experiment, in Appendix A, Tables A2, A3, A4, and A5 display the output of the describe function, which indicates, among other values, the minimum, maximum, and standard deviation of each variable (all values are rounded to two decimal places for better viewing). Many of the minimum values displayed (-999.00) are because of the absolute value being extremely high. Due to the absence of both missing values and physical meaning for the variables, it was decided that no further processing would be performed. The only thing to keep in mind is that two options are available regarding the data that goes into the models. It can be entirely normalized or not. Both are taken into consideration further ahead.

For this dataset, only the binary classification can be performed, and the distribution of the target variable is approximately: (0 with 71,9% and 1 with 28,1%). Due to the imbalance

between the two classes, extra precautions need to be taken when applying the models. These are discussed in a later section.

4.4. MODELLING

This section outlines the modeling approach used in both experiments, covering feature selection, Subsampling, model choice, and parameterization. Given the differences in dataset size and complexity, specific steps are required to ensure fair comparisons between classical and quantum-enhanced models. Feature selection and Subsampling are used due to the computational limitations of quantum models, which restrict the number of features and data points that can be processed efficiently.

For consistency, the same classification models are used across both experiments. The Support Vector Classifier is applied in the classical approach, while the Quantum Support Vector Classifier (QSVC) is used for quantum-enhanced learning. Experiment 1 involves both binary and multiclass classification, allowing classical models to be tested on both normalized and non-normalized data, while QSVC requires normalization. Experiment 2 is a binary classification task where the same models are applied to a larger dataset.

4.4.1. Feature Selection and Subsampling

Due to differences in dataset size and dimensionality, feature selection and Subsampling are necessary to optimize model performance and ensure feasibility, particularly for quantum-enhanced models. Reducing the number of features helps mitigate computational complexity while keeping the most relevant information for classification. Subsampling is also applied to manage dataset size. For comparison purposes, these transformations are applied even when they are not strictly necessary. A direct connection can be established between scores through the same parameter values across different models, thanks to this technique. Experiments 1 and 2 differ in the way this process is applied, which is explained in the following sections.

4.4.1.1. Experiment 1 - Heart Disease

For the first experiment, there are four different situations: binary and multiclass classification for both the classical model and the quantum-enhanced model. Ideally, the scores obtained would all be related to using the entire dataset (both rows and features). However, due to computational limitations, the hybrid models cannot be run using the whole dataset. It was decided to run the models for different amounts of features. With a total of 13, the chosen values were 3, 5, 7, 9, 11, and 13, as applicable. This way, there are comparison scores between the models for all these different instances, also enabling a more in-depth analysis of how the models perform in varying circumstances.

To determine which attributes are used, the correlation values (Pandas.DataFrame.Corr — Pandas 2.2.3 Documentation) of all variables with both the target variables, binary and multiclass, were obtained (Tables B1 and B2 in Appendix B). They were always chosen in descending order of the correlation value so that the highest correlations were included first.

As the features are all normalized from the processing steps, the data for the hybrid models is the same, and so are the correlation values.

4.4.1.2. Experiment 2 - MiniBooNE

In the second experiment, there is only binary classification. With this dataset having both more features and significantly more data points, both dimensionality reduction techniques need to be employed, given the computational limitations of running the quantum models: feature selection and Subsampling. As in the first experiment, all the combinations chosen for these parameters were equal for both the classical and the quantum-enhanced models. The source of this data is more complex than the first dataset, and the meaning of each feature is unknown, so interpretability is lower in this experiment.

The correlation values were obtained by applying the same method. The values chosen for the number of attributes were: 3, 5, 10, 15, 20, 25, 30, 40 and 50. And for the dataset sizes: 200, 350, 500, 750, and 1000. For the classical model, an instance is also run using all the data (with varying number of attributes). The data used for the classical model is not normalized (explained further in a later section). Despite the quantum-enhanced model needing to have the entire dataset normalized, because that transformation is applied to the entire dataset, the correlation values and order remain the same. The ten highest correlations are displayed in Appendix B (Table B3).

4.4.2. Models and Parameters

This section outlines the models and their corresponding parameters used in both experiments. Hyperparameter tuning is conducted to optimize them, with parameters such as kernel type and tolerance adjusted for SVC, while QSVC relies on quantum kernel configurations and feature maps. Since Experiment 1 includes both binary and multiclass classification, parameter optimization considers both problem types. In Experiment 2, which is a strictly binary classification problem, the same tuning approach is applied, with adaptations to handle the larger dataset. The same models and parameter selection strategies are maintained across both experiments, allowing for direct performance comparisons.

For reference, all code created was executed in the same computer, which has the following specifications that have a direct impact on some of the results: *Processor:* "Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz 1.80 GHz"; *Installed RAM:* "8.00 GB (7.88 GB usable)".

For all the following models, the function (*Train_test_split*) is used to split the data in the required format. The training and test proportions of each dataset are defined as 75% and 25%, respectively. A random state is also included to ensure that the split is randomized, and the stratify option is utilized to maintain approximately equal proportions of the different target variable classes in both the training and testing datasets. This way, the results are more accurate in relation to the model's capabilities and less influenced by external statistical effects.

4.4.2.1. Experiment 1 - Heart Disease

Classical Section:

In the binary instance for Experiment 1, the proportions of the classes resulting from the data splitting are *0: 0.547 and 1: 0.453*. And for the multiclass problem: *0: 0.547; 1: 0.179; 2: 0.121; 3: 0.107; 4: 0.04.* The classical model (for both classical and multiclass problems) used is SVC. This model finds the optimal hyperplanes (with the same number of dimensions as the number of features being used) that maximize the distances between data points of different classes. For some of the parameters available considered the most important for the results obtained, a grid search is performed. The values selected for the binary problem were:

- "kernel": ['linear,' 'poly,' 'rbf,' 'sigmoid']
- "C": [0.5, 1.0, 1.5, 2.0, 2.5]
- "gamma": ['scale,' 'auto,' 0.1, 0.01]
- "degree": [2, 3, 4]
- "tol": [0.01, 0.001]
- "max_iter": [1000, 1500]

This results in a maximum of 960 combinations among all the selected parameters. In the grid search of the multiclass problem, only the parameter "C" varies: "C": [1.0, 1.5, 2.0, 2.5, 3.0].

By providing a brief description of these parameters, their values and differences become better understood. The 'kernel' (Plot Classification Boundaries with Different SVM Kernels) defines the type of hyperplanes, with 'linear' being suitable for linearly separable classes and 'poly' allowing for malleable borders. *"C"* manages the balance between minimizing classification errors and maximizing the margins between classes, so a higher value prioritizes a correct classification but may overfit. *"gamma"* makes the decision boundary smoother or more complex depending on the radius of influence each datapoint has. The *"degree"* parameter only applies when the kernel is poly, also adjusting the complexity of the decision boundary. *"tol"* defines the value at which the loss function ends the training process. A smaller value yields more precise results, but it delays the convergence process. And finally, *"max_iter"* limits the number of iterations if convergence is not reached before it is reached.

For both binary and multiclass instances, all these combinations (alongside the different number of features) are separately run and evaluated, with the best model being the one that achieves the highest F1 Score (explained in the following section). For all model executions, the entire dataset is utilized.

Quantum Section:

Quantum-enhanced models impose the condition of the data being normalized. Because the data used for the classical model is already normalized, no extra processing steps are necessary. Consequently, the proportions of the target classes present in the training and testing datasets are the same as in the classical section. The model used is a Quantum Support Vector Classifier provided by the Qiskit library developed by IBM with the available documentation: (*Machine Learning Tutorials - Qiskit Machine Learning 0.8.2*), ((*QML Algorithms (Qiskit_machine_learning.Algorithms) - Qiskit ML 0.8.2*,). This item is a quantum-enhanced version of the classical model, which, instead of using classical kernels, utilizes a quantum kernel computed with quantum feature maps. This quality can be beneficial for

accurately finding more complex decision boundaries. In this problem, the time required to run the quantum-enhanced models is exponentially higher than that of the classical ones, making it impossible to use grid searches that encompass so many possible combinations. Because of this, smaller subsamples are first used to empirically determine which values for the parameters tend to yield the highest scores. This way, the grid search used is significantly smaller:

- "feature_map": [ZZFeatureMap, PauliFeatureMap]
- "reps": [1, 2, 3]
- "kernel": [FidelityQuantumKernel]

The "feature map" is the parameter responsible for encoding classical data into quantum states, which are represented in a Hilbert space. This encoding can be achieved using different functions that differ in their implementation. "reps" means the number of repetitions applied to the feature map circuit. A higher value can create more complex decision boundaries but with a higher computational cost and runtime. The "kernel" defines how similarity is calculated between data points, which ultimately determines the classification attributed to each test instance.

Despite the high computational demand of this model, for experiment one, the entire dataset is used to run the model and obtain the results for this quantum section, both for binary and multiclass problems. Independent models are run for different numbers of features utilized, and the considered best model is also the one that results in the highest F1 score among all.

4.4.2.2. Experiment 2 - MiniBooNE

Classical Section:

Experiment 2 only has a binary problem, with proportions of the classes derived from the data splitting: 0: 0.719; 1: 0.281, which is valid for both the training and test subsets. The same model of experiment 1 is used with a different grid search:

- "kernel": ['linear']
- "C": [0.25, 0.5, 1.0]
- "gamma": ['scale,' 'auto']
- "tol": [1.0, 0.1]
- "max_iter": [2900, 3300, 3700]

The characteristics of this dataset were adapted to reflect these new values, with some initial empirical adjustments made to determine the direction to follow to maximize the model's potential. First, the model was separately applied using the earlier defined number of features for the entire dataset, both with and without normalization of the data. Given that the results were considerably better for the non-normalized dataset, that is the version used to represent the classical instance in this problem. The defined best model in each case is also the one that achieves the highest F1 Score.

Due to the computational limitations of the quantum models, the data must be subsampled for these models to be effective. The classical model does not present those limitations; however, for comparison purposes, the sizes defined are also applied to the classical model.

When the entire dataset is not being used, the results oscillate substantially between consecutive runs. Thus, to ensure accuracy, the values obtained for the specified metrics are the average of 50 independent runs. Both these and the scores obtained from the usage of the entire dataset will be considered when analyzing the results.

Quantum Section:

For Experiment 2, the data used in the classical section are not normalized. Therefore, an additional processing step is taken to normalize the dataset. The grid search is very similar to the one used in Experiment 1, except that PauliFeatureMap is not included in the "feature_map" parameter.

Given the size of this dataset, more problems were expected when running the quantum models in comparison to the first experiment. Even for the sizes and number of features chosen in the independent model runs, it quickly became evident that the computer did not have the necessary capabilities to obtain the desired set of results we wanted. As noted in the results section, only the least computationally intensive models were successfully run. To resolve this issue, time was allocated to an online service that offers improved processing capacity (RunPod). Only a few more models were run successfully for a GPU with 62 GB of RAM (RTX 6000 Ada). This aspect reflects the exponential growth of resources utilized by quantum-enhanced models and the difficulties it currently poses for the field.

5. RESULTS AND DISCUSSION

The metrics used to evaluate the models are defined in this chapter, and their performances are displayed in a way that compares the associated models in each experiment. Conclusions are, therefore, easier to deduce. The full results, which serve as input for the graphics presented in the following sections, are displayed in Appendix C.

5.1. EVALUATION METRICS

The evaluation performance is measured through defined metrics, allowing us to compare it directly across different models. The first one chosen is the F1 score (*F1_score*) defined in expression 2.

$$F1 = \frac{2*\mathrm{TP}}{2*\mathrm{TP}+\mathrm{FP}+\mathrm{FN}} \tag{2}$$

where *TP* is the number of true positives, *FP* is the number of false positives, and *FN* is the number of false negatives. The second one is accuracy (*Accuracy_score*), which in this classification context is calculated through expression 3.

$$Accuracy = \frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{TN}+\mathrm{FP}+\mathrm{FN}} \tag{3}$$

with *TN* being the number of true negatives. The final one is the time (in seconds) it takes for any given model to execute. The parameter values of the grid search corresponding to the given model are also registered for each run, as some conclusions can be drawn from these results.

