QUANTUM-ENHANCED NEURAL NETWORK FOR FORECASTING KENYAN ECONOMIC GROWTH

SAIF KINYORI

A Thesis Submitted to the Graduate School in Partial Fulfillment of the Requirements for the Award of the Degree of Master of Science in Computer Science of Chuka University

CHUKA UNIVERSITY

OCTOBER; 2023

DECLARATION AND RECOMMENDATION

Declaration

This thesis is my original work and has not been presented for any award of a Diploma or conferment of a degree in this or any other institution.

Signature: .... Date: ...6/10/2023...

Saif Kinyori,
SM22/40012/19.

Recommendation

This thesis has been examined, passed and submitted with our approval as the University supervisors.

Signature: .... Date: ...6/10/2023...

Dr. Edna C. Too, PhD,
Chuka University.

Signature: .... Date: ...6/10/2023...

Dr. David D. Mwathi, PhD,
Chuka University.

CHUKA UNIVERSITY
12 OCT 2023
DIRECTOR BOARD OF POSTGRADUATE STUDIES AND RESEARCH
P.O. Box 109-60400, CHUKA.

COPYRIGHT

DEDICATION

This piece of work is dedicated to my grandmother Mwanaisha Mohamed for her unwavering prayers and support, my father, mother and siblings for their inspirational words of encouragement accorded to me during my period of study. May God bless you all.

ACKNOWLEDGEMENT

I wish to express my gratitude to the Divine Creator for bestowing upon me the vigor and resolve to initiate and finalize this endeavor. I also extend my sincere thanks to Chuka University for providing me with a chance to pursue my studies in an environment so conducive and supportive during my master's degree journey. Blessings to this premier and prophetic University.

My profound appreciation goes to my mentors, Dr. Edna Too Chebet and Dr. David Gitonga Mwathi, for their unwavering support, mentorship, and motivation, as well as their constructive and honest feedback on this thesis. Their encouragement, insights, and invaluable contributions are deeply valued.

I also wish to extend my gratitude to all members of the Computer Science and ICT departments who facilitated my research and various consultations. My heartfelt thanks also go to my peers, family, and friends for their perpetual help, guidance, and support throughout my scholarly journey. To everyone, I encapsulate my gratitude by saying, may God bless you all.

ABSTRACT

Machine learning has had success in solving real world problems using classical computers. Since its adoption, it has undergone tremendous algorithms improvements. One of the most important advancements in this area of computer science is deep learning. Deep learning has already outperformed humans in a number of fields, e.g stock market forecasting. On the other hand, the data that machine learning algorithms consume becomes more complex and keeps on growing through the use of personal computers and mobile phones. Deep learning algorithms have been employed in these data analytics to come up with trends or classifications that can be translated into actionable results that are useful in many areas. However, these very large or complex datasets take a very long time to train. This is due to the fundamentals of classical computing operations in processing data in the basic binary of 0s and 1s. Quantum computers run on qubits and researchers have been able to prove that they have an advantage over the current classical computers in processing of data. Therefore, this study employed experimental quantum-enhanced paradigms to develop a quantum-enhanced neural network model to forecast Kenyan economic growth. It took advantage of quantum-enhanced simulators, currently in place to experiment on the efficiency of data analysis. Analysis on Kenyan economy indicators datasets from the World Bank was used to evaluate the performance and how fast actionable results and economic growth forecasts can be obtained. The data was transformed to a dimension of quantum vector to allow for the data dimensions to be within the boundaries for quantum computers. The quantum-enhanced neural network model demonstrated a computing time reduction of approximately 97.7% when compared to the artificial neural network model, demonstrating a remarkable increase in efficiency. The model mean absolute error indicated a relatively small average deviation of 0.01047 from the actual values. In addition, the mean squared error indicated a low average squared deviation of 0.00025 from the true values. The discrepancy from the expected and actual values was 0.99775, which indicated a high degree of predictability and a strong fit of the model to the data. The quantum-enhanced neural network model demonstrated an overall good performance in all areas of the forecast study.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS AND ACRONYMS

AI:	Artificial Intelligence
ANN:	Artificial Neural Network
DLP:	Deep Learning Processors
FPGA:	Field programmable gateway array
GDP:	Gross Domestic Product
GPU:	Graphics Processing Units
LOQC:	Linear Optical Quantum Computing or Linear Optics Quantum Computation
ML:	Machine Learning
PPP:	Purchasing Power Parity
QC:	Quantum Computer
QGA:	Quantum Generic Algorithm.
QNN:	Quantum Neural network
SVM:	Support Vector Machine
QML:	Quantum Machine Learning

CHAPTER ONE
INTRODUCTION

1.1 Background of the Study

1.1.1 Economic Forecasting

Technology keeps on changing at enormous speed and the generation of complex sets of data is on the rise (Ghani *et al.*, 2019). People are able to upload and download huge volumes of entertainment videos from websites like YouTube, post photos on social networks like Facebook and Instagram. People are also able to surf the web and purchase items online and all these avenues generate massive amounts of data (Mugdha Ghotkar & Rokde, 2016). In order to find hidden patterns, correlations, and other insights, data analytics investigates these complex or massive amounts of data (Talia & Marozzo, 2016). Finding knowledge from unstructured data gathered from the internet is difficult since this material is intended for human consumption rather than machine consumption (Chiroma *et al.*, 2019). Additionally, exploiting these incredibly complex datasets requires a significant amount of training, testing, and validation effort.

China, the world's largest digital market, actively engages in value creation activities from these types of data (Zeng & Glaster, 2018). Firms in the country have been able to compete globally and locally leading to efficient payment systems that are secure and cashless (Oussous *et al.*, 2018). Additionally, better and improved security through all the data collected via street cameras where crime is isolated and hot spots identified (Ferreira *et al.*, 2019). Furthermore, improved urban planning hence the rise of skyscrapers and multiple cities across the country (Zeng & Glaster, 2018). Consequently, this has led to a positive impact to the country's GDP and PPP growth and now the country is one of the largest economies in the world (Ferreira *et al.*, 2019).

These data avenues have the potential to provide value over time by enhancing things like smart tourist destinations (Vecchio *et al.*, 2018). Accordingly, data gathering from user-generated material, together with its aggregation, interconnection, analysis, and integration, real-time synchronization, and intelligent use of data, can be used to create a smart experience (Qiu *et al.*, 2016). According to (Buhalis & Amaranggana, 2015), this is also described as an experience that is mediated by technology and improved by personalisation, context awareness, and real-time monitoring.

Furthermore, data analysis offers substantial benefits within the public and governmental domains (Lake, 2017). Governments that opt to share their extensive datasets foster the exchange of information, enhance transparency, and cultivate trust among their constituents (Ukhalkar, 2018). Leveraging data provided by government agencies, encompassing aspects like crops, weather, and geography, holds the potential to enhance existing approaches in agriculture and industry (Kastouni & Lahcen, 2020). Similarly, the examination of hospital admission data in conjunction with demographic and census information has the potential to enhance healthcare effectiveness in underserved regions (Hardy & Maurushat, 2017).

1.1.2 Machine Learning and Data Analytics

Data approaches and technologies come from a variety of disciplines, including economics, computer science, statistics, and applied mathematics (McKinsey, 2018). These conventional approaches, which include econometrics, A/B testing, statistical models, and mathematical formulas, are primarily successful in resolving linear or nearly linear issues, as well as some difficult nonlinear time-varying issues with restrictions (Chiroma *et al.,* 2019). According to Lake (2017), these data analysis techniques perform poorly when dealing with computational complexity, ambiguity, and inconsistencies. This poses a considerable challenge in devising effective techniques and resources for managing computational intricacy (Muthulakshmi & Udhayapriya, 2018). Owing to their heightened capability in tackling complex real-world issues in comparison to conventional methods, computational intelligence algorithms such as artificial neural networks are garnering increasing interest due to the limitations of traditional approaches (L'Heureux *et al*., 2017).

Researchers are using more complex techniques to comprehend and make sense of the exponentially growing amount of data (Zhou *et al.*, 2017). In this analysis, artificial intelligence, a subfield of computer science that deals with giving machines intelligence, has shown considerable promise in recent years (Ertel, 2018). Data analysis has relied heavily on machine learning methods (Qiu *et al.,* 2016). Artificial neural networks have made considerable advances in big data processing over the years, and this has had a substantial impact on the field of deep learning (Buscema *et al.,* 2018).

In an effort to represent the capacities in processing data with critical qualities, artificial neural network research is being conducted (Mamatha *et al.,* 2017). Neural networks consider a solution to the computation problem. They are information processing structures that provide the frequently overlooked relationship simulating the physiological structure and operation of human brain structures (Buscema *et al.,* 2018).

In data analytics with artificial neural networks, huge or complex volumes of data are involved in research and researchers have to perform cross validation to archive the best optimal results (Srinivasan *et al.,* 2019). However, this process takes a lot of time running on classical machines (Berrar, 2018). Cross validation is the process where a repetition of training data sets is performed a number of times and the accuracy results are weighted (Sze *et al.,* 2017).