These are the metrics saved for each model executed and used to draw conclusions in this empirical study.

5.2. PERFORMANCE ASSESSMENT

This section evaluates the performance of the models based on the defined metrics: F1 score, accuracy, and execution time, assessing how well the classical and quantum-enhanced models perform under different configurations and dataset conditions. For each experiment, performance is evaluated across varying numbers of features and dataset sizes, ensuring a fair comparison between classical and quantum models. Special attention is given to the impact of feature selection and subsampling on the results. Execution time is analyzed to assess the computational feasibility of quantum-enhanced models in comparison to their classical counterparts. While the accuracy values are presented in the output given in Appendix C, this section focuses on the F1 score and the time taken to run, and only these two metrics are featured in the graphics built in the following sections. The F1 score is chosen over accuracy because it considers the imbalances between the different target values. For specific individual situations, the parameters identified as the optimal combination for that particular model have a significant impact on the values of the evaluation metrics defined. These and other instances are analyzed in the section ahead created for that purpose.

Another essential detail to notice is that the graphics in the following sections have lines connecting each of the data points. This facet, of course, is not accurate because the actual points are the only items we have data on, not what is in between them. These lines are added for visualization purposes only, making it easier to spot trends and follow each individual line as needed.

5.2.1. Experiment 1 - Heart Disease

The results in Appendix C, which serve as the source for the following graphics regarding Experiment 1, are printed in images C1 to C4. As stated before, the evaluation metrics that are represented with more expressiveness are the F1 score and the time it takes for each model to run.

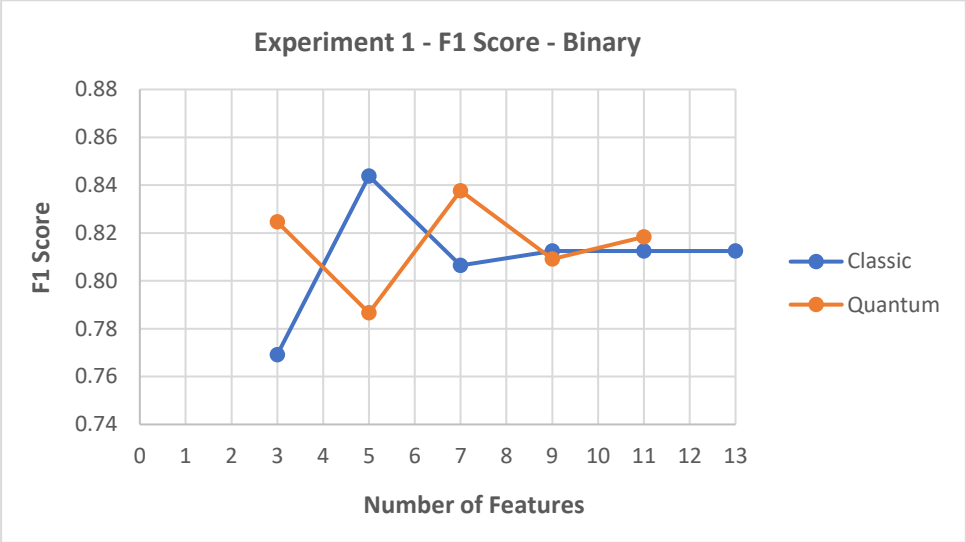

Figure 5.1 - Evolution of the F1 Scores with the number of features for both classical and quantum instances in the binary classification problem (Experiment 1)

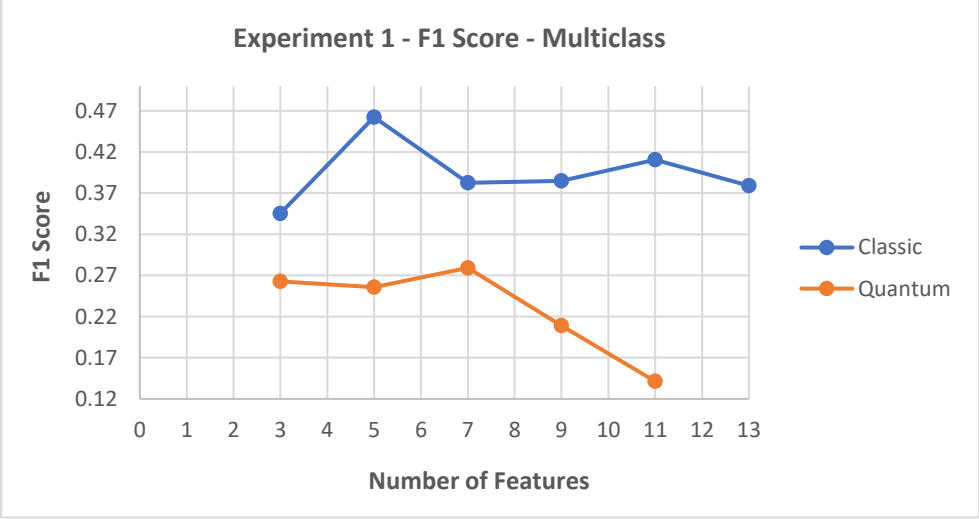

Figure 5.2 - Evolution of the F1 Scores with the number of features for both classical and quantum instances in the multiclass classification problem (Experiment 1)

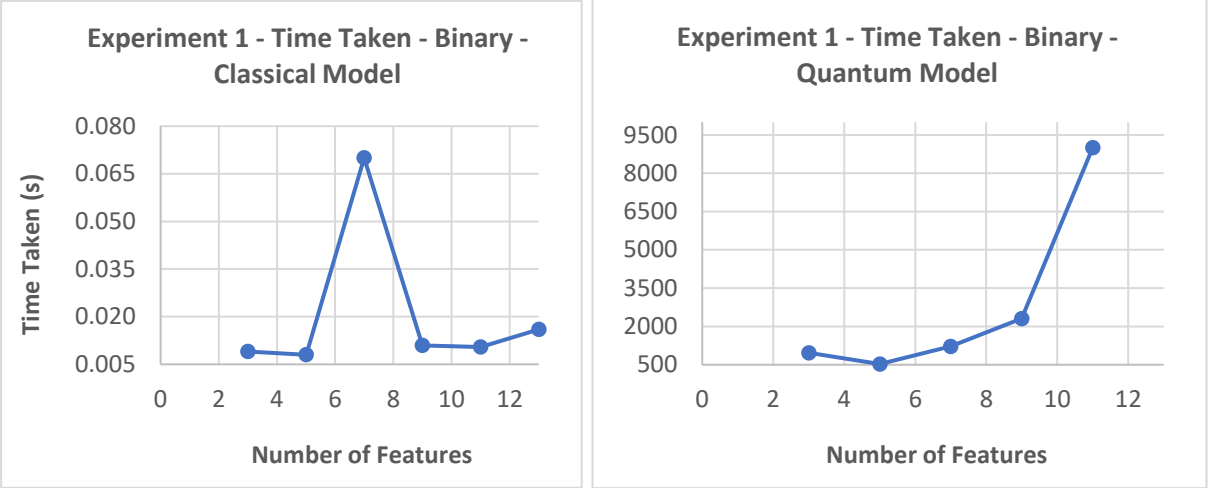

Figure 5.3 - Evolution of the time taken to run the models with the number of features for both classical and quantum instances in the binary classification problem (Experiment 1)

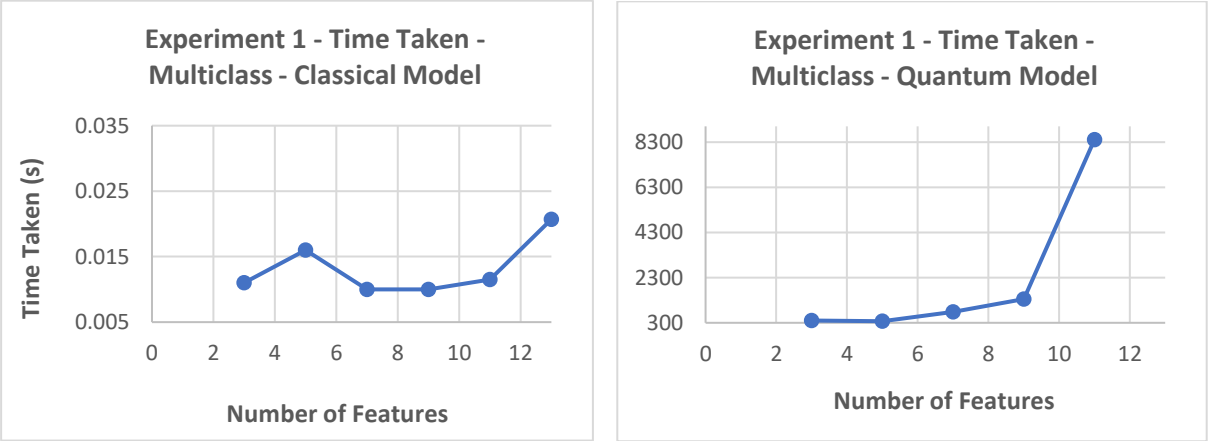

Figure 5.4 - Evolution of the time taken to run the models with the number of features for both classical and quantum instances in the multiclass classification problem (Experiment 1)

5.2.2. Experiment 2 - MiniBooNE

The primary source of comparison between the different models is the use of the same conditions on both sides, ensuring the same number of features and data points. Despite that, to establish a benchmark and fully explore the potential of the classical version of the model, results using the entire dataset in this domain are also obtained. These are displayed in the outputs shown in Images C5 and C6 of Appendix C.

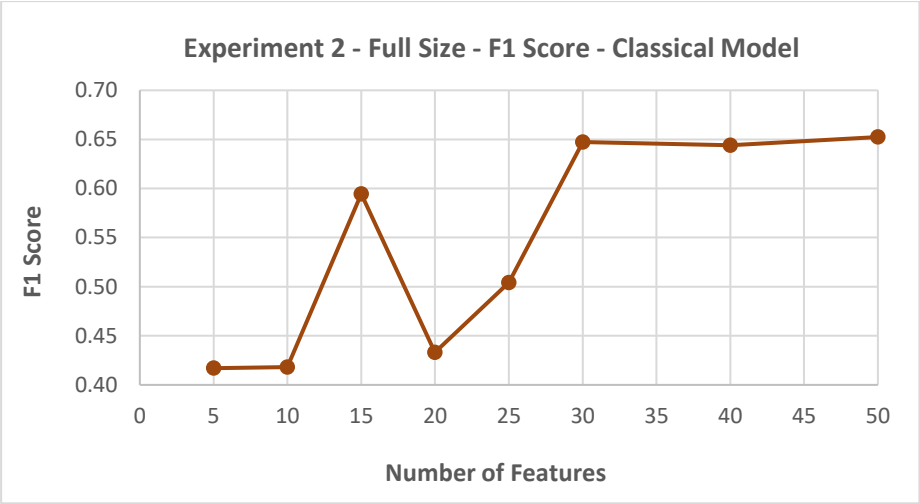

Figure 5.5 - Evolution of the F1 Score with the number of features for the classical model using the entire length of the dataset (Experiment 2)

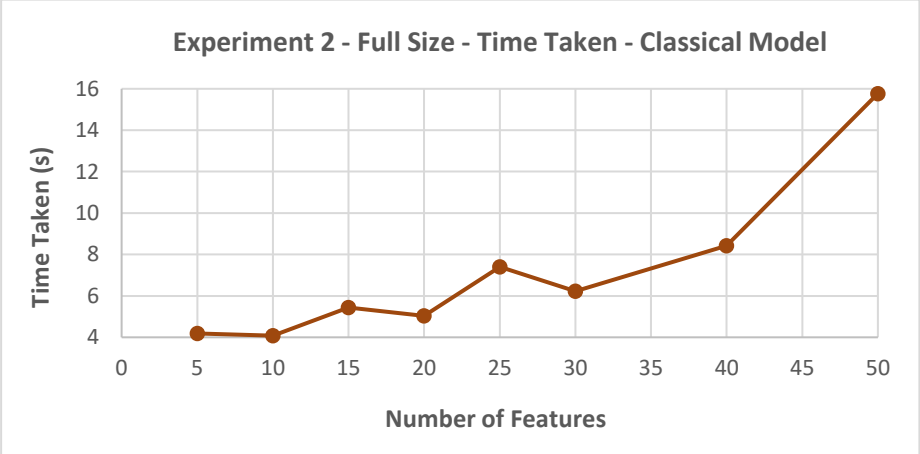

Figure 5.6 - Evolution of the time taken to run the classical model with the number of features for the classical model using the entire length of the dataset (Experiment 2)

Regarding the direct comparisons between the models, the results in Appendix C, which serve as the source for the following graphics, are printed in images C7 to C18. As stated before, the evaluation metrics represented with more expressiveness are the F1 score and the time it takes for each model to run.

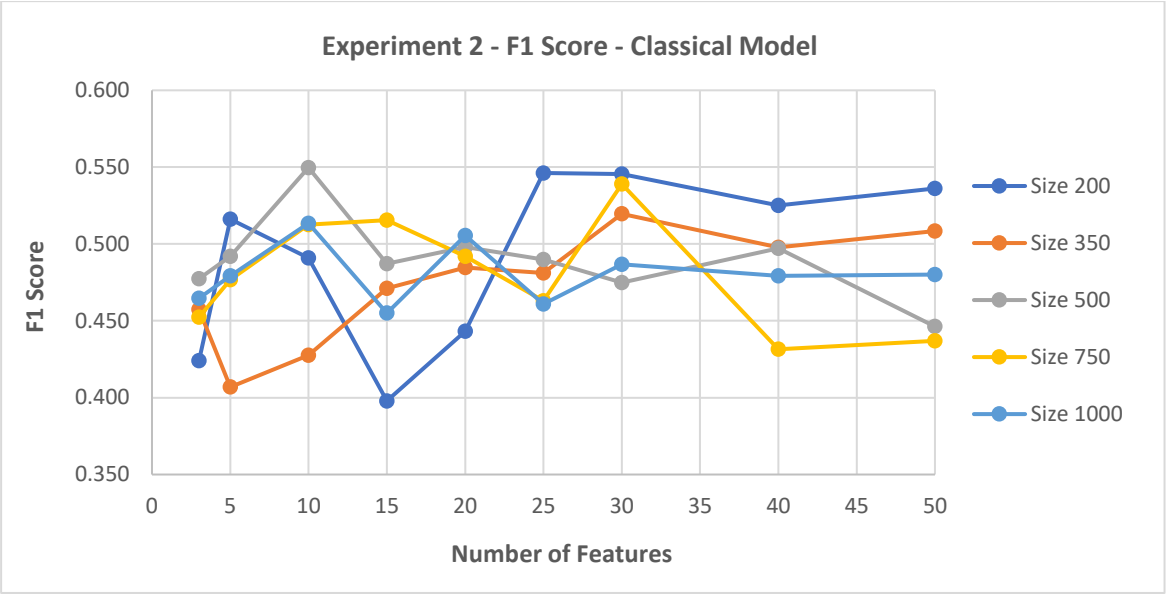

Figure 5.7 - Evolution of the F1 Scores with the number of features for the classical model and varying dataset sizes (Experiment 2)

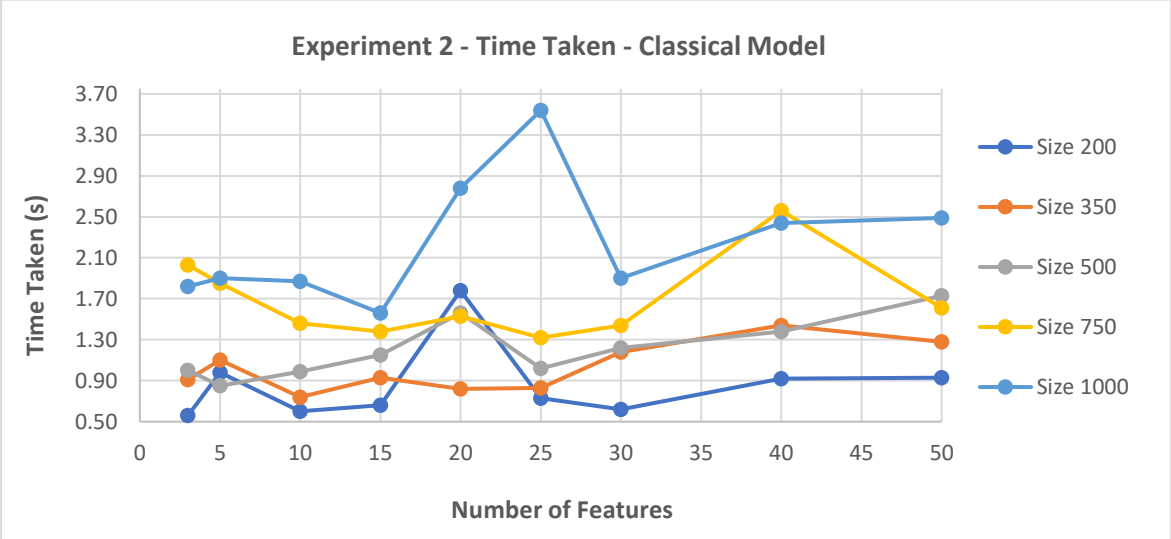

Figure 5.8 - Evolution of the time taken to run each of the classical models with the number of features and for varying dataset sizes (Experiment 2)

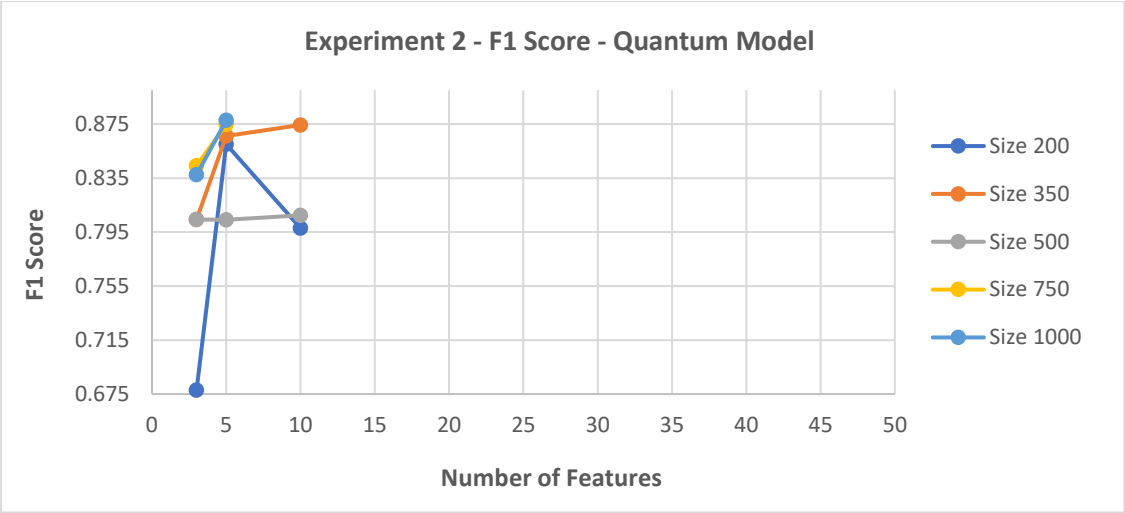

Figure 5.9 - Evolution of the F1 Score with the number of features for the quantum model and varying dataset sizes (Experiment 2)

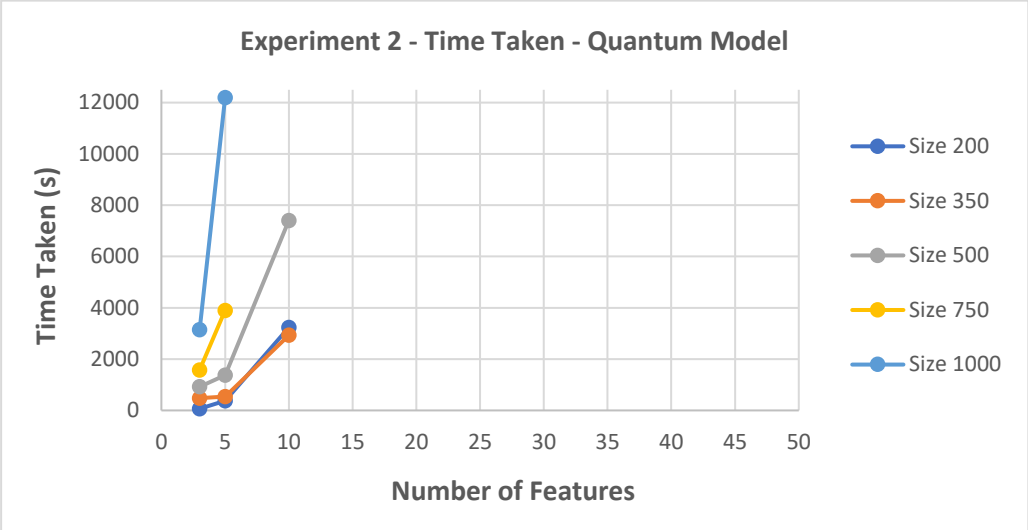

Figure 5.10 - Evolution of the time taken to run the quantum model with the number of features and for varying dataset sizes (Experiment 2)

5.3. Results Exploration

In this section, some data points of the given results in the previous section are analyzed more closely in situations where a deviation from the "norm" is apparent. Because only the F1 score and the time taken to run each model are directly represented in these graphics, other information, such as the values for the parameters obtained from the grid searches that match the best models, is not directly considered. However, as shown in Appendix C, it can be used to justify some of the visibly noticeable discrepancies.

5.3.1. Experiment 1 - Heart Disease

For Experiment 1, the results are obtained based on only one execution of any given model, unlike Experiment 2, where, in some situations for the classical part, the results displayed come from an average of running the same model with the same parameters several times. Because of this, statistically, there is a higher probability of falling outside the norm values. For different runs, the results will change slightly, with some showing a bigger difference to the average result. This factor can help explain some of the variability seen in the graphics for experiment 1.

The most extreme data point is shown in Figure 5.4 regarding the classical model. For the value of 7 features, there is a significant increase in the time taken value, returning to the norm in the following data point. Examining the corresponding parameter values, the instance with seven features is the only case where the parameter “kernel” is set to “rbf,” which may help explain the significant difference in this specific data point.

5.3.2. Experiment 2 - MiniBooNE

There is a higher instability (despite the method of averages being used for the classical models in Experiment 2) when the number of features utilized is just a fraction of the total available in the dataset. With a higher number of features, these values stabilize, indicating that the initial volatility is statistically normal.

As expected, when the model has quantum capabilities, it requires significantly more computational resources to be executed, approximately exponentially, as shown in Figure 5.4, where the curve for the time taken to run the quantum-enhanced model in the experiment exhibits this kind of evolution. The missing data points result from memory errors and kernel shutdowns, which occur due to the excessive computational resources required for that specific combination of variables. While this restricts the conclusions that can be taken from these results, it doesn’t annul some of the trends that can be seen and are discussed in the next section.

5.4. Trends and Analysis

In this section, the results obtained are analyzed in greater detail to identify patterns, trends, and possible explanations for the observed behaviors. The analysis examines the relationships between the number of features, computational time, and classification performance, with a focus on how the F1 Score compares across different models under similar conditions.

5.4.1. Experiment 1 - Heart Disease

For experiment 1, we have binary and multiclass problems. For the binary instance, the values around where the F1 Score stabilizes are similar between the classical and quantum-enhanced models (Figure 5.1). Minor differences cannot be considered at face value due to the lack of more data points and knowledge about the concrete variability of the results between different runs on the same parameters (akin to a potential error bar for each point). However,

one detail to note from the graph is that the quantum model appears to reach the final range of values sooner, suggesting a more efficient learning process compared to the classical model. This fact will be noted in the graphs to see if a similar trend occurs.

Regarding the time taken to run the models (Figure 5.3), the classical version is several orders of magnitude faster, with its evolution also being much less steep than that of the quantum one, which displays an exponential curve with the increase in features used. More specifically, considering a value of 0.02 seconds for the classical model and 9000 seconds for the quantum one, the ratio obtained is approximately 450,000 times greater. With this, we can conclude that the classical model is preferential over the quantum one mainly due to the reduced computational and time resources necessary to obtain the same results.