Deep learning has emerged as a prominent sub-field within machine learning, focusing on algorithms that acquire multi-level representations to effectively capture intricate data relationships (Gulli & Pal., 2017). This approach employs deep architectures, known as deep neural networks, to construct models (Kim, 2017). Many of these models are constructed through unsupervised learning of representations (Ahmad *et al.,* 2019). Additionally, these learned models contain multiple tiers, each corresponding to distinct conceptual levels, where higher-level concepts are constructed from lower-level constituents, and a single lower-level idea might give rise to numerous higher-level concepts (Bengio *et al.,* 2013).

Field-programmable gate arrays (FPGAs), application-specific integrated circuits (ASICs), and graphics processing units (GPUs) that are specifically designed for machine learning are special machine learning chips that are employed in data analytics (Khan & Mann, 2020). Like all technologies, there are advantages and tradeoffs to every solution. Overall, researchers have to choose the best technology based on their specific end application as these chips perform better in specific environments and applications (Sze *et al.,* 2017). The amount of data that need to be stored and processed in machine learning chips is much larger than that used in common applications before (Wani *et al.,* 2020). This creates a huge demand and is reflected not only in the demand of machine learning applications, but also in the new computing paradigm required by machine learning, especially deep learning (Qolomany *et al.,* 2016). The demands require more efficient hardware to process intelligent computing and analysis, and we also encounter some bottlenecks in the current technical framework, especially the von Neumann bottlenecks, CMOS process and device bottlenecks (Li *et al.,* 2019). Therefore, most of the hardware architecture innovations for machine learning, especially for accelerating neural network processing, are struggling with this problem (ICFC, 2018). This presents an opportunity to explore the quantum realm.

1.1.3 Quantum Computing

Quantum computers harness certain enigmatic properties of quantum mechanics (Alexeev *et al.,* 2021), a scientific realm delving into the behavior of matter and light on the atomic and subatomic level, to bring about significant advancements in processing capability (Giles, 2019). Quantum machines are anticipated to surpass even the mightiest classical supercomputers (Bourassa *et al.,* 2021). Traditional computers rely on bits, which are sequences of electrical or optical signals representing 1s or 0s (Wolf, 2017). In contrast, quantum computers employ qubits, often subatomic particles like electrons or photons (Zhou *et al.,* 2020). A grouped ensemble of qubits can process information far more expeditiously than an equivalent number of binary bits (Castelvecchi, 2017). These properties, specifically entanglement and superposition, distinguish qubits (Zhou *et al.,* 2020).

Quantum computers utilize quantum bits or qubits to represent and process information, guided by the principles of quantum mechanics (Tacchino *et al.,* 2020). These computers have the ability to operate within an immensely expanded computational domain, with resource requirements growing at a polynomial rather than exponential rate, as dictated by specific quantum phenomena (Alvarez *et al.,* 2018). Algorithms that can be effectively executed on quantum computers hold the promise of achieving exponential speedup compared to the most advanced classical methods currently in use (Martonosi & Roetteler, 2019).

1.1.4 Sub-Saharan Africa Economy

Sub-Saharan Africa is characterized by diverse economies, cultures, and natural resources (Morrison *et al.,* 2023). As countries in this region strive for economic growth and development, understanding key economic indicators, such as Gross Domestic

Product (GDP) and Purchasing Power Parity (PPP), is essential (Kouton, 2019). GDP is a crucial measure of economic activity, whereas PPP provides insights into the relative purchasing power of currencies across countries.

The GDP is a fundamental indicator used to gauge the overall economic performance of a country or region (Ajayi *et al.,* 2020). In sub-Saharan Africa, GDP plays a vital role in assessing economic growth, income distribution, and living standards (Doan, 2019). Countries in this region face unique challenges including poverty, political instability, and inadequate infrastructure, which can significantly impact GDP growth rates (Djomo *et al.,* 2023). Studies have examined the factors influencing GDP in Sub-Saharan Africa and the impact of foreign direct investment (FDI) and trade openness on GDP growth in selected countries in Sub-Saharan Africa (Iyke & Odhiambo, 2020). The results show that FDI and trade openness have positive effects on GDP growth, highlighting the importance of external economic factors in the region.

PPP, or purchasing power parity, is a method employed to assess the relative purchasing power of different currencies, taking into account the price levels of goods and services in varying nations (Okelele *et al.,* 2022). Within sub-Saharan Africa, where a multitude of currencies are utilized, PPP offers valuable insights into the comparative affordability of commodities and the overall cost of living among different countries (Doan, 2019). A comprehensive grasp of PPP holds crucial significance for trade, investment, and economic strategizing within this region. The ramifications of PPP on the actual exchange rate within sub-Saharan Africa underscore the necessity of integrating PPP considerations into economic policies, thereby fostering sustainable and well-rounded economic development (Fosu, 2019).

Economic forecasting is a critical process that helps policymakers, investors, and businesses make informed decisions (Mahomed, 2022). Accurate economic forecasts can lead to better resource allocation, increased investments, and overall economic stability (Kouton, 2019). However, forecasting in sub-Saharan Africa poses unique challenges because of the region's economic and political complexities (Morrison *et al.,* 2023). Studies have explored various economic forecasting models to enhance the prediction accuracy. Machine learning techniques have been utilized to forecast GDP growth in selected countries in sub-Saharan Africa (Mahadeo & Heinlein, 2018). The studies have also demonstrated the potential of machine learning algorithms for improving forecasting accuracy in the region.

Understanding and projecting future economic performance, especially Gross Domestic Product (GDP) growth, depends heavily on economic forecasting (Stock & Watson, 2021). The value of all the products and services generated inside a nation's boundaries within a certain time period is tracked by the main statistic of GDP. For governments, entrepreneurs, and investors to make wise judgments and develop future plans, accurate GDP estimates are crucial.

Predicting GDP involves the examination of diverse economic indicators, such as consumer expenditure, investment, governmental outlays, and net exports, (Kalamara *et al.,* 2022). These factors are integrated into various analytical frameworks, including econometric models, time series analysis, and machine learning algorithms, to anticipate forthcoming economic performance (Smith *et al.,* 2020). The Integrated Moving Average (ARIMA) model stands out as an effective tool for capturing temporal patterns and trends within GDP data (Smith *et al.,* 2020). Additionally, more sophisticated machine learning techniques like support vector machines and artificial neural networks are gaining prominence in economic prediction. These advanced methods are adept at recognizing intricate nonlinear relationships and correlations within data, thus enhancing the precision of GDP projections (Lee & Lee, 2019).

Economic forecasting has also been enhanced by the availability of big data and the expanding use of sophisticated data analytics (Stock & Watson, 2021). Forecasting models include real-time data from a number of sources, such as social media, online transactions, and satellite pictures, to provide more accurate and timely predictions (Diebold & Yilmaz, 2020). Data analytics has many uses, one of which is forecasting economic growth (Chen et al., 2021). Researchers and practitioners have recently given it a lot of attention (Shahbaz *et al.,* 2021).

Making macroeconomic policy is one of the main uses of economic growth forecasting (Zhou *et al.,* 2021). Governments use economic growth forecasts to formulate policies that promote economic stability and growth. If economic growth is predicted to slow down, policymakers may implement measures to stimulate economic activity, such as reducing interest rates or increasing government spending (Boone *et al.,* 2019). Conversely, if economic growth is expected to accelerate, policymakers may take measures to prevent overheating of the economy, such as raising interest rates or implementing fiscal austerity measures (Chen *et al.,* 2021).

Investors also use economic growth forecasts to identify investment opportunities in different sectors of the economy (Li *et al.,* 2021). Businesses use economic growth forecasts to make informed decisions about production, marketing, and investment (Zhou *et al.,* 2021).

A developing field of study is the application of quantum computing to economic forecasting (Martonosi & Roetteler, 2019). Complex optimization and prediction issues may be solved by quantum-enhanced algorithms more quickly than by conventional computers (Harrow *et al.,* 2019). Enhancing GDP projections may be possible as a result of new insights into economic patterns and trends provided by quantum machine learning techniques like quantum neural networks.

In conclusion, this research seeks to forecast Kenyan economic growth using complex data with a hybridized quantum enhanced framework. This is a crucial application area which helps policymakers, investors, and businesses make informed decisions in managing economic risks for the Kenyan economy.

1.2 Statement of the Problem

Deep learning algorithms have revolutionized data analytics by enabling the modeling of complex trends and classifications, offering actionable insights into various domains. However, the process of data classification, particularly with large or complex datasets, poses significant challenges owing to the time-consuming nature of the training models and running cross-validation iterations. Training times can be extended to days or even weeks, impeding the efficiency and agility of the overall experiment. To address this issue, specialized chips such as Graphics Processing Unit and Field-Programmable Gate Array have been introduced to enhance the performance of machine learning and data analytics. Although these chips offer some improvements, they remain limited in their versatility and are designed for specific tasks. Additionally, they encounter a Von Neumann bottleneck, where the data transfer rate is constrained by the relatively slower processor speed, hindering the overall throughput of the computer system. Efforts to devise a more effective and efficient solution are crucial to overcome the challenges posed by lengthy training times and hardware limitations. A scalable and versatile approach is required to accelerate data classification tasks and enable faster iterations, thus empowering researchers and practitioners with timely and accurate results. With the introduction of quantum computers, it provides for a chance to explore its computing power to accelerate research for faster and accurate data analysis.

1.3 Objectives of the Study

1.3.1 General Objective

To create a fast and efficient data analytics model using quantum-enhanced machine learning to enhance Kenyan economic growth forecasting.