Now, for the multiclass problem (Figure 5.2), there is a clear separation between the performances of the models, with the classical one showing better scores. And for the time taken, as in the classical situation, it's orders of magnitude faster, and the evolution is much slower. The quantum-enhanced model again presents exponential growth in computational resources with the increase in features considered. In experiment one, the classical model is the clear winner out of the two. This time, we repeat the same exercise as in the binary problem to get an idea of the difference in running times. Considering 0.025 seconds for the classical one and 8,300 for the quantum one, we obtain a ratio of 332,000. Therefore, considering this is a rough estimate of the same order of magnitude as the classical model.

5.4.2. Experiment 2 - MiniBooNE

Using the classical model with the entire dataset, the F1 score appears to stabilize around the mark of 30 features used, at 0.65 (Figure 5.5). In contrast, for smaller data sizes, it never reaches these values, instead remaining approximately between 0.45 and 0.55 (Figure 5.7). This element indicates that the best performance of the classical model can only be obtained with a higher amount of data, confirming the general idea that for these models, the more data put into it, the better. For the time taken, when using the entire dataset, the evolution exhibits a clear upward trend, with the last data point indicating 16 seconds when using all features (Figure 5.6). For the smaller sizes, the evolution trends are not prominent at all (Figure 5.8) due to the statistical variabilities associated with running on smaller sizes, despite taking action to consider the average of fifty separate runs on the same parameters for each instance.

Now, examining the quantum model, only two or three data points were obtained for these sizes due to the substantial computational resources required to run these models on the larger dataset (Figure 5.9). This discourse limits the conclusions that can be drawn but does not annul the apparent difference in the F1 scores compared to the classical model, with values over 0.85 being reached despite the limited number of features used. The lack of data points makes it impossible to see any trends in the evolution of the scores. Nonetheless, by the first ones, and given that the scores are already very high, one can confirm the detail pointed out in the first experiment: that the quantum-enhanced model seems to reach the final range of scores sooner than the classical one, being more efficient. Regarding the time taken to run the models, as expected, it's once again orders of magnitude higher for the quantum model. However, this time, using the same calculation as in Experiment One,

considering the values 2.50 seconds for the classical model and 12193 seconds for the quantum-enhanced one, the ratio obtained is approximately 4877. Compared with the ratios obtained in Experiment 1, we have two orders of magnitude less here. This element indicates that for the larger dataset with more complex relationships, the relative difference between the two is decreasing, suggesting that for these datasets, the benefits of using quantum-enhanced models seem greater than for shorter, less complex datasets.

6. CONCLUSIONS AND FUTURE WORKS

In this thesis, we explored the application of quantum-enhanced ML models and compared their performance with that of classical approaches in classification tasks. The study involved two separate experiments in which the corresponding F1 scores and computational efficiency were evaluated across different conditions. Empirical results were presented and analyzed, also taking into account their limitations. Despite their limitations, the results provide insights into when quantum models outperform classical approaches and their associated computational costs.

Returning to the initial proposed RQ:

RQ1: What kinds of ML problems benefit from the use of quantum computing?

The experiments suggest that quantum-enhanced models are more likely to be beneficial in tasks where the relationships between the different variables are more complex and where traditional models struggle to achieve higher results without significantly increasing the dataset size. Quantum models demonstrate the ability to reach optimal performance with fewer features, indicating a more efficient learning process.

RQ2: Quantify the benefits relative to classical computation for these instances.

The findings indicate that quantum models can achieve higher evaluation scores for these more complex problems, particularly for higher-dimensional datasets. However, this comes at a significant computational cost, with execution times being orders of magnitude higher than classical counterparts. As dataset complexity increases, the relative computational gap seems to narrow based on the two experiments performed.

Due to hardware constraints, only a limited number of features and data points can be processed in the quantum-enhanced models, resulting in incomplete results for specific experiments. Another limitation is that the experiments were conducted using particular machine learning (ML) algorithms and datasets. The findings may not apply to other classification problems or machine learning domains. Finally, the computational resources required for quantum models remain the biggest challenge, limiting their practical implementation in real scenarios.

Future studies could investigate a broader range of machine learning (ML) problems beyond classification, such as clustering or generative modeling, to determine the impact of quantum computing across various areas. Expanding the experimental setup to include different quantum architectures and larger datasets would also help provide a more comprehensive evaluation of the real-world applicability of these models. Future research should also focus on overcoming the computational limitations of quantum-enhanced models.

BIBLIOGRAPHICAL REFERENCES

Accuracy_score. . Scikit-Learn. Retrieved 9 March 2025, from https://scikit-learn/stable/modules/generated/sklearn.metrics.accuracy_score.html

Alharbi, A., Seh, A. H., Alosaimi, W., Alyami, H., Agrawal, A., & Kumar, R. (2021). Analyzing the Impact of Cyber Security Related Attributes for Intrusion Detection Systems. *Sustainability (Switzerland)*, 1–19.

Bauer, B., Bravyi, S., Motta, M., & Chan, G. K.-L. (2020). Quantum algorithms for quantum chemistry and quantum materials science. *Chemical Reviews*, *120*(22), 12685–12717. https://doi.org/10.1021/acs.chemrev.9b00829

Biamonte, J., Wittek, P., Pancotti, N., Rebentrost, P., Wiebe, N., & Lloyd, S. (2017). Quantum Machine Learning. *Nature*, *549*(7671), 195–202. https://doi.org/10.1038/nature23474

Cao, Y., Romero, J., & Aspuru-Guzik, A. (2018). Potential of quantum computing for drug discovery. *IBM Journal of Research and Development*, *62*(6), 6:1-6:20. https://doi.org/10.1147/JRD.2018.2888987

Ceschini, A., Rosato, A., & Panella, M. (2022). Hybrid Quantum-Classical Recurrent Neural Networks for Time Series Prediction. *2022 International Joint Conference on Neural Networks (IJCNN)*, 1–8. https://doi.org/10.1109/IJCNN55064.2022.9892441

Chen, S. Y.-C., Wei, T.-C., Zhang, C., Yu, H., & Yoo, S. (2020). *Quantum Convolutional Neural Networks for High Energy Physics Data Analysis* (arXiv:2012.12177). arXiv. http://arxiv.org/abs/2012.12177

Cirq. . Google Quantum AI. Retrieved 17 April 2025, from https://quantumai.google/cirq

Exact and noisy simulation with Qiskit Aer primitives. . IBM Quantum Documentation. Retrieved 17 April 2025, from https://docs.quantum.ibm.com/guides/docs.quantum.ibm.com/guides/simulate-with-qiskit-aer

F1_score. . Scikit-Learn. Retrieved 9 March 2025, from https://scikit-learn/stable/modules/generated/sklearn.metrics.f1_score.html

Fermilab | Science at Fermilab | Experiments & Projects | Intensity Frontier | MiniBooNE. . Retrieved 22 February 2025, from https://www.fnal.gov/pub/science/experiments/intensity/miniboone.html

Ghosh, A., Saha, S., Bhagat, V., & Ganie, A. H. (2024). Quantum vs. Classical: A Rigorous Comparative Study on Neural Networks for Advanced Satellite Image Classification. *2024 International Conference on Trends in Quantum Computing and Emerging Business Technologies*, 1–5. https://doi.org/10.1109/TQCEBT59414.2024.10545133

Harrow, A. W., & Montanaro, A. (2017). Quantum Computational Supremacy. *Nature*, *549*(7671), 203–209. https://doi.org/10.1038/nature23458

Janosi, A., Steinbrunn, W., Pfisterer, M., & Detrano, R. (1989). Heart Disease [Dataset]. UCI Machine Learning Repository. https://doi.org/10.24432/C52P4X

IBM Quantum. . IBM Quantum. Retrieved 17 April 2025, from https://quantum.ibm.com/

Jadhav, A., Rasool, A., & Gyanchandani, M. (2023). Quantum Machine Learning: Scope for real-world problems. *Procedia Computer Science*, *218*, 2612–2625. https://doi.org/10.1016/j.procs.2023.01.235

Jordan, M. I., & Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. *Science*, *349*(6245), 255–260. https://doi.org/10.1126/science.aaa8415

Jurcevic, P., Javadi-Abhari, A., Bishop, L. S., Lauer, I., Bogorin, D. F., Brink, M., Capelluto, L., Günlük, O., Itoko, T., Kanazawa, N., Kandala, A., Keefe, G. A., Krsulich, K., Landers, W., Lewandowski, E. P., McClure, D. T., Nannicini, G., Narasgond, A., Nayfeh, H. M., … Gambetta, J. M. (2021). Demonstration of quantum volume 64 on a superconducting quantum computing system. *Quantum Science and Technology*, *6*(2), 025020. https://doi.org/10.1088/2058-9565/abe519

Kashif, M., & Al-Kuwari, S. (2022). Demonstrating Quantum Advantage in Hybrid Quantum Neural Networks for Model Capacity. *2022 IEEE International Conference on Rebooting Computing (ICRC)*, 36–44. https://doi.org/10.1109/ICRC57508.2022.00011

Kobayashi, M., Nakaji, K., & Yamamoto, N. (2022). Overfitting in quantum machine learning and entangling dropout. *Quantum Machine Intelligence*, *4*(2), 30. https://doi.org/10.1007/s42484-022-00087-9

Kösoglu-Kind, B., Loredo, R., Grossi, M., Bernecker, C., Burks, J. M., & Buchkremer, R. (2023). A biological sequence comparison algorithm using quantum computers. *Scientific Reports*, *13*(1), 14552. https://doi.org/10.1038/s41598-023-41086-5

Liberati, A., Altman, D. G., Tetzlaff, J., Mulrow, C., Gotzsche, P. C., Ioannidis, J. P. A., Clarke, M., Devereaux, P. J., Kleijnen, J., & Moher, D. (2009). The PRISMA statement for reporting systematic reviews and meta-analyses of studies that evaluate healthcare interventions: Explanation and elaboration. *BMJ*, *339*(jul21 1), b2700–b2700. https://doi.org/10.1136/bmj.b2700

Machine Learning Tutorials—Qiskit Machine Learning 0.8.2. . Retrieved 22 March 2025, from https://qiskit-community.github.io/qiskit-machine-learning/tutorials/index.html

Maldonado-Romo, A., Montiel-Pérez, J. Y., Onofre, V., Maldonado-Romo, J., & Sossa-Azuela, J. H. (2024). Quantum K-Nearest Neighbors: Utilizing QRAM and SWAP-Test Techniques for Enhanced Performance. *Mathematics*, *12*(12), Article 12. https://doi.org/10.3390/math12121872

Matic, A., Monnet, M., Lorenz, J. M., Schachtner, B., & Messerer, T. (2022). Quantum-classical convolutional neural networks in radiological image classification. *2022 IEEE International Conference on Quantum Computing and Engineering (QCE)*, 56–66. https://doi.org/10.1109/QCE53715.2022.00024

MinMaxScaler. . Scikit-Learn. Retrieved 2 March 2025, from https://scikit-learn/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html

Moher, D., Liberati, A., Tetzlaff, J., & Altman, D. (2009). Moher D, Liberati A, Tetzlaff J, Altman DG, Group PPreferred reporting items for systematic reviews and meta-analyses: The PRISMA statement. PLoS Med 6: e1000097. *Open Medicine : A Peer-Reviewed, Independent, Open-Access Journal*, *3*, e123-30. https://doi.org/10.1016/j.jclinepi.2009.06.005

Nielsen, M. A., & Chuang, I. L. (2010, December 9). *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Higher Education from Cambridge University Press; Cambridge University Press. https://doi.org/10.1017/CBO9780511976667

Orus, R., Mugel, S., & Lizaso, E. (2019). Quantum computing for finance: Overview and prospects. *Reviews in Physics*, *4*, 100028. https://doi.org/10.1016/j.revip.2019.100028

Özpolat, Z., & Karabatak, M. (2023). Exploring the Potential of Quantum-Based Machine Learning: A Comparative Study of QSVM and Classical Machine Learning Algorithms. *2023 11th International Symposium on Digital Forensics and Security (ISDFS)*, 1–5. https://doi.org/10.1109/ISDFS58141.2023.10131821

pandas.DataFrame.corr—Pandas 2.2.3 documentation. . Retrieved 8 March 2025, from https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.corr.html

Peruzzo, A., McClean, J., Shadbolt, P., Yung, M.-H., Zhou, X.-Q., Love, P. J., Aspuru-Guzik, A., & O'Brien, J. L. (2014). A variational eigenvalue solver on a quantum processor. *Nature Communications*, *5*(1), 4213. https://doi.org/10.1038/ncomms5213

Plot classification boundaries with different SVM Kernels. . Scikit-Learn. Retrieved 22 March 2025, from https://scikit-learn/stable/auto_examples/svm/plot_svm_kernels.html

Poggiali, A., Berti, A., Bernasconi, A., Del Corso, G. M., & Guidotti, R. (2024). Quantum Clustering with k-Means: A Hybrid Approach. *Theoretical Computer Science*, *992*, 114466. https://doi.org/10.1016/j.tcs.2024.114466

Preskill, J. (2018). Quantum Computing in the NISQ era and beyond. *Quantum*, *2*, 79. https://doi.org/10.22331/q-2018-08-06-79

Qiskit | IBM Quantum Computing. . Retrieved 3 July 2024, from https://www.ibm.com/quantum/qiskit

Quantum machine learning algorithms (qiskit_machine_learning.algorithms)—Qiskit Machine Learning 0.8.2. . Retrieved 22 March 2025, from https://qiskit-community.github.io/qiskit-machine-learning/apidocs/qiskit_machine_learning.algorithms.html

Rebentrost, P., Gupt, B., & Bromley, T. R. (2018). Quantum computational finance: Monte Carlo pricing of financial derivatives. *Physical Review A*, *98*(2), 022321. https://doi.org/10.1103/PhysRevA.98.022321

Reka, S. S., Karthikeyan, H. L., Shakil, A. J., Venugopal, P., & Muniraj, M. (2024). Exploring Quantum Machine Learning for Enhanced Skin Lesion Classification: A Comparative Study of Implementation Methods. *IEEE Access*, *12*, 104568–104584. IEEE Access. https://doi.org/10.1109/ACCESS.2024.3434681

Roe, B. (2005a). *MiniBooNE particle identification* [Dataset]. UCI Machine Learning Repository. https://doi.org/10.24432/C5QC87

Roe, B. (2005b). *MiniBooNE particle identification* [Dataset]. UCI Machine Learning Repository. https://doi.org/10.24432/C5QC87

RunPod. . *RunPod—The Cloud Built for AI*. Retrieved 23 March 2025, from https://www.runpod.io/

Senokosov, A., Sedykh, A., Sagingalieva, A., Kyriacou, B., & Melnikov, A. (2024). Quantum machine learning for image classification. *Machine Learning: Science and Technology*, *5*(1), 015040. https://doi.org/10.1088/2632-2153/ad2aef

sklearn.neighbors.NearestNeighbors—Scikit-learn 0.15-git documentation. (-b). Retrieved 17 April 2025, from https://scikit-learn.org/0.15/modules/generated/sklearn.neighbors.NearestNeighbors.html

sklearn.neighbors.NearestNeighbors—Scikit-learn 0.15-git documentation. (-a). Retrieved 17 April 2025, from https://scikit-learn.org/0.15/modules/generated/sklearn.neighbors.NearestNeighbors.html

Tantawi, B., Ahmed, H. K., Magdy, M., Adel, M., & Sayed, G. I. (2023). Exploring the Power of Quantum Convolutional Neural Networks for Brain MRI Image Classification. *2023 Intelligent Methods, Systems, and Applications (IMSA)*, 549–584. https://doi.org/10.1109/IMSA58542.2023.10217624

Train_test_split. . Scikit-Learn. Retrieved 22 March 2025, from https://scikit-learn/stable/modules/generated/sklearn.model_selection.train_test_split.html

Upama, P. B., Sridevi, P., Rabbani, M., Syam, M., Bashar, A. H. M., Mondal, Md. R. H., Khan, R. A., & Ahamed, S. I. (2024). A Comparative Study of Classical and Quantum Algorithms for Heart Disease Prediction Using Patients' Vital Signs. *2024 IEEE 48th Annual Computers, Software, and Applications Conference (COMPSAC)*, 1835–1840. https://doi.org/10.1109/COMPSAC61105.2024.00290

Wiedmann, M., Hölle, M., Periyasamy, M., Meyer, N., Ufrecht, C., Scherer, D. D., Plinge, A., & Mutschler, C. (2023). An Empirical Comparison of Optimizers for Quantum Machine Learning with SPSA-based Gradients. *2023 IEEE International Conference on Quantum Computing and Engineering (QCE)*, 450–456. https://doi.org/10.1109/QCE57702.2023.00058

Yadav, A., S, S., R, P. K., & S, B. (2023). Hybrid Quantum-Classical Convolutional Neural Network for Allen Telescope SETI image Classification. *2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, 1–6. https://doi.org/10.1109/ICCCNT56998.2023.10308102

Ying, M. (2010). Quantum computation, quantum theory and AI. *Artificial Intelligence*, *174*(2), 162–176. https://doi.org/10.1016/j.artint.2009.11.009

Zeguendry, A., Jarir, Z., & Quafafou, M. (2022). Quantum Machine Learning: Practical Cases. *2022 International Conference on INnovations in Intelligent SysTems and Applications (INISTA)*, 1–6. https://doi.org/10.1109/INISTA55318.2022.9894249

APPENDIX A - DATA PROCESSING

Table A1: Output of the "describe" function applied to the dataset of experiment 1

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	num
count	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	299.000000	301.000000	303.000000
mean	54.438944	0.679868	3.158416	131.689769	246.693069	0.148515	0.990099	149.607261	0.326733	1.039604	1.600660	0.672241	4.734219	0.937294
std	9.038662	0.467299	0.960126	17.599748	51.776918	0.356198	0.994971	22.875003	0.469794	1.161075	0.616226	0.937438	1.939706	1.228536
min	29.000000	0.000000	1.000000	94.000000	126.000000	0.000000	0.000000	71.000000	0.000000	0.000000	1.000000	0.000000	3.000000	0.000000
25%	48.000000	0.000000	3.000000	120.000000	211.000000	0.000000	0.000000	133.500000	0.000000	0.000000	1.000000	0.000000	3.000000	0.000000
50%	56.000000	1.000000	3.000000	130.000000	241.000000	0.000000	1.000000	153.000000	0.000000	0.800000	2.000000	0.000000	3.000000	0.000000
75%	61.000000	1.000000	4.000000	140.000000	275.000000	0.000000	2.000000	166.000000	1.000000	1.600000	2.000000	1.000000	7.000000	2.000000
max	77.000000	1.000000	4.000000	200.000000	564.000000	1.000000	2.000000	202.000000	1.000000	6.200000	3.000000	3.000000	7.000000	4.000000

Image A1: Boxplot display of the variable representing the resting blood pressure (mm Hg)

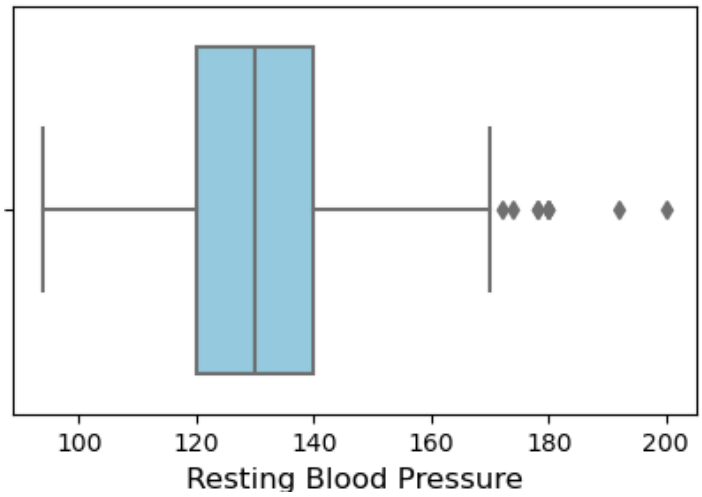

Image A2: Boxplot display of the variable representing the serum cholesterol (mg/dl)

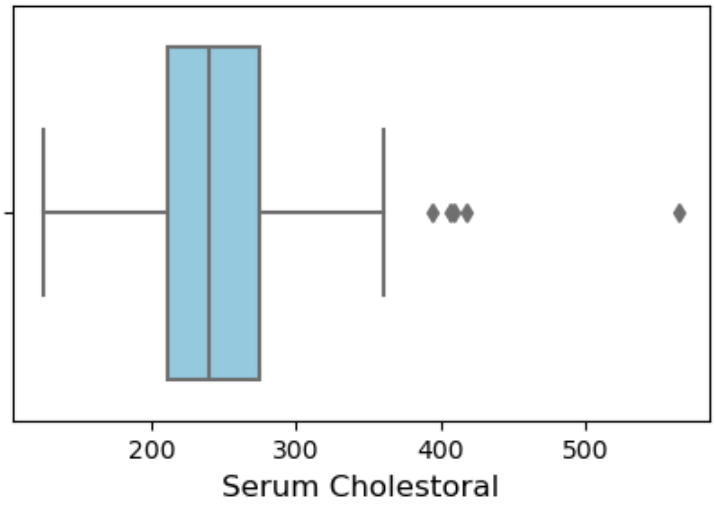

Image A3: Boxplot display of the variable representing the maximum heart rate (beats per minute)

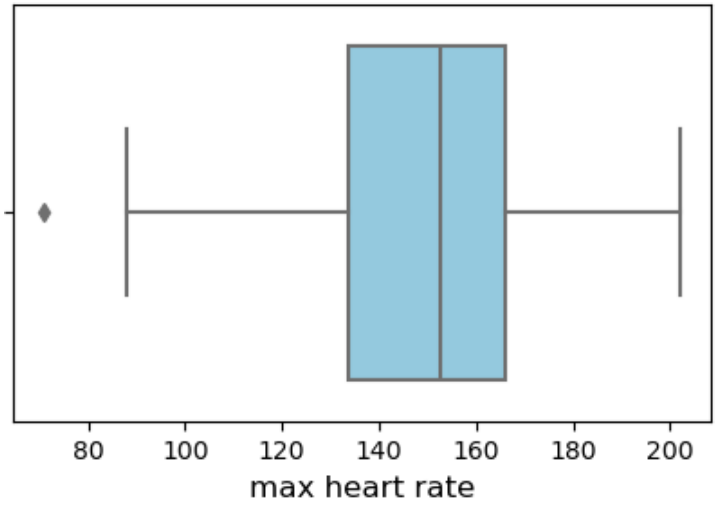

Image A4: Boxplot display of the variable representing the ST depression induced by exercise relative to rest

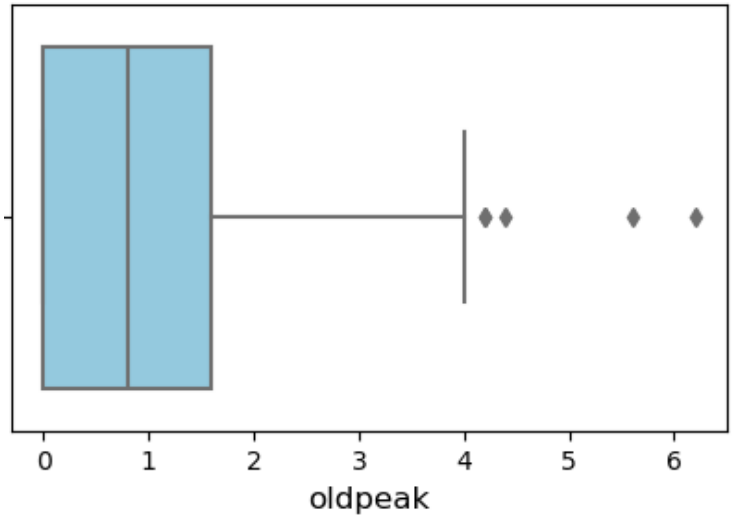

Table A2: Part 1 of the describe function output for experiment 2

	Feature_1	Feature_2	Feature_3	Feature_4	Feature_5	Feature_6	Feature_7	Feature_8	Feature_9	Feature_10	Feature_11	Feature_12	Feature_13	Feature_14
count	130064.00	130064.00	130064.00	130064.00	130064.00	130064.00	130064.00	130064.00	130064.00	130064.00	130064.00	130064.00	130064.00	130064.00
mean	1.25	-2.11	123.80	-3.31	-3.59	-3.43	-2.63	-2.77	-0.16	-3.42	0.71	161.45	-3.15	-3.29
std	60.12	59.91	196.01	59.83	59.82	59.83	59.88	59.87	60.02	59.83	60.08	134.25	59.85	59.84
min	-999.00	-999.00	-999.00	-999.00	-999.00	-999.00	-999.00	-999.00	-999.00	-999.00	-999.00	-999.00	-999.00	-999.00
25%	4.01	0.88	49.30	0.24	0.00	0.09	0.72	0.78	3.28	0.14	4.15	73.39	0.15	0.11
50%	4.79	1.38	106.14	0.27	0.00	0.14	0.92	0.83	3.43	0.17	4.29	155.76	0.63	0.24
75%	5.63	1.95	138.85	0.32	0.01	0.21	1.18	0.87	3.59	0.20	4.43	244.56	0.87	0.47
max	17.06	8.80	4747.67	0.74	0.18	0.70	6.24	0.99	7.17	0.52	9.56	537.26	1.00	1.00