1.3.2 Specific Objectives

i. To efficiently map classical computer-based dataset to quantum dataset
ii. To develop a quantum enhanced neural network model to forecast Kenyan economic growth
iii. To evaluate the efficiency of the quantum enhanced neural network model for forecasting Kenyan economic growth.

1.4 Research Questions

i. How can classical computer-based dataset be efficiently mapped to quantum dataset?
ii. How can a quantum-enhanced neural network model be developed to accurately forecast Kenyan economic growth?
iii. How does the performance of quantum-enhanced neural network compare to classical neural network in forecasting Kenyan economic growth?

1.5 Justification of Study

There has been tremendous development in machine learning (Huang *et al.*, 2021). There has also been a significant advancement in the hardware used by deep learning algorithms to solve problems in machine learning (LeCun, 2019).

However, with all these advances, most large or complex problems take days to solve with the current computing infrastructure available (Lilienfeld, 2018). These problems lead to inconclusive results or results that do not paint a complete picture of the desired study (Najafabadi *et al.*, 2015). Researchers have been trying to determine whether we can harness the power of the quantum computers in data analysis, whether the same rules apply, or whether we need to create very different rules to solve current and future problems (Huang *et al.*, 2021). This study contributed to the development of quantum machine learning algorithms. The study shows with paring quantum gates and appending them in artificial neutral network, significant results can be obtained. With the fields of quantum computing and quantum machine learning being relatively young, there is room for further research on various algorithms. Classical data handling, encoding and decoding can also be seamlessly transacted on quantum environments, which can improve data analysis. The study also shows that the developed quantum enhanced neural network model can generalize for other countries.

1.6 Scope

The research study utilized a dataset from Word Bank and using Quantum-enhanced Neural Network to forecast Kenyan economic growth. This will enable policymakers to make informed decisions that promote economic stability and sustainable development for the Kenyan economy. A quantum environment was used to develop and test a quantum-enhanced neural network model. The main aim of the model was to test its efficiency given a very complex or large sample size. The model was then used to fit the data and show some predictive analysis based on the data used in the study. The results were tabulated and presented showing the different efficiency metrics.

1.7 Assumptions of the Study

i. Data input for the Quantum Neural network model was unclassified.
ii. Numeric values used were statistically identical to the real-world scenario.
iii. The quantum environment was free from noise interference.

1.8 Limitation

This study was limited to a cloud-simulated environment due to the nature of noise present in physical quantum computing machines.

1.9 Operational Definition of Terms

Artificial Neural Network	Artificial Neural Networks are potent and adaptable machine learning algorithms that can learn from data to create predictions and categorizations and replicate how the brain operates.
Deep Learning.	Technique of building complex multi-layered neural networks.
Machine Learning	An application of AI that enables systems to automatically learn from experience and get better over time without explicit programming.
Quantum Computing	Utilizing collective quantum state properties, such as superposition and entanglement, to perform calculations.
Quantum Machine Learning	The incorporation of quantum algorithms into machine learning software.
Quantum Neural Network	an algorithm with multiple layers that transfers input from one layer of qubits to another layer of qubits. Qubits in this layer evaluate this data and transmit the results to the layer below.

CHAPTER TWO
LITERATURE REVIEW

2.1 General Overview of Machine Learning

Machine learning represents an application of artificial intelligence (AI) where systems are endowed with the capability to autonomously learn from experience and enhance their performance without explicit programming (Rebala et al., 2019). The focal point of machine learning lies in crafting computer programs that can access and process data to acquire self-learning abilities (Yurtoğlu, 2018). Machine learning facilitates the analysis of extensive data sets, furnishing swift and precise insights for the identification of profitable opportunities or potential risks; however, it may necessitate a considerable investment of time and resources for adequate training (Mahesh, 2020). Furthermore, machine learning is regarded as a collection of techniques for extracting valuable insights from data, thereby expediting the advancement of data-centric approaches in anomaly detection, diagnosis, and prognosis (Kang & Jameson, 2018). In view of these considerations, this chapter introduces the fundamentals of deep learning, quantum computing, comprehensive data analysis, and machine learning. Figure 2.1 illustrates the classic isolated learning paradigm in machine learning, where the learned model is used in its intended application.

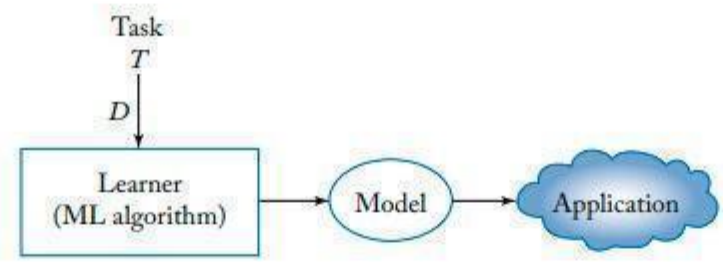


Figure 1: Classic Machine Learning Pattern .

2.1.1 Machine Learning Categories.

According to the quantity and type of supervision they require during training, machine learning can be categorized into the following three categories: supervised, unsupervised, and reinforcement learning (Kadhim, 2019).

In the context of supervised learning, the training dataset includes data paired with their corresponding correct outcomes for a specific task (Louridas & Ebert, 2016).

Supervised learning can be likened to providing a student with a collection of problems along with their solutions and instructing them to deduce strategies for solving upcoming challenges (Shrestha *et al.,* 2018). Classification techniques encompass a variety of tools such as logistic regression, classification trees, support vector machines, random forests, artificial neural networks (ANNs), and other machine learning algorithms (Binkhonain & Zhao, 2019).

Unsupervised learning involves utilizing an unlabeled training dataset to generate visual representations of complex, high-dimensional data in either two-dimensional (2D) or three-dimensional (3D) formats, which can be readily visualized (Kang & Jameson, 2018). Reinforcement learning pertains to the scenario where an entity acquires action knowledge through iterative experimentation within a dynamic setting (Chen & Liu, 2018). During each interaction step, the entity receives input conveying the current environment state, selects an action from an array of possibilities, and subsequently alters the environment state (Shrestha *et al.,* 2018). Following this, the entity receives a value reflecting the outcome of this state change, encompassing rewards or penalties. This cyclic procedure unfolds as the entity constructs a sequence of actions to optimize its goal. The fundamental objective of reinforcement learning involves the acquisition of an optimal policy, facilitating the mapping of states to actions that ultimately maximize cumulative rewards over an extended duration (Chen *et al.*, 2020).

2.1.2 Deep Learning

Deep learning constitutes a subset of the broader realm of machine learning methodologies, centering on artificial neural networks (Learning, 2020). Characterized by the integration of multiple tiers within the network, deep learning involves each stratum mastering the conversion of its input data into a slightly more intricate and composite depiction (Bengio *et al.,* 2013). Deep learning, categorized as a distinct domain within machine learning, hinges on algorithms geared towards the acquisition of multilayered representations, essential for capturing intricate data relationships (Kelleher, 2019). This approach establishes higher-level facets and concepts through the lens of more elemental ones, resulting in a hierarchy termed a deep architecture (Dong *et al.*, 2021). Notably, these models predominantly rely on unsupervised learning for representation, thereby delineating a deep neural network as an artificial neural network encompassing multiple strata positioned between the input and output layers (Ahmad *et al.*, 2019).

2.1.3 Artificial Neural Network

An artificial neural network constitutes a component within a computational framework devised to emulate the cognitive processes of the human brain when deciphering and handling data (Brynjolfsson, 2017). These structures facilitate the elucidation of the (often concealed) correlation between input and output data, emulating the physiological framework and operational mechanics of human brain components (Buscema *et al.*, 2018). The inherent neural arrangement naturally forms a sophisticated network, engendering intelligent behaviors through intricate interactions among interlinked components (Kelleher, 2019).

2.1.3.1 Feedforward Neural Network

In a feedforward neural network, data propagation occurs unidirectionally, proceeding solely from the input layer to the output layer, potentially traversing intermediate hidden nodes (Asteris *et al.*, 2017). Such networks are devoid of any cyclic or self-referential patterns (Shrestha, 2019).

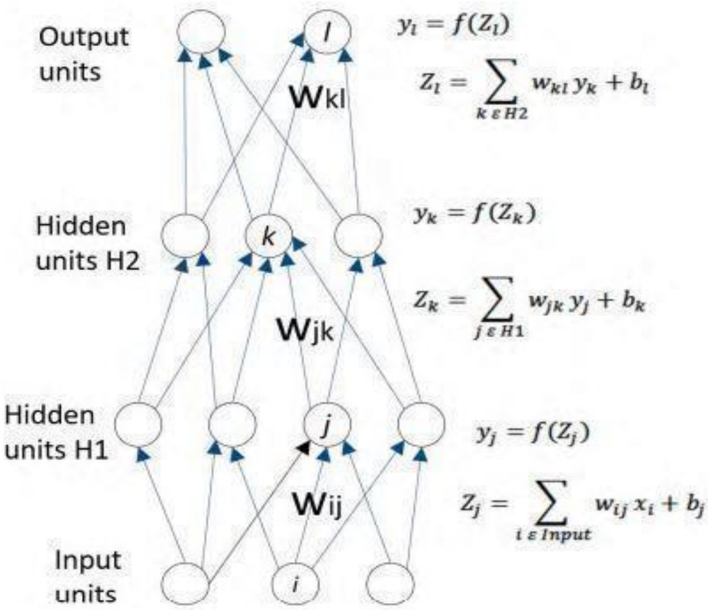


Figure 2:Implementation of a Multilayer Feedforward Neural Network

Figure 2 depicts the execution of a multilayer feedforward neural network, performing computations and applying functions during the forward pass process. The aggregated input's weighted summation is denoted as Z, and the non-linear activation function f acting on Z at each layer is denoted as y. The interconnecting weights among units in adjacent layers, labeled with subscripts, are symbolized as W, while the unit's bias value is depicted as b.