Table A3: Part 2 of the describe function for experiment 2

ture_14	Feature_15	Feature_16	Feature_17	Feature_18	Feature_19	Feature_20	Feature_21	Feature_22	Feature_23	Feature_24	Feature_25	Feature_26	Feature_27
count	130064.00	130064.00	130064.00	130064.00	130064.00	130064.00	130064.00	130064.00	130064.00	130064.00	130064.00	130064.00	130064.00
mean	-1.35	933.04	-3.14	23.31	-3.33	119.91	-3.69	-3.29	80.65	0.24	-3.16	-4.19	97.40
std	59.96	652.83	59.85	68.40	59.83	44367.69	59.84	59.84	106.96	60.07	59.84	59.82	98.22
min	-999.00	-999.00	-999.00	-999.00	-999.00	-17256.70	-999.00	-999.00	-999.00	-999.00	-999.00	-999.00	-999.00
25%	1.78	562.80	0.21	9.89	0.23	0.47	-0.86	0.28	35.94	2.69	0.41	-0.94	33.05
50%	2.10	732.30	0.31	18.27	0.26	0.78	0.22	0.31	57.87	3.50	0.46	0.00	109.06
75%	2.52	1048.53	0.51	33.13	0.29	1.08	0.95	0.34	98.04	4.56	0.48	0.40	145.08
max	11.30	6907.69	3.90	1097.13	0.76	16000900.00	22.40	0.60	1428.59	78.36	0.50	4.72	1024.21

Table A4: Part 3 of the describe function for experiment 2

ture_27	Feature_28	Feature_29	Feature_30	Feature_31	Feature_32	Feature_33	Feature_34	Feature_35	Feature_36	Feature_37	Feature_38	Feature_39	Feature_40
count	130064.00	130064.00	130064.00	130064.00	130064.00	130064.00	130064.00	130064.00	130064.00	130064.00	130064.00	130064.00	130064.00
mean	-2.03	-3.42	-3.90	3.95	-2.58	-2.38	361.94	-3.52	-3.30	-3.21	-6.36	-2.32	-3.50
std	59.91	59.83	59.80	60.32	59.88	59.90	126.44	59.82	59.84	59.85	59.66	59.89	59.82
min	-999.00	-999.00	-999.00	-999.00	-999.00	-999.00	-999.00	-999.00	-999.00	-999.00	-999.00	-999.00	-999.00
25%	1.39	0.14	-0.69	5.89	0.75	0.64	305.13	0.04	0.25	-0.20	-3.49	1.09	0.07
50%	1.56	0.16	-0.25	7.21	1.03	1.04	385.88	0.07	0.29	0.20	-2.73	1.27	0.08
75%	1.72	0.20	0.09	8.82	1.29	1.60	444.89	0.10	0.34	0.78	-2.06	1.45	0.11
max	3.65	0.65	5.87	23.93	4.13	7.99	500.00	0.40	0.90	8.19	0.67	6.06	0.43

Table A5: Part 4 of the describe function for experiment 2

ature_38	Feature_39	Feature_40	Feature_41	Feature_42	Feature_43	Feature_44	Feature_45	Feature_46	Feature_47	Feature_48	Feature_49	Feature_50	Target
count 0	130064.00	130064.00	130064.00	130064.00	130064.00	130064.00	130064.00	130064.00	130064.00	130064.00	130064.00	130064.00	130064.00
mean 6	-2.32	-3.50	141.18	-25.23	-2.86	2.39	-3.59	-3.42	-3.75	-0.74	-1.99	-3.36	0.28
std 6	59.89	59.82	75.50	67.58	59.87	60.35	59.82	59.83	59.82	60.02	59.95	59.83	0.45
min 0	-999.00	-999.00	-999.00	-999.00	-999.00	-999.00	-999.00	-999.00	-999.00	-999.00	-999.00	-999.00	0.00
25% 9	1.09	0.07	122.90	-40.07	-0.03	3.41	0.00	0.12	-0.34	1.42	0.27	0.20	0.00
50% 3	1.27	0.08	145.41	-21.59	0.46	4.76	0.00	0.17	0.04	2.36	1.12	0.23	0.00
75% 6	1.45	0.11	167.09	-3.33	1.15	6.86	0.00	0.22	0.29	3.78	2.28	0.26	1.00
max 7	6.06	0.43	331.93	161.30	43.65	85.82	0.45	0.61	1.93	21.77	25.42	0.63	1.00

APPENDIX B - CORRELATION VALUES

Table B1: Correlation Values in experiment 1 for the binary target variable

Variable Name	Correlation Value
thal	0.522828
ca	0.459244
exercise induced	0.432432
oldpeak	0.424693
max heart rate	0.422717
chest pain type	0.415238
slope	0.332461
sex	0.284103
age	0.226694
Resting electrocardiographic results	0.167795
resting blood pressure	0.121318
serum cholestoral	0.115259
Fasting blood sugar	0.020278

Table B2: Correlation Values in experiment 1 for the multiclass target variable

Variable Name	Correlation Value
ca	0.515719
thal	0.506438
oldpeak	0.487178
max heart rate	0.411022
chest pain type	0.400756
exercise induced	0.392858
slope	0.357023
sex	0.239709
age	0.224700
Resting electrocardiographic results	0.184545
resting blood pressure	0.129888
serum cholestoral	0.102413
Fasting blood sugar	0.050821

Table B3: Ten highest correlation values in experiment 2

Variable Name	Correlation Value
Feature_16	0.512727
Feature_23	0.399635
Feature_27	0.193092
Feature_42	0.172224
Feature_3	0.092840
Feature_26	0.040783
Feature_21	0.039808
Feature_15	0.039556
Feature_37	0.039395
Feature_17	0.038327

APPENDIX C - RESULTS OUTPUTS

Image C1: Results for experiment 1, classical model, binary classification

```
# FOR 3 FEATURES #

# Best Parameters: {'kernel': 'sigmoid', 'C': 1.5, 'gamma': 'auto', 'degree': 2, 'tol': 0.01, 'max_iter': 1000}
# Best F1-Score: 0.7692
# Best Accuracy: 0.8000
# Time: 0.0090 seconds
```

```
# FOR 5 FEATURES #

# Best Parameters: {'kernel': 'sigmoid', 'C': 2.5, 'gamma': 'auto', 'degree': 2, 'tol': 0.01, 'max_iter': 1000}
# Best F1-Score: 0.8438
# Best Accuracy: 0.8667
# Time: 0.008 seconds
```

```
# FOR 7 FEATURES #

# Best Parameters: {'kernel': 'rbf', 'C': 2.5, 'gamma': 'scale', 'degree': 2, 'tol': 0.01, 'max_iter': 1000}
# Best F1-Score: 0.8065
# Best Accuracy: 0.8400
# Time: 0.0070 seconds
```

```
# FOR 9 FEATURES #

# Best Parameters: {'kernel': 'linear', 'C': 1.5, 'gamma': 'scale', 'degree': 2, 'tol': 0.01, 'max_iter': 1000}
# Best F1-Score: 0.8125
# Best Accuracy: 0.8400
# Time: 0.0110 seconds
```

```
# FOR 11 FEATURES #

# Best Parameters: {'kernel': 'linear', 'C': 0.5, 'gamma': 'scale', 'degree': 2, 'tol': 0.01, 'max_iter': 1000}
# Best F1-Score: 0.8125
# Best Accuracy: 0.8400
# Time: 0.0105
```

```
# FOR 13 (ALL) FEATURES #

# Best Parameters: {'kernel': 'linear', 'C': 1, 'gamma': 'scale', 'degree': 2, 'tol': 0.01, 'max_iter': 1000}
# Best F1-Score: 0.8125
# Best Accuracy: 0.8400
# Time: 0.0160 seconds
```

Image C2: Results for experiment 1, classical model, multiclass classification

```
# FOR 3 FEATURES #

# Best Parameters: {'kernel': 'sigmoid', 'C': 3.0, 'gamma': 'auto', 'degree': 2, 'tol': 0.01, 'max_iter': 1000}
# Best F1-Score (Macro): 0.3454
# Best Accuracy: 0.6000
# Time: 0.0110
```

```
# FOR 5 FEATURES #

# Best Parameters: {'kernel': 'poly', 'C': 3.0, 'gamma': 'scale', 'degree': 4, 'tol': 0.01, 'max_iter': 1000}
# Best F1-Score (Macro): 0.4624
# Best Accuracy: 0.6400
# Time: 0.0160 seconds
```

```
# FOR 7 FEATURES #

# Best Parameters: {'kernel': 'poly', 'C': 2.5, 'gamma': 'scale', 'degree': 4, 'tol': 0.01, 'max_iter': 1000}
# Best F1-Score (Macro): 0.3827
# Best Accuracy: 0.6000
# Time: 0.0170 seconds
```

```
# FOR 9 FEATURES #

# Best Parameters: {'kernel': 'poly', 'C': 1, 'gamma': 'scale', 'degree': 2, 'tol': 0.01, 'max_iter': 1000}
# Best F1-Score (Macro): 0.3849
# Best Accuracy: 0.6000
# Time: 0.0100 seconds
```

```
# FOR 11 FEATURES #

# Best Parameters: {'kernel': 'linear', 'C': 3.0, 'gamma': 'scale', 'degree': 2, 'tol': 0.01, 'max_iter': 1000}
# Best F1-Score (Macro): 0.4104
# Best Accuracy: 0.6133
# Time: 0.0115
```

```
# FOR 13 (ALL) FEATURES #

# Best Parameters: {'kernel': 'linear', 'C': 2.5, 'gamma': 'scale', 'degree': 2, 'tol': 0.01, 'max_iter': 1000}
# Best F1-Score (Macro): 0.3790
# Best Accuracy: 0.6133
# Time: 0.0207
```

Image C3: Results for experiment 1, quantum model, binary classification

```
# FOR 3 FEATURES

# Best Combination:
# Feature Map: ZZFeatureMap, Reps: 3, Kernel: FidelityQuantumKernel
# Accuracy: 0.8267, F1 Score: 0.8247, Time: 971.30 seconds
```

```
# FOR 5 FEATURES

# Best Combination:
# Feature Map: ZZFeatureMap, Reps: 1, Kernel: FidelityQuantumKernel
# Accuracy: 0.7867, F1 Score: 0.7866, Time: 528.47 seconds
```

```
# FOR 7 FEATURES

# Best Combination:
# Feature Map: ZZFeatureMap, Reps: 2, Kernel: FidelityQuantumKernel
# Accuracy: 0.8400, F1 Score: 0.8377, Time: 1214.41 seconds
```

```
# FOR 9 FEATURES

# Best Combination:
# Feature Map: ZZFeatureMap, Reps: 2, Kernel: FidelityQuantumKernel
# Accuracy: 0.8133, F1 Score: 0.8092, Time: 2315.17 seconds
```

```
# FOR 11 FEATURES

# Combination: Feature Map = ZZFeatureMap, Reps = 1, Kernel = FidelityQuantumKernel
# Accuracy: 0.8267, F1 Score: 0.8184, Time: 9000.23 seconds
```

```
# FOR 13 FEATURES (ALL)

# MemoryError
# AlgorithmError: 'Sampler job failed!'
```

Image C4: Results for experiment 1, quantum model, multiclass classification

```
# FOR 3 FEATURES #

# Best Combination:
# Feature Map: ZZFeatureMap, Reps: 2, Kernel: FidelityQuantumKernel
# Accuracy: 0.6000, F1 Score: 0.2628, Time: 397.52 seconds
```

```
# FOR 5 FEATURES #

# Combination: Feature Map = ZZFeatureMap, Reps = 1, Kernel = FidelityQuantumKernel
# Accuracy: 0.5333, F1 Score: 0.2558, Time: 370.28 seconds
```

```
# FOR 7 FEATURES #

# Combination: Feature Map = ZZFeatureMap, Reps = 1, Kernel = FidelityQuantumKernel
# Accuracy: 0.6000, F1 Score: 0.2792, Time: 781.96 seconds
```

```
# FOR 9 FEATURES #

# Combination: Feature Map = ZZFeatureMap, Reps = 1, Kernel = FidelityQuantumKernel
# Accuracy: 0.5733, F1 Score: 0.2090, Time: 1348.66 seconds
```

```
# FOR 11 FEATURES #

# Combination: Feature Map = ZZFeatureMap, Reps = 1, Kernel = FidelityQuantumKernel
# Accuracy: 0.5467, F1 Score: 0.1414, Time: 8412.10 seconds
```

```
# FOR 13 FEATURES (ALL)

# MemoryError
# AlgorithmError: 'Sampler job failed!'
```

Image C5: Results for experiment 2, classical model, full size dataset, part 1

```
# FOR 5 FEATURES #

# Best Parameters: {'kernel': 'linear', 'C': 0.25, 'gamma': 'scale', 'tol': 1.0, 'max_iter': 3300}
# Best F1-Score: 0.4170
# Best Accuracy: 0.2642
# Time Taken for Best Combination: 4.18 seconds
```

```
# FOR 10 FEATURES #

# Best Parameters: {'kernel': 'linear', 'C': 0.25, 'gamma': 'scale', 'tol': 1.0, 'max_iter': 2900}
# Best F1-Score: 0.4182
# Best Accuracy: 0.3432
# Time Taken for Best Combination: 4.08 seconds
```

```
# FOR 15 FEATURES #

# Best Parameters: {'kernel': 'linear', 'C': 0.5, 'gamma': 'scale', 'tol': 1.0, 'max_iter': 3700}
# Best F1-Score: 0.5946
# Best Accuracy: 0.6695
# Time Taken for Best Combination: 5.44 seconds
```

```
# FOR 20 FEATURES #

# Best Parameters: {'kernel': 'linear', 'C': 0.5, 'gamma': 'scale', 'tol': 1.0, 'max_iter': 3300}
# Best F1-Score: 0.4330
# Best Accuracy: 0.3428
# Time Taken for Best Combination: 5.03 seconds
```

Image C6: Results for experiment 2, classical model, full size dataset, part 2

```
# FOR 25 FEATURES #

# Best Parameters: {'kernel': 'linear', 'C': 0.25, 'gamma': 'scale', 'tol': 1.0, 'max_iter': 3700}
# Best F1-Score: 0.5043
# Best Accuracy: 0.8043
# Time Taken for Best Combination: 7.40 seconds
```

```
# FOR 30 FEATURES #

# Best Parameters: {'kernel': 'linear', 'C': 0.5, 'gamma': 'scale', 'tol': 1.0, 'max_iter': 3700}
# Best F1-Score: 0.6472
# Best Accuracy: 0.7530
# Time Taken for Best Combination: 6.23 seconds
```

```
# FOR 40 FEATURES #

# Best Parameters: {'kernel': 'linear', 'C': 0.5, 'gamma': 'scale', 'tol': 1.0, 'max_iter': 3300}
# Best F1-Score: 0.6440
# Best Accuracy: 0.8293
# Time Taken for Best Combination: 8.42 seconds
```

```
# FOR 50 FEATURES #

# Best Parameters: {'kernel': 'linear', 'C': 1, 'gamma': 'scale', 'tol': 1.0, 'max_iter': 3300}
# Best F1-Score: 0.6524
# Best Accuracy: 0.8297
# Time Taken for Best Combination: 15.77 seconds
```

Image C7: Results for experiment 2, classical model, 3 features

```
### FOR 3 FEATURES ###

# size 1000:
'''
Best Parameters Based on Average F1-Score:
{'kernel': 'linear', 'C': 0.25, 'gamma': 'scale', 'tol': 1.0, 'max_iter': 2900}
Best Average F1-Score: 0.4646
Best Average Accuracy: 0.5148
Total Time for Best Parameter Combination: 1.82 seconds
'''
# size 750:
'''
Best Parameters Based on Average F1-Score:
{'kernel': 'linear', 'C': 0.25, 'gamma': 'scale', 'tol': 0.1, 'max_iter': 3300}
Best Average F1-Score: 0.4524
Best Average Accuracy: 0.5320
Total Time for Best Parameter Combination: 2.03 seconds
'''
# size 500:
'''
Best Parameters Based on Average F1-Score:
{'kernel': 'linear', 'C': 0.5, 'gamma': 'scale', 'tol': 1.0, 'max_iter': 2900}
Best Average F1-Score: 0.4773
Best Average Accuracy: 0.5278
Total Time for Best Parameter Combination: 1.00 seconds
'''
# size 350:
'''
Best Parameters Based on Average F1-Score:
{'kernel': 'linear', 'C': 0.5, 'gamma': 'scale', 'tol': 1.0, 'max_iter': 3700}
Best Average F1-Score: 0.4574
Best Average Accuracy: 0.5670
Total Time for Best Parameter Combination: 0.91 seconds
'''
# size 200:
'''
Best Parameters Based on Average F1-Score:
{'kernel': 'linear', 'C': 0.25, 'gamma': 'scale', 'tol': 1.0, 'max_iter': 2900}
Best Average F1-Score: 0.4240
Best Average Accuracy: 0.5380
Total Time for Best Parameter Combination: 0.56 seconds
'''
```

Image C8: Results for experiment 2, classical model, 5 features

```
### FOR 5 FEATURES ###

# size 1000:
'''
Best Parameters Based on Average F1-Score:
{'kernel': 'linear', 'C': 1, 'gamma': 'scale', 'tol': 0.1, 'max_iter': 3300}
Best Average F1-Score: 0.4793
Best Average Accuracy: 0.5462
Total Time for Best Parameter Combination: 1.90 seconds
'''
# size 750:
'''
Best Parameters Based on Average F1-Score:
{'kernel': 'linear', 'C': 1, 'gamma': 'scale', 'tol': 1.0, 'max_iter': 3700}
Best Average F1-Score: 0.4768
Best Average Accuracy: 0.5770
Total Time for Best Parameter Combination: 1.85 seconds
'''
# size 500:
'''
Best Parameters Based on Average F1-Score:
{'kernel': 'linear', 'C': 0.5, 'gamma': 'scale', 'tol': 0.1, 'max_iter': 3700}
Best Average F1-Score: 0.4920
Best Average Accuracy: 0.5605
Total Time for Best Parameter Combination: 0.85 seconds
'''
# size 350:
'''
Best Parameters Based on Average F1-Score:
{'kernel': 'linear', 'C': 0.5, 'gamma': 'scale', 'tol': 0.1, 'max_iter': 3300}
Best Average F1-Score: 0.4069
Best Average Accuracy: 0.5091
Total Time for Best Parameter Combination: 1.10 seconds
'''
# size 200:
'''
Best Parameters Based on Average F1-Score:
{'kernel': 'linear', 'C': 0.25, 'gamma': 'scale', 'tol': 1.0, 'max_iter': 2900}
Best Average F1-Score: 0.5162
Best Average Accuracy: 0.6028
Total Time for Best Parameter Combination: 0.98 seconds
'''
```

Image C9: Results for experiment 2, classical model, 10 features

```
### FOR 10 FEATURES ###

# size 1000:
'''
Best Parameters Based on Average F1-Score:
{'kernel': 'linear', 'C': 0.25, 'gamma': 'scale', 'tol': 1.0, 'max_iter': 3700}
Best Average F1-Score: 0.5135
Best Average Accuracy: 0.6062
Total Time for Best Parameter Combination: 1.87 seconds
'''
# size 750:
'''
Best Parameters Based on Average F1-Score:
{'kernel': 'linear', 'C': 0.25, 'gamma': 'scale', 'tol': 1.0, 'max_iter': 3300}
Best Average F1-Score: 0.5127
Best Average Accuracy: 0.5946
Total Time for Best Parameter Combination: 1.46 seconds
'''
# size 500:
'''
Best Parameters Based on Average F1-Score:
{'kernel': 'linear', 'C': 0.25, 'gamma': 'scale', 'tol': 1.0, 'max_iter': 2900}
Best Average F1-Score: 0.5497
Best Average Accuracy: 0.6333
Total Time for Best Parameter Combination: 0.99 seconds
'''
# size 350:
'''
Best Parameters Based on Average F1-Score:
{'kernel': 'linear', 'C': 0.5, 'gamma': 'scale', 'tol': 0.1, 'max_iter': 3300}
Best Average F1-Score: 0.4277
Best Average Accuracy: 0.5680
Total Time for Best Parameter Combination: 0.74 seconds
'''
# size 200:
'''
Best Parameters Based on Average F1-Score:
{'kernel': 'linear', 'C': 1, 'gamma': 'scale', 'tol': 1.0, 'max_iter': 3700}
Best Average F1-Score: 0.4909
Best Average Accuracy: 0.6460
Total Time for Best Parameter Combination: 0.60 seconds
'''
```

Image C10: Results for experiment 2, classical model, 15 features

```
### FOR 15 FEATURES ###

# size 1000:
'''
Best Parameters Based on Average F1-Score:
{'kernel': 'linear', 'C': 0.5, 'gamma': 'scale', 'tol': 1.0, 'max_iter': 2900}
Best Average F1-Score: 0.4550
Best Average Accuracy: 0.5350
Total Time for Best Parameter Combination: 1.56 seconds
'''
# size 750:
'''
Best Parameters Based on Average F1-Score:
{'kernel': 'linear', 'C': 0.5, 'gamma': 'scale', 'tol': 1.0, 'max_iter': 2900}
Best Average F1-Score: 0.5155
Best Average Accuracy: 0.6234
Total Time for Best Parameter Combination: 1.38 seconds
'''
# size 500:
'''
Best Parameters Based on Average F1-Score:
{'kernel': 'linear', 'C': 0.25, 'gamma': 'scale', 'tol': 1.0, 'max_iter': 3300}
Best Average F1-Score: 0.4872
Best Average Accuracy: 0.6114
Total Time for Best Parameter Combination: 1.15 seconds
'''
# size 350:
'''
Best Parameters Based on Average F1-Score:
{'kernel': 'linear', 'C': 0.25, 'gamma': 'scale', 'tol': 1.0, 'max_iter': 3700}
Best Average F1-Score: 0.4712
Best Average Accuracy: 0.5934
Total Time for Best Parameter Combination: 0.93 seconds
'''
# size 200:
'''
Best Parameters Based on Average F1-Score:
{'kernel': 'linear', 'C': 0.5, 'gamma': 'scale', 'tol': 1.0, 'max_iter': 3700}
Best Average F1-Score: 0.3979
Best Average Accuracy: 0.6016
Total Time for Best Parameter Combination: 0.66 seconds
'''
```