2.1.3.2 Autoencoder

An autoencoder stands as a neural network that operates through an unsupervised algorithm, learning the data representation within the input dataset for the purpose of dimensionality reduction while also reconstructing the original dataset (Zhang et al., 2020). The underlying learning mechanism follows the principles of backpropagation. Autoencoders extend the foundational concept of principal component analysis (PCA) (Tschannen *et al.*, 2018), yet diverge by incorporating nonlinear representations. Unlike PCA, which identifies linear variables along the directions of maximal variance (Ladjal *et al.*, 2019), autoencoders can delve deeper into nonlinearity.

PCA projects original data points onto principal directions, thus excluding information present in corresponding orthogonal directions (Manning, 2018). In contrast, autoencoders employ encoder and decoder structures composed of nonlinear hidden layers, effectively generalizing PCA for dimensionality reduction and ultimate data reconstruction. These autoencoders undergo layer-by-layer unsupervised pre-training followed by fine-tuning through back-propagation (Alsenan *et al.*, 2020). Despite utilizing backpropagation, which is typically associated with supervised training, autoencoders remain unsupervised deep neural networks as they restore input x (i) itself rather than distinct target values y (i), specifically y (i) = x (i). The success of autoencoders was highlighted by Hinton et al., who achieved nearly flawless image reconstruction using a 784-pixel autoencoder, surpassing the capabilities of PCA. Figure 3 illustrates the architecture of the autoencoder.

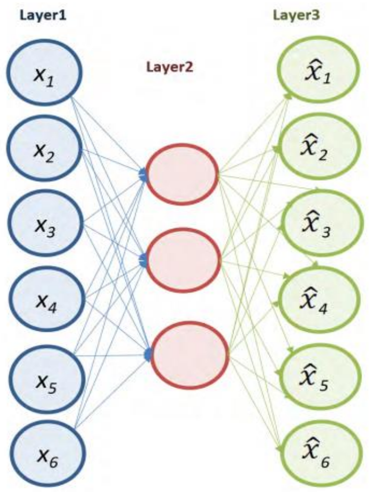


Figure 3: Autoencoder

2.2 Quantum Computing

A Quantum Computer harnesses certain enigmatic phenomena of quantum mechanics (Gyongyosi & Imre, 2019) to achieve significant leaps in computational capability (Giles, 2019). While classical computers employ sequences of electrical or optical signals to represent 1s and 0s, Quantum computers operate with qubits (Bassman et al., 2021), typically embodied by subatomic entities like electrons or photons. Qubits exhibit quantum attributes, enabling a correlated set to offer more computational power compared to an equivalent number of classical binary bits (Henriet et al., 2020). These attributes encompass superposition and entanglement (Bernhardt, 2019).

Quantum computers utilize the principles of quantum mechanics to represent and manipulate data as quantum bits or qubits (Huang *et al.,* 2020). Exploiting specific quantum properties, a quantum computer can function within an immensely vast computational realm while maintaining a resource requirement that grows polynomially (Orus *et al.*, 2019). Quantum algorithms suitable for implementation on such computers have the potential to provide substantial acceleration, sometimes reaching exponential levels, compared to the current leading classical methodologies (Martonosi & Roetteler, 2019). Fig 4: The Quantum von Neumann architecture.

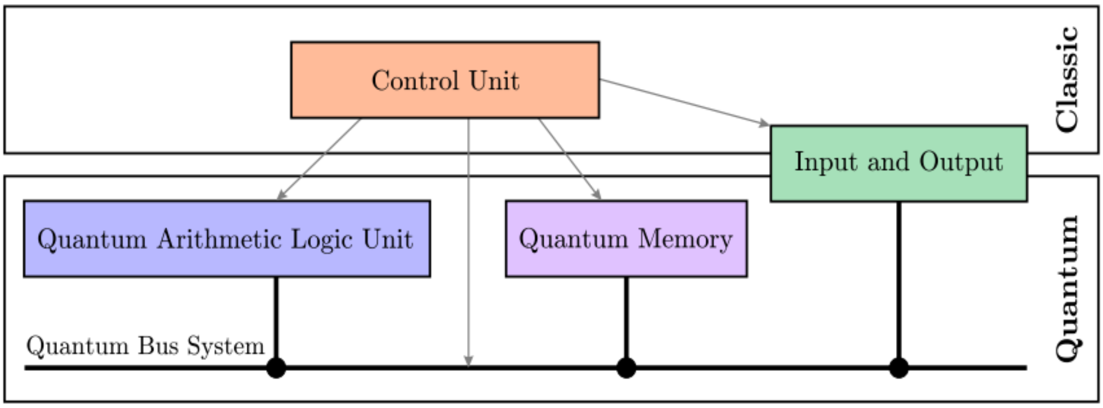


Figure 4:Quantum Von Neumann Architecture of a quantum computer

2.2.2 Techniques used in Quantum Data mapping of Classical Data

One of the many challenges in quantum computing is the mapping of classical data to qubits (Runeson & Richardson, 2020). Classical computing data were presented in bits. This unit of information is stored and processed in Classical Computing (Binary Digital systems) (McCaskey et al., 2018). Classical computers use binary or 2-bit system (Li et al., 2018). 2-bits are 1 and 0. A 2-bit system is considered to be a computer processing or computing the data and instructions or programs from the user and uses 0 and 1 for execution (Ayanzadeh et al., 2019). Today's classical computers include smartphones, laptops, desktops, and servers (Gyongyosi & Imre, 2019).

Quantum computing data: Quantum computers introduce ternary or base-3 systems (Gomes, 2018). These systems rely on the principles of quantum physics (Mavroeidis *et al.*, 2018). Quantum computers employ binary bits (0 and 1) while also enabling a bit to exist in a state of both 0 and 1 simultaneously, known as superposition (Belenchia *et al.*, 2018), thereby introducing a third distinct state. This expanded set of states includes 0, 1, and the superposed state of 0 and 1. Quantum bits, or qubits, serve as the fundamental units of quantum information, capable of embodying both 0 and 1 simultaneously (Anastopoulos & Hu, 2020). This third state enhances processing capabilities (Lu *et al.*, 2018).

2.2.2.1 Basic Encoding

Basis encoding finds its main application in scenarios where quantum algorithms need to perform arithmetic operations with real numbers (Schuld *et al.*, 2021). This method involves the conversion of real numbers into binary numbers, which are then translated into a quantum state within a computational basis (Yu *et al.*, 2018). In the context of basis encoding for numerical data points, the actual value is approximated by its corresponding binary representation. Consequently, the resultant bitstring corresponds to the encoding of the value ABCD in the |ABCD> state (Schuld, *et al.*, 2021).

2.2.2.2 Amplitude Encoding

In this approach, information is embedded within the amplitudes of a quantum state (Sierra et al., 2020). When quantum algorithms prioritize tasks other than arithmetic manipulation, more concise data representations come into play. Specifically, the expansive Hilbert space of quantum devices is harnessed for such encoding purposes (Phillipson, 2020). Utilizing this encoding method, log2(n) qubits are necessary to represent an n-dimensional data point. An illustration of this concept can be observed in figure 2.5, where two qubits are employed to encode 4-dimensional data points.

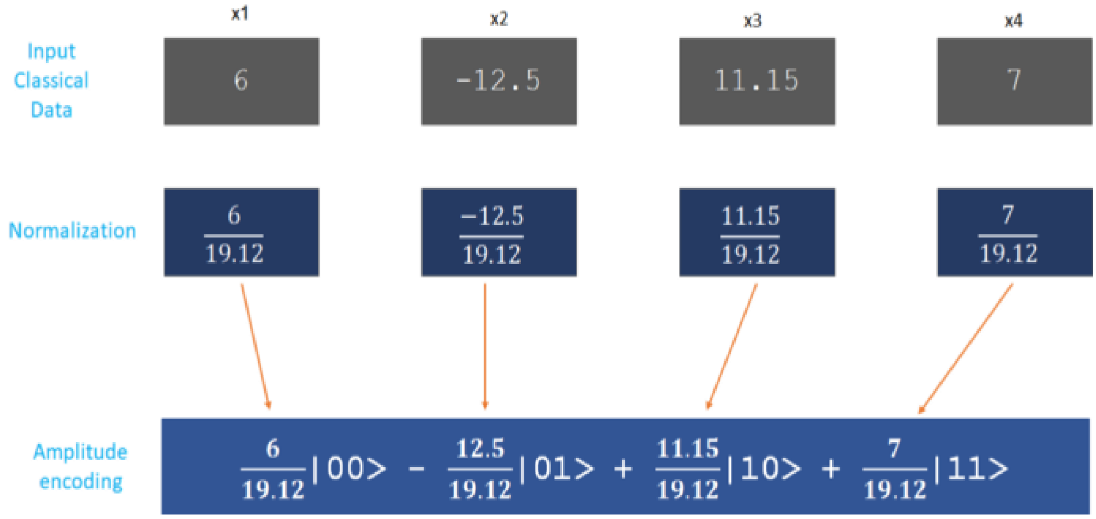


Figure 5: Amplitude encoding

2.2.2.3 Angle Encoding

The fundamental approach to embedding classical data within a quantum state is angle encoding (Ryu & Kang, 2018). In this technique, the rotational angle of qubit n is employed to encode the corresponding classical features. Sometimes referred to as tensor product encoding (He *et al.*, 2018), angle encoding demands n qubits to represent n-dimensional data, yet it offers cost-effective preparation in terms of complexity, requiring just one rotation per qubit. This encoding strategy finds utility in quantum neural networks for data processing (Hoang *et al.*, 2018). To execute angle encoding, a gate rotation around the x-axis (v) or y-axis Ry(v) is applied, with v representing the value for encoding. When applied to the input in the amplitude encoding example, angle encoding (employing a y-axis rotation) would take on the appearance below after state preparation. The Angle Encoding process is depicted in figure 6.

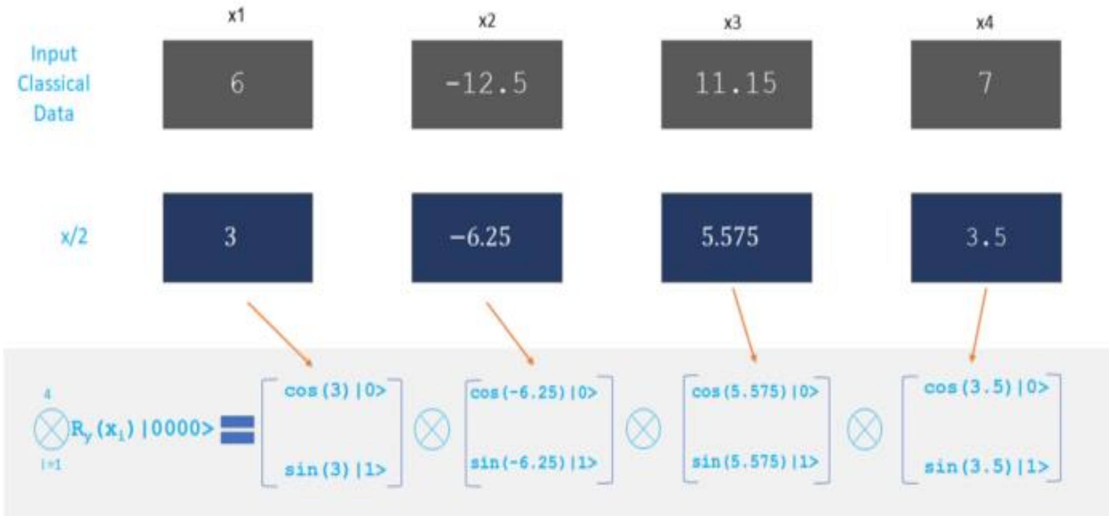


Figure 6: Angle Encoding

This encoding method offers a key advantage in its operational efficiency, requiring a consistent number of parallel operations irrespective of the quantity of data values being encoded (Cui *et al.*, 2018). However, this efficiency doesn't translate optimally to qubits, as each input vector component necessitates a separate qubit. An associated approach, termed dense angle encoding, capitalizes on an additional feature of qubits involving relative phase (LaRose & Coyle, 2020), enabling the encoding of n data points using only n/2 qubits.

2.2.2.4 Quantum Associative Memory

This encoding strategy relies on the principle of superposition to encode a collection of data points within a qubit register of uniform length (Shapoval & Calafiura, 2019). Achieving this demands a binary portrayal of equitably sized values or the application of zero padding. Quantum associative memory (QuAM) is harnessed to establish a superposition of values encoded in the same qubit register format (Njafa & Engo, 2018). The quantum register represents an evenly distributed superposition of the basis-encoded values. The figure.7 shows a depiction of the Quantum Associative Memory.

Input variable	Input Classical Data	Binary Number	Basis encoded Quantum Data	QUAM encoded value
X1	10	1010	$\|1010>$	$\frac{1}{\sqrt{3}}\|1010> + \frac{1}{\sqrt{3}}\|1111> + \frac{1}{\sqrt{3}}\|1000>$
X2	15	1111	$\|1111>$	
X3	8	1000	$\|1000>$	

Figure 7:Quantum Associative Memory

Because QuAM's final encoding is digital, it can be used for mathematical calculations.

2.2.2.5 Qsample Encoding

Qsample encoding represents a synthesis of both basis and amplitude encoding approaches (Schuld & Petruccione, 2018). Qsample integrates a real amplitude vector with conventional discrete probability distributions. Notably, while amplitude encoding is employed, all the features are concurrently encoded within the qubit (Schuld & Petruccione, 2018). The process of state preparation mirrors that of amplitude encoding for a specified probability distribution. When amalgamating two quantum states via a tensor product to construct a composite system, the resulting binary string is sampled as a product of the two original probabilities (Sergioli, 2020). In essence, the qsample of two joint qsamples yields the product distribution.

2.2.2.6 Divide-and-Conquer Encoding

Divide-and-conquer encoding demonstrates an exponential time advantage when populating an N-dimensional vector, facilitated by a quantum circuit featuring polylogarithmic depth and entangled information in auxiliary qubits (Araujo *et al.*, 2021). The outcomes reveal the feasibility of employing a divide-and-conquer approach to effectively input data into quantum devices, effectively trading computational time for spatial resources (Yuliana & Chang, 2020). For further insights on this encoding method, readers are encouraged to explore the referenced section of the research paper.

2.2.2.7 Dimensionality Reduction

This process involves converting data from a high-dimensional space to a lower-dimensional one (Reddy *et al.*, 2020), aiming to preserve significant attributes of the initial data in the reduced representation, ideally closely mirroring its inherent dimensionality (Zhang *et al.,* 2019). Selecting the optimal technique for dimensionality reduction is not straightforward, and there is no predetermined alignment of techniques with specific problems. Rather, a recommended strategy entails conducting well-structured and controlled experiments to determine which dimensionality reduction methods, when coupled with your chosen model, yield the highest performance on your dataset (Zhang *et al.,* 2019).

2.2.3 Challenges of Mapping Classical Data in Quantum State

Harnessing the potential of quantum computation necessitates the conversion and preparation of classical data into a quantum superposition state (Anastopoulos & Hu, 2020). The foundational technique for achieving a quantum superposition state is known as basic encoding (Weigold *et al.,* 2020); however, this approach demands substantial resources, with its complexity escalating exponentially with the number of qubits (Weigold *et al.,* 2020), thereby posing considerable computational challenges in practical scenarios (Biswas *et al.*, 2017).

A key theoretical concern in the realm of quantum computation and mapping is quantum noise, which has the potential to lead to classical probabilistic computations (Lee *et al.,* 2017). The exploration of quantum error correction codes (Brun, 2019) and the development of fault-tolerant quantum computing concepts have been crucial endeavors in addressing this issue (Brun, 2019).

2.2.4 Quantum Algorithms

Quantum computers are engineered to surpass conventional computers through the utilization of quantum algorithms (Thapliyal & Munoz-Coreas, 2019). Quantum algorithms necessitate the construction of circuits involving quantum gates. Thanks to the inherent characteristics of quantum hardware, a quantum circuit retains its information throughout computations and remains reversible (Thapliyal & Munoz-Coreas, 2019).

2.2.4.1 Deutsch’s Algorithm

Deutsch's algorithm is a deterministic procedure designed to address a somewhat artificially constructed problem:

Given a function f: {0, 1} → {0, 1}, determine if f is balanced (i.e. if f(0) ?= f(1)) or constant (f(0) = f(1)).

Calculating f (0) and f (1) and comparing them is a simple solution that can be accomplished by any classical computer. A quantum computer, however, only needs to do one evaluation of the provided function to get a result because a quantum system can exist in several states at once. The figure 8 illustrates the Deutsch's Algorithm.

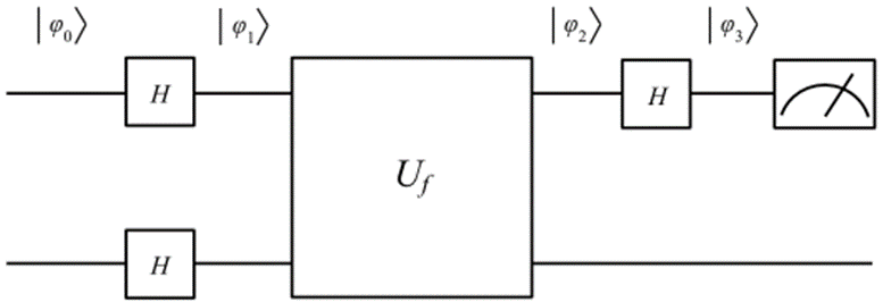


Figure 8: Deutsch's Algorithm

Multiple results can be obtained by a single evaluation thanks to a quantum system's ability to complete all necessary computations in parallel (Portugal, 2022). This method demonstrates how quantum algorithms and quantum computers can solve some problems significantly more quickly than their classical counterparts with the least amount of manipulation necessary to produce a useful result (Zhou *et al.,* 2023). Although this problem's answer is not particularly useful, it does demonstrate the potential of quantum computing and the basic principle of how the characteristics of quantum-mechanical systems might be exploited to accelerate the resolution of some problems. The Deutsch algorithm is depicted in figure 8.