Image C11: Results for experiment 2, classical model, 20 features

```
### FOR 20 FEATURES ###

# size 1000:
'''
Best Parameters Based on Average F1-Score:
{'kernel': 'linear', 'C': 0.25, 'gamma': 'scale', 'tol': 1.0, 'max_iter': 3700}
Best Average F1-Score: 0.5055
Best Average Accuracy: 0.6188
Total Time for Best Parameter Combination: 2.78 seconds
'''
# size 750:
'''
Best Parameters Based on Average F1-Score:
{'kernel': 'linear', 'C': 0.25, 'gamma': 'scale', 'tol': 0.1, 'max_iter': 2900}
Best Average F1-Score: 0.4919
Best Average Accuracy: 0.5609
Total Time for Best Parameter Combination: 1.53 seconds
'''
# size 500:
'''
Best Parameters Based on Average F1-Score:
{'kernel': 'linear', 'C': 0.25, 'gamma': 'scale', 'tol': 1.0, 'max_iter': 3300}
Best Average F1-Score: 0.4980
Best Average Accuracy: 0.5939
Total Time for Best Parameter Combination: 1.56 seconds
'''
# size 350:
'''
Best Parameters Based on Average F1-Score:
{'kernel': 'linear', 'C': 0.5, 'gamma': 'scale', 'tol': 1.0, 'max_iter': 3700}
Best Average F1-Score: 0.4847
Best Average Accuracy: 0.5841
Total Time for Best Parameter Combination: 0.82 seconds
'''
# size 200:
'''
Best Parameters Based on Average F1-Score:
{'kernel': 'linear', 'C': 1, 'gamma': 'scale', 'tol': 0.1, 'max_iter': 3300}
Best Average F1-Score: 0.4432
Best Average Accuracy: 0.6220
Total Time for Best Parameter Combination: 1.78 seconds
'''
```

Image C12: Results for experiment 2, classical model, 25 features

```
### FOR 25 FEATURES ###

# size 1000:
'''
Best Parameters Based on Average F1-Score:
{'kernel': 'linear', 'C': 0.5, 'gamma': 'scale', 'tol': 0.1, 'max_iter': 3300}
Best Average F1-Score: 0.4609
Best Average Accuracy: 0.5954
Total Time for Best Parameter Combination: 3.54 seconds
'''
# size 750:
'''
Best Parameters Based on Average F1-Score:
{'kernel': 'linear', 'C': 0.25, 'gamma': 'scale', 'tol': 1.0, 'max_iter': 2900}
Best Average F1-Score: 0.4631
Best Average Accuracy: 0.6256
Total Time for Best Parameter Combination: 1.32 seconds
'''
# size 500:
'''
Best Parameters Based on Average F1-Score:
{'kernel': 'linear', 'C': 0.5, 'gamma': 'scale', 'tol': 1.0, 'max_iter': 3300}
Best Average F1-Score: 0.4898
Best Average Accuracy: 0.5739
Total Time for Best Parameter Combination: 1.02 seconds
'''
# size 350:
'''
Best Parameters Based on Average F1-Score:
{'kernel': 'linear', 'C': 0.25, 'gamma': 'scale', 'tol': 0.1, 'max_iter': 3300}
Best Average F1-Score: 0.4811
Best Average Accuracy: 0.5861
Total Time for Best Parameter Combination: 0.83 seconds
'''
# size 200:
'''
Best Parameters Based on Average F1-Score:
{'kernel': 'linear', 'C': 0.25, 'gamma': 'scale', 'tol': 1.0, 'max_iter': 3300}
Best Average F1-Score: 0.5462
Best Average Accuracy: 0.6516
Total Time for Best Parameter Combination: 0.73 seconds
'''
```

Image C13: Results for experiment 2, classical model, 30 features

```
### FOR 30 FEATURES ###

# size 1000:
'''
Best Parameters Based on Average F1-Score:
{'kernel': 'linear', 'C': 0.25, 'gamma': 'scale', 'tol': 1.0, 'max_iter': 2900}
Best Average F1-Score: 0.4867
Best Average Accuracy: 0.5604
Total Time for Best Parameter Combination: 1.90 seconds
'''
# size 750:
'''
Best Parameters Based on Average F1-Score:
{'kernel': 'linear', 'C': 1, 'gamma': 'scale', 'tol': 1.0, 'max_iter': 2900}
Best Average F1-Score: 0.5390
Best Average Accuracy: 0.6316
Total Time for Best Parameter Combination: 1.44 seconds
'''
# size 500:
'''
Best Parameters Based on Average F1-Score:
{'kernel': 'linear', 'C': 0.25, 'gamma': 'scale', 'tol': 1.0, 'max_iter': 3700}
Best Average F1-Score: 0.4749
Best Average Accuracy: 0.5707
Total Time for Best Parameter Combination: 1.22 seconds
'''
# size 350:
'''
Best Parameters Based on Average F1-Score:
{'kernel': 'linear', 'C': 0.25, 'gamma': 'scale', 'tol': 1.0, 'max_iter': 3300}
Best Average F1-Score: 0.5196
Best Average Accuracy: 0.6216
Total Time for Best Parameter Combination: 1.18 seconds
'''
# size 200:
'''
Best Parameters Based on Average F1-Score:
{'kernel': 'linear', 'C': 0.5, 'gamma': 'scale', 'tol': 1.0, 'max_iter': 2900}
Best Average F1-Score: 0.5455
Best Average Accuracy: 0.6772
Total Time for Best Parameter Combination: 0.62 seconds
'''
```

Image C14: Results for experiment 2, classical model, 40 features

```
### FOR 40 FEATURES ###

# size 1000:
'''
Best Parameters Based on Average F1-Score:
{'kernel': 'linear', 'C': 0.25, 'gamma': 'scale', 'tol': 0.1, 'max_iter': 3700}
Best Average F1-Score: 0.4793
Best Average Accuracy: 0.5530
Total Time for Best Parameter Combination: 2.44 seconds
'''
# size 750:
'''
Best Parameters Based on Average F1-Score:
{'kernel': 'linear', 'C': 0.25, 'gamma': 'scale', 'tol': 1.0, 'max_iter': 3700}
Best Average F1-Score: 0.4315
Best Average Accuracy: 0.5182
Total Time for Best Parameter Combination: 2.56 seconds
'''
# size 500:
'''
Best Parameters Based on Average F1-Score:
{'kernel': 'linear', 'C': 0.25, 'gamma': 'scale', 'tol': 1.0, 'max_iter': 3700}
Best Average F1-Score: 0.4972
Best Average Accuracy: 0.6066
Total Time for Best Parameter Combination: 1.38 seconds
'''
# size 350:
'''
Best Parameters Based on Average F1-Score:
{'kernel': 'linear', 'C': 0.25, 'gamma': 'scale', 'tol': 0.1, 'max_iter': 3300}
Best Average F1-Score: 0.4978
Best Average Accuracy: 0.6007
Total Time for Best Parameter Combination: 1.44 seconds
'''
# size 200:
'''
Best Parameters Based on Average F1-Score:
{'kernel': 'linear', 'C': 0.5, 'gamma': 'scale', 'tol': 1.0, 'max_iter': 3300}
Best Average F1-Score: 0.5250
Best Average Accuracy: 0.6680
Total Time for Best Parameter Combination: 0.92 seconds
'''
```

Image C15: Results for experiment 2, classical model, 50 features

```
### FOR 50 FEATURES ###

# size 1000:
'''
Best Parameters Based on Average F1-Score:
{'kernel': 'linear', 'C': 0.25, 'gamma': 'scale', 'tol': 1.0, 'max_iter': 2900}
Best Average F1-Score: 0.4801
Best Average Accuracy: 0.5640
Total Time for Best Parameter Combination: 2.49 seconds
'''
# size 750:
'''
Best Parameters Based on Average F1-Score:
{'kernel': 'linear', 'C': 0.25, 'gamma': 'scale', 'tol': 0.1, 'max_iter': 2900}
Best Average F1-Score: 0.4369
Best Average Accuracy: 0.5568
Total Time for Best Parameter Combination: 1.61 seconds
'''
# size 500:
'''
Best Parameters Based on Average F1-Score:
{'kernel': 'linear', 'C': 0.25, 'gamma': 'scale', 'tol': 1.0, 'max_iter': 3700}
Best Average F1-Score: 0.4463
Best Average Accuracy: 0.6034
Total Time for Best Parameter Combination: 1.73 seconds
'''
# size 350:
'''
Best Parameters Based on Average F1-Score:
{'kernel': 'linear', 'C': 0.25, 'gamma': 'scale', 'tol': 1.0, 'max_iter': 2900}
Best Average F1-Score: 0.5085
Best Average Accuracy: 0.6284
Total Time for Best Parameter Combination: 1.28 seconds
'''
# size 200:
'''
Best Parameters Based on Average F1-Score:
{'kernel': 'linear', 'C': 0.25, 'gamma': 'scale', 'tol': 1.0, 'max_iter': 2900}
Best Average F1-Score: 0.5362
Best Average Accuracy: 0.6404
Total Time for Best Parameter Combination: 0.93 seconds
'''
```

Image C16: Results for experiment 2, quantum model, 3 features

```
### FOR 3 FEATURES ###

# size 1000:
'''
Best Combination:
Feature Map: ZZFeatureMap, Reps: 2, Kernel: FidelityQuantumKernel
Accuracy: 0.8520, F1 Score: 0.8374, Time: 3145.52 seconds
'''
# size 750:
'''
Best Combination:
Feature Map: ZZFeatureMap, Reps: 2, Kernel: FidelityQuantumKernel
Accuracy: 0.8511, F1 Score: 0.8442, Time: 1576.92 seconds
'''
# size 500:
'''
Best Combination:
Feature Map: ZZFeatureMap, Reps: 3, Kernel: FidelityQuantumKernel
Accuracy: 0.8320, F1 Score: 0.8043, Time: 920.78 seconds
'''
# size 350:
'''
Best Combination:
Feature Map: ZZFeatureMap, Reps: 3, Kernel: FidelityQuantumKernel
Accuracy: 0.8182, F1 Score: 0.8043, Time: 478.83 seconds
'''
# size 200:
'''
Best Combination:
Feature Map: ZZFeatureMap, Reps: 1, Kernel: FidelityQuantumKernel
Accuracy: 0.7200, F1 Score: 0.6780, Time: 65.36 seconds
'''
```

Image C17: Results for experimente 2, quantum model, 5 features

```
### FOR 5 FEATURES ###

# size 1000:
'''
Best Combination:
Feature Map: ZZFeatureMap, Reps: 3, Kernel: FidelityQuantumKernel
Accuracy: 0.8800, F1 Score: 0.8778, Time: 12193.48 seconds
'''
# size 750:
'''
Best Combination:
Feature Map: ZZFeatureMap, Reps: 2, Kernel: FidelityQuantumKernel
Accuracy: 0.8777, F1 Score: 0.8747, Time: 3900.86 seconds
'''
# size 500:
'''
Combination: Feature Map = ZZFeatureMap, Reps = 1, Kernel = FidelityQuantumKernel
Accuracy: 0.8160, F1 Score: 0.8041, Time: 1377.16 seconds
'''
# size 350:
'''
Best Combination:
Feature Map: ZZFeatureMap, Reps: 1, Kernel: FidelityQuantumKernel
Accuracy: 0.8750, F1 Score: 0.8661, Time: 542.65 seconds
'''
# size 200:
'''
Best Combination:
Feature Map: ZZFeatureMap, Reps: 1, Kernel: FidelityQuantumKernel
Accuracy: 0.8730, F1 Score: 0.8601, Time: 371.38 seconds
'''
```

Image C18: Results for experiment 2, quantum model, 10 features

```
### FOR 10 FEATURES ###

# size 1000:
'''

'''
# size 750:
'''

'''
# size 500:  (Memory Error), only ran reps=1
'''
Combination: Feature Map = ZZFeatureMap, Reps = 1, Kernel = FidelityQuantumKernel
Accuracy: 0.8320, F1 Score: 0.8074, Time: 7407.87 seconds
'''
# size 350:
'''
Best Combination:
Feature Map: ZZFeatureMap, Reps: 1, Kernel: FidelityQuantumKernel
Accuracy: 0.8864, F1 Score: 0.8743, Time: 2930.93 seconds
'''
# size 200:
'''
Combination: Feature Map = ZZFeatureMap, Reps = 3, Kernel = FidelityQuantumKernel
Accuracy: 0.8200, F1 Score: 0.7980, Time: 3235.89 seconds
'''
```

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação

Universidade Nova de Lisboa