2.2.4.2 Grover's Algorithm

Grover's algorithm, another quantum algorithm, is employed for unstructured database searching (Wang *et al.,* 2020). This problem cannot be solved by searching for every item using classical computation in less than linear time. While classical algorithms must test the indices one at a time, quantum algorithms can use quantum parallelism to examine several indices at once. Grover's technique is therefore much faster and can search a database with N items in only O(√ N) steps, according to Grover's technique is therefore much faster and can search a database with N items in only O(√ N) steps, according to Kerenidis *et al.*, (2019).

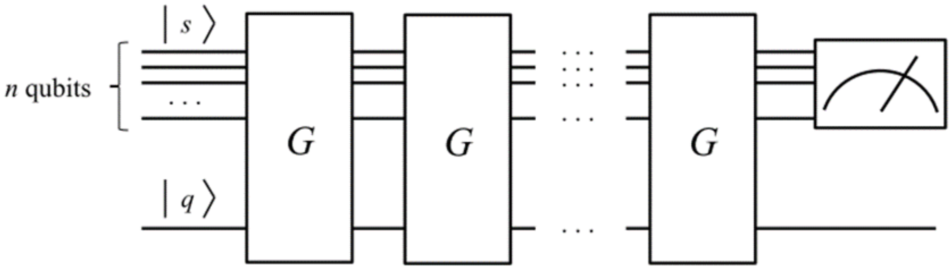


Figure 9: Grover’s Algorithm

Grover's algorithm is shown in figure 9 and can be used to solve a number of issues, including the collision problem and the molecular distance geometry problem as well as determining the mean and median of a set. It can be crucial in cryptography as well.

2.2.4.3 Shor’s Algorithm

The fastest quantum factoring algorithm, known as Shor's factoring algorithm, takes just O((logN)3) time, in contrast to the greatest classical factoring techniques that need O(e1,9(logN)3 (log logN)3) time. The most well-known quantum algorithm was created in 1994 by Peter Shor, and its development led to the development of quantum computers and the investigation of other quantum algorithms. The factoring issue is best formulated as follows:

The order-finding subroutine in Shor's algorithm is the quantum portion, which requires a quantum computer to execute. The classical portion of Shor's algorithm can be run on a classical computer. A model of a quantum circuit for Shor's algorithm is shown in Figure 10.

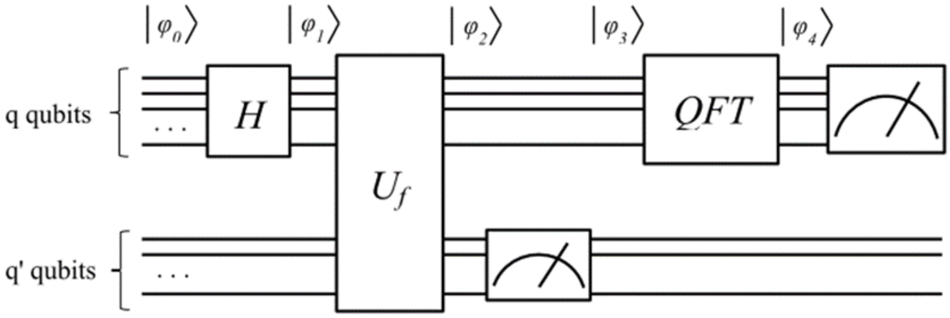


Figure 10: Quantum Circuit for Shor's Algorithm

2.2.5 Quantum Error Correction

The progress in quantum error-correction techniques has allowed us to meet the demands for both high accuracy and sufficient coherence duration in quantum computations (Roffe, 2019). In contrast to classical systems, quantum computation systems exhibit distinct categories of errors (Cai et al., 2021). Despite the emergence of numerous physical approaches to facilitate dependable fault-tolerant quantum computing across a range of quantum hardware realizations, the quest for an all-encompassing universal framework for quantum error correction remains a continuous endeavor.

Quantum error-correction techniques typically utilize input quantum states and syndrome quantum states to identify and correct inaccuracies in the information (Gyongyosi & Imre, 2019). Within experimental quantum computing, a distinction arises between the physical and logical representations of quantum states, as a logical quantum system is encoded across multiple quantum states at the physical level (Roffe, 2019). Quantum error-correction codes, including strategies like topological error correction schemes, can be customized to fit various physical implementations, adapting to specific needs (Sivak *et al.,* 2023).

2.3 Quantum Machine Learning

Positioned at the crossroads of quantum physics and machine learning, quantum machine learning represents an emerging field characterized by interdisciplinary investigation (Schuld & Killoran, 2019). This concept primarily pertains to the enhancement of machine learning through quantum techniques, wherein classical data is subjected to machine learning algorithms within the context of quantum computing. Quantum computers harness phenomena like quantum coherence and entanglement to carry out information processing that surpasses the limitations of conventional computers (Broughton *et al.,* 2021).

The continuous advancement in building more powerful quantum computers has been progressing steadily (Zhang & Ni, 2020). Quantum algorithms, executed on quantum computers, provide a systematic approach to solving problems like database search (Biamonte *et al.,* 2017). These algorithms serve as the foundation for quantum machine learning software in data processing. Notably, in certain scenarios, quantum algorithms have the theoretical potential to outperform classical counterparts (Biamonte *et al.,* 2017), underscoring the emerging domain of quantum machine learning as a transformative force in advancing the challenges and applications of artificial intelligence (Dunjko *et al.,* 2017).

2.3.1 Evolution of Quantum Machine Learning Algorithms

The origins of quantum machine learning can be identified in the 1990s, marked by researchers' exploration of quantum algorithms for distinct machine-learning objectives (Dunjko & Wittek, 2020). A pivotal early advancement emerged with the introduction of the quantum nearest-neighbor algorithm by (Biamonte *et al.,* 2008), showcasing the application of quantum methodologies to enhance distance-based machine-learning processes (García *et al.,* 2022).

2.3.1.1 Quantum Support Vector Machines (QSVM)

Support Vector Machines (SVMs) are a popular class of classical machine learning algorithms (Khan & Robles-Kelly, 2020). The adaptation of SVMs to quantum computing has led to the development of quantum support vector machines (QSVM) (Garg & Ramakrishnan, 2020). Researchers have demonstrated that QSVM can offer quadratic speedup compared to classical SVMs for certain tasks (Lloyd *et al.,* 2013).

2.3.1.2 Quantum Data Encoding and Quantum Feature Maps

Crucial components in quantum machine learning algorithms encompass quantum data encoding and quantum feature maps (Schuld & Killoran, 2022). Quantum data encoding entails the transformation of classical data into quantum states, while quantum feature maps facilitate the manipulation of classical features within the quantum realm (Abohashima *et al.*, 2020). The skillful development of these techniques for data encoding and feature mapping significantly influences the efficacy of quantum machine learning algorithms.

2.3.1.3 Quantum Neural Networks (QNN)

Quantum neural networks integrate principles from classical neural networks and quantum computing (Khan & Robles-Kelly, 2020). The idea of incorporating quantum circuits as layers within neural networks was initially introduced by Schuld et al. (2014). Quantum neural networks hold promise for enhanced representation and generalization capacities (Dunjko & Wittek, 2020).

2.3.1.4 Variational Quantum Algorithms

Variational quantum algorithms belong to a category of algorithms that utilize quantum circuits to optimize classical cost functions (Schuld, 2021). This group of algorithms, encompassing the quantum approximate optimization algorithm (QAOA) and the variational quantum eigensolver (VQE), have shown potential in tackling optimization challenges and emulating quantum systems (Garg & Ramakrishnan, 2020).

2.3.1.5 Quantum Generative Models

Generative models constitute a subset of machine-learning algorithms designed to produce novel data samples based on a specified distribution (Martyn et al., 2021) Within this realm, quantum generative models like quantum Boltzmann machines (QBM) and quantum generative adversarial networks (QGANs) have exhibited promise in generating samples from intricate quantum distributions (Cerezo et al., 2021).

2.3.1.6 Hybrid Quantum-Classical Algorithms

Hybrid quantum-classical algorithms amalgamate quantum and classical computations to address optimization and machine learning challenges (Carrasquilla, 2020). These methodologies harness quantum resources for specific tasks while relying on classical resources for the remaining computations, optimizing the overall computational process (Martín-Guerrero & Lamata, 2022).

2.3.1.7 Quantum Transfer Learning

Transfer learning is a strategy that utilizes acquired knowledge from a particular task to enhance performance in a distinct yet interconnected task. Quantum transfer learning is an emerging field that investigates the application of quantum insights gained in one domain to another domain. Despite the progress made in the QML algorithms, few challenges remain (Schuld, 2021). These challenges include the need for more efficient quantum hardware, development of more error-correction techniques, and scalability of QML algorithms for large datasets (Garg & Ramakrishnan, 2020). In addition, the potential for quantum advantages over classical methods in specific applications requires further exploration (Zhang & Ni, 2020).

The development of quantum machine learning algorithms has been a voyage marked by exploration, innovation, and collaboration between the quantum computing and machine learning communities (Abohashima *et al.,* 2020). As quantum computing technologies continue to advance, the potential for quantum machine learning to reshape various industries becomes increasingly apparent (Schuld & Killoran, 2022). However, much remains to be uncovered and comprehended in this rapidly progressing domain. Future investigations in quantum machine learning hold the promise of pushing the boundaries of both quantum computing and machine learning, opening doors to novel breakthroughs and transformative applications (Carrasquilla, 2020).

2.3.2 Quantum Neural Network

Scientists have investigated several quantum machine learning algorithms that mirror their classical equivalents (Jia *et al.,* 2019), some achieving exponential acceleration and others attaining quadratic speedup. Nonetheless, a clear advantage of quantum neural networks over their classical counterparts has yet to be established (Zhang & Ni, 2020). Figure 11 illustrates the quantum neural network.

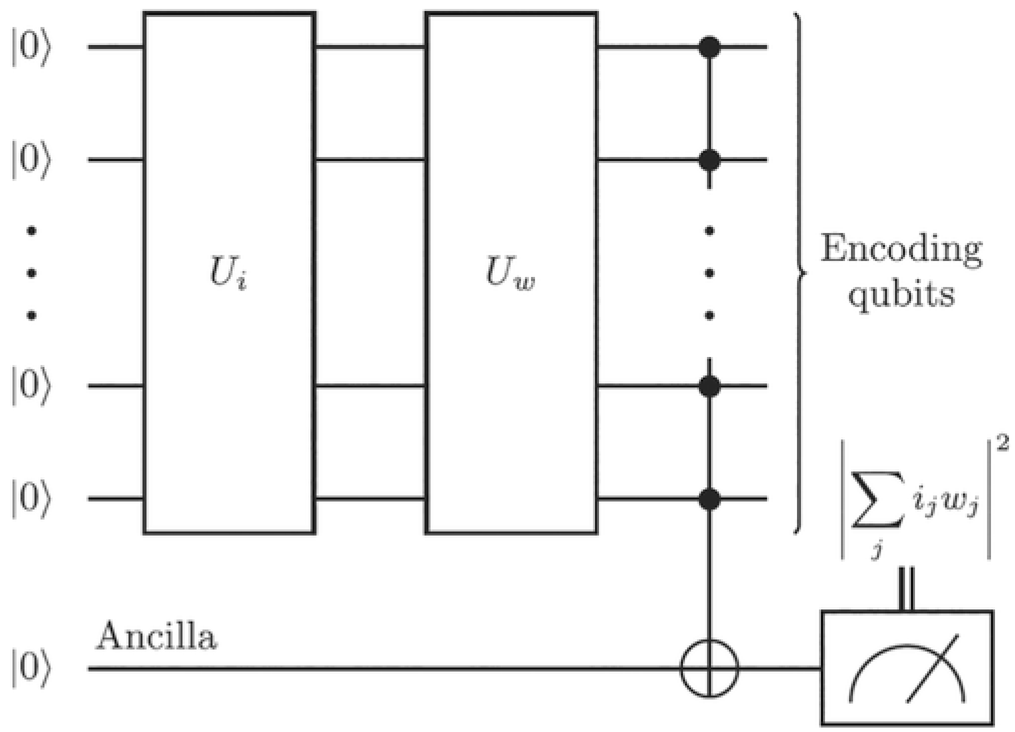


Figure 11: The Quantum Scheme of Artificial Neuron on Quantum Processor

Quantum neural networks operate in parallel (Killoran *et al.,* 2019), and theoretically, these networks allow for the combination of multiple activation functions, a scenario where traditional neural networks tend to experience performance reduction and heightened complexity (Zhong *et al.,* 2020).

The fundamental component of a Neural Network is the perceptron layer, and its quantum counterpart should effectively address both linear and nonlinear challenges (Abbas *et al.,* 2021). Quantum Concepts involve a blend of linear aspects (superposition calculus) and nonlinear elements (probability-based state approximation) (Bausch, 2020). To establish a quantum perceptron, it's essential to introduce a transformation (activation function) for nonlinearity within certain bounds, and this is facilitated by the phase estimation algorithm.

2.3.3 Big Data

Traditional data sources, including automatically generated sensor data, business data, social data, and information from myriad devices like cameras, smartphones, laptops, and images, offer a rich resource for generating insights (Oussous *et al.*, 2018). Data is quantifiable in units like megabytes and gigabytes, or even larger scales such as terabytes and petabytes (Yaqoob *et al.*, 2016). The term "big data" encompasses vast volumes of data that can be structured, semi-structured, or unstructured (Purohit, 2017). When conventional databases and software technologies struggle to manage such data loads, it is categorized as big data (Olivera *et al.,* 2019). Consequently, big data denotes the realm of digital information that exceeds the capacities of traditional data management methods (Borgman, 2017). This encompasses extensive volumes of digital data, necessitating varying velocities based on specific application demands. Moreover, big data exhibits diverse data types and origins, tailored to the unique characteristics of each organization (Khine & Shun, 2017).

Data is categorized as structured data that we can process and store in a fixed format, unstructured data one that has no specific format or semi structured a combination of the previous two. It also has volume which grows exponentially as (Patgiri & Ahmed, 2017) suggests. It also has velocity that always increases even if we use compressing technology. It also has variety and takes different forms and variety depending on how accurate it is. Finally, data has value and hence many researchers try to make meaning for every data collected. (Allam & Dhunny, 2019)
Big Data = Data + Value

2.3.3.1 Techniques and Models for Data Analysis

There are several practical models for predictive analytics on data.

2.3.3.1.1 Classification Model

The majority of machine learning algorithms can be categorized into two main groups: classification and regression (Nasteski, 2017). Each category serves distinct predictive analytics purposes; specifically, classification algorithms are employed to categorize data into various classes (Wexler *et al.*, 2019). Such classification models assist organizations in optimizing the allocation of resources, including human resources (Dutta *et al.*, 2020).

2.3.3.1.2 Regression Model

This model is used in statistical analysis where large datasets are needed to determine specific patterns (Maulud & Abdulazeez, 2020). Also, there must be a linear relationship between the inputs (Yildiz *et. al,* 2017). The model creates an expression that shows the specific relationship between all the inputs found in the dataset (Yildiz *et. al,* 2017).

2.3.3.1.3 Neural Network

This model mimics the workings of the human brain (Mahesh, 2020). It supports complex data relationships that apply to AI and pattern recognition (Ahmad & Pothuganti, 2020). This model is a useful tool for problems with large amounts of data that need to understand the relationship between inputs and outputs (Saif *et al.,* 2018), or if you need to anticipate events.

Neural networks are biologically inspired data processing technologies (Silva *et. al.,* 2019) that captures past and present data to estimate future value. Their design allows finding of complex correlations hidden in data in a way that simulates the pattern recognition mechanism of the human brain.

Commonly used in applications such as image recognition and patient diagnostics, multiple receive inputs (input layers) (Kriegeskorte & Golan, 2019), compute predictions (hidden layers) (Cheng *et. al.,* 2021), and provide outputs (output layers) in the form of a single prediction.

2.3.3.1.4 Decision Trees

The model looks like a tree (Kadiyala & Kumar, 2018), the branches of the tree show the available choices, and the individual leaves show the decisions (Praveena & Jaiganesh, 2017). This model is easy to use and saves time when making urgent decisions by predicting the best results in a short period of time (Nasteski, 2017).

2.3.4 Special Machine Learning chips in Data Analytics

Machine learning chips encompass graphics processing units (GPUs), field-programmable gate arrays (FPGAs), and application-specific integrated circuits

(ASICs) meticulously tailored for machine learning applications (Ma *et al.*, 2014). Similar to general-purpose CPUs, the efficiency and swiftness of these machine-learning chips are derived from integrating a vast number of increasingly smaller transistors, which not only operate faster but also consume less energy than their larger counterparts (Al-Jarrah *et al.*, 2015). These specialized chips possess distinct attributes that significantly expedite the repetitive, foreseeable, and independent calculations demanded by machine learning algorithms (Nakata *et al.*, 2017). These features encompass parallel execution of numerous calculations concurrently, a technique diverging from the sequential approach of CPUs; the utilization of low-precision calculations that proficiently execute machine learning algorithms (Qiu *et al.*, 2019), thereby curbing the number of transistors essential for identical computations; optimization of memory access speed, potentially achieved by housing an entire machine learning algorithm within a single specialized chip; and the employment of programming languages meticulously designed to streamline the translation of machine learning code for efficient execution on dedicated chips (Brown *et al.*, 2019).

Nonetheless, distinct categories of these specialized chips prove advantageous for varying functions (Jawandhiya, 2018). GPUs are predominantly employed for the initial development and enhancement of machine learning algorithms, a phase termed as training. FPGAs find their primary utility in the application of trained machine learning algorithms to real-world data inputs, often referred to as inference (Nath *et al.*, 2016). ASICs, on the other hand, can be tailored for either training or inference purposes. This specialization renders these unique chips application-specific rather than possessing general-purpose attributes (Chen *et al.*, 2021). Moreover, the escalating demands of machine learning algorithms in terms of resources necessitate more proficient hardware to handle the computational aspects of machine learning (Chen *et al.*, 2016), resulting in bottlenecks within the current technological framework, particularly the von Neumann bottlenecks and CMOS process and device constraints.

Furthermore, the volume of data to be stored and processed in machine learning chips far surpasses that of typical applications in the past (Galvez *et al.*, 2019). Consequently, the Von Neumann bottleneck becomes increasingly critical in machine learning scenarios (ICFC, 2018). Figure 12 illustrates a visual depiction of the Von Neumann bottleneck.

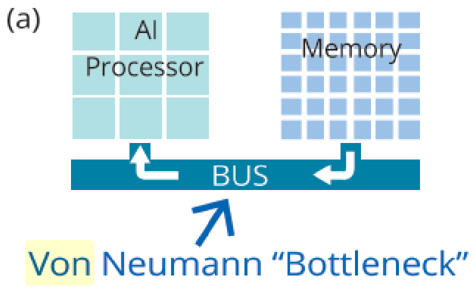


Figure 12: The Von Neumann “bottleneck” in Machine Learning Chips

2.3.5 Forecasting Economic growth with Big Data

Forecasting economic growth with big data has become an increasingly popular research area in recent years (Mohammed, 2020). This approach involves utilizing various techniques to extract insights from vast, complex datasets to produce precise predictions regarding economic trends (Giannone *et. al.*, 2021). The use of big data in economic forecasting offers several advantages, including the ability to capture real-time data, analyze large datasets quickly and accurately, and identify patterns and relationships that may not be evident through traditional methods (Hassani & Silva, 2021).

Some of the primary data sources used in forecasting economic growth with big data include social media data, transactional data, web data, and sensor data (Kang *et al.*, 2021). Social media platforms such as Twitter and Facebook provide real-time data that can be used to track consumer sentiment, purchasing behavior, and other economic indicators (Zhao *et al.*, 2021). Transactional data from credit card companies, retailers, and other sources can be used to track consumer-spending patterns and identify trends in real-time (Cho *et al.*, 2021). Web data can provide insights into consumer behavior, such as search queries and website visits that can be used to forecast demand for particular products or services (Hsieh *et al.*, 2021). Sensor data from IoT devices can be used to track economic activity in real-time, such as traffic patterns and shipping volumes (Lee *et al.*, 2021).

To effectively forecast economic growth with big data, analysts typically use a range of machine learning and statistical techniques, including regression analysis, time series analysis, and neural networks (Zhang *et al.*, 2021). These techniques help to identify patterns in the data, build predictive models, and generate forecasts.

While there are some challenges to using big data in economic forecasting, such as ensuring the accuracy and completeness of the data, the potential benefits are significant (Obschonka & Audretsch, 2020). By using big data to generate more accurate economic forecasts, policymakers and businesses can make more informed decisions and better manage economic risks (Huang *et al.*, 2021).

2.3.6 Challenges in Big Data Analytics

While the potential preferences of Big Data are authentic and immense (Pastorino *et al.,* 2019), and some basic triumphs have started to be proficient, there remain various particular troubles that must have had a tendency to totally comprehend this potential (Hariri *et al.*, 2019).

Big data presents significant analytical challenges that need to be addressed (Elshawi *et al.,* 2018). Analyzing the extensive volumes of data, whether structured, semi-structured, or unstructured, demands a substantial proficiency in advanced techniques. Additionally, the nature of the analysis required is closely tied to the intended outcomes, particularly decision-making (Ang *et al.*, 2020). This can be accomplished through two approaches: integrating extensive data quantities into the analysis or predefining the pertinent big data aspects (Katal *et al.,* 2013).

The foremost obstacle in big data analysis pertains to storage mediums and enhanced input/output speed (Bhattarai *et al.*, 2019). In such scenarios, ensuring data accessibility takes precedence for effective knowledge discovery and representation. Another challenge inherent in Big Data analysis arises from the data's diversity (Galetsi *et al.,* 2019). As datasets continue to expand, the demands on data mining tasks have notably surged. The key challenge here lies in giving more emphasis to the design of storage systems (Vassakis *et al.*, 2018) and advancing efficient data analysis tools that can ensure reliable outputs when dealing with diverse data sources. Moreover, the development of machine learning algorithms for data analysis is imperative to enhance efficiency and scalability (Wang *et al.,* 2018).

Analyzing extensive datasets introduces heightened computational intricacies (Saggi & Jain, 2018). The primary concern revolves around managing the discrepancies and unpredictability inherent in the datasets. Generally, a structured approach involving the systematic modeling of computational intricacies is employed (Zeadally *et al.*, 2019). Establishing a comprehensive mathematical framework that can be universally applied to address the scope of Big Data can be challenging (Khan *et al.*, 2018). Presently, existing tools for big data analysis exhibit suboptimal performance in handling computational intricacies, uncertainty, and disparities (Vassakis *et al.*, 2018). This engenders a significant challenge in devising techniques and technologies capable of effectively managing computational complexities, uncertainty, and disparities.

Within the realm of big data analysis, extensive volumes of data are interconnected, examined, and explored to extract significant patterns (Bell *et al.*, 2021). This process raises concerns regarding information security in the context of big data analytics. Despite considerable research endeavors aimed at enhancing the security of big data, there remains ample room for improvement (Khanan *et al.*, 2019). The primary hurdle lies in the creation of a multi-tiered security framework that ensures privacy while managing large datasets, addressing a crucial challenge in the domain of big data (Katal *et al.*, 2013).

The existing machine learning framework is not exempt from its limitations (Chen & Liu, 2018). Although the promise of Big Data is undeniably substantial, realizing its full potential necessitates a fresh perspective. Conventional machine learning algorithms were primarily tailored for datasets that could be entirely accommodated in memory (Mahesh, 2020). Nevertheless, in the era of Big Data, this premise no longer remains valid (Bao, 2018). Consequently, the demand arises for algorithms capable of learning from extensive volumes of data.

Despite the considerable progress in extensive deep learning, there remains a substantial requirement to tackle numerous critical issues presented by Big Data (Younas, 2019). These challenges are frequently delineated by the three V's framework: volume, variety, and velocity, which correspond to the vast extent of data, diverse data types, and the rapidity of data streaming, respectively.

2.4 Quantum Computing for Big Data Analysis

A quantum computer possesses an exponentially expanding memory capacity relative to its physical size, enabling it to concurrently manipulate an extensive array of inputs (Gyongyosi & Imre, 2019). This exponential enhancement in computing systems is achievable (Orus *et al.,* 2019). With the ongoing advancements in quantum computing, it holds the potential to address exceptionally intricate problems that challenge contemporary computers, including today's prevailing issues with big data (Huang *et al.,* 2020). Quantum computing offers a pathway to incorporate the principles of quantum mechanics into the realm of information processing. In classical computers, information is encoded using lengthy sequences of bits, representing either a zero or a one (Franson, 2013). Conversely, a quantum computer employs quantum bits or qubits (LeCun, 2019).

The differentiation between a qubit and a bit lies in the aspect that a qubit is a quantum system that encodes zero and one into two distinct quantum states (Alexeev *et al.,* 2021). This unique property enables it to harness the phenomena of superposition and entanglement, which stem from the quantum nature of qubits. To illustrate, a quantum system containing 100 qubits requires the storage of 2100 complex values within a classical computer system. Consequently, numerous substantial challenges posed by big data can be addressed significantly more rapidly using larger-scale quantum computers in comparison to their classical counterparts (Preskill, 2018). Thus, the current generation is tasked with the endeavor of constructing quantum computers and facilitating quantum computing to effectively tackle prominent big data issues (Muthulakshmi & Udhayapriya, 2018).

2.4.1 Big Data Analysis with Quantum Neural Networks

Extracting concealed insights and knowledge from extensive datasets holds the potential to elevate our living standards, yet accomplishing this feat is neither simple nor direct (Hu & Hu, 2019). Tackling this intricate and demanding endeavor, which surpasses the capacities of conventional inference and learning methods, necessitates the introduction of fresh technologies, algorithms, and frameworks (Mohammadi *et al.,* 2018). Artificial Neural networks algorithms being among the leading algorithms of deep learning has had great success in recent years (Buscema *et al.*, 2018). However, the main issue that has towered over the success of these algorithms is that organizations do not know how to manage data and conduct research efficiently (Miklosik & Evans, 2020).

Deep learning architectures are crafted through the arrangement of multi-layered artificial neural networks (LeCun *et al.,* 2015). These networks comprise numerous non-linear processing units, referred to as neurons, which often utilize activation functions like the sigmoid function. From a conceptual standpoint, the neurons within each layer can be understood as multiple variables, as elucidated by (Abbas *et al.,* 2021). The process of deep analysis involves a deterministic or stochastic transformation function that maps inputs to outputs, a concept that can be theoretically linked to numerical approximation models, data-fitting functions, and maximum likelihood estimators (Li *et al.,* 2019).

The potential principles that form the basis for constructing multi-layered artificial neural networks encompass the incorporation of multiple layers, variables, and non-linear activation units, which collectively empower these networks with significant aptitude in representing complex datasets (Mangini *et al.,* 2021). Nevertheless, it's important to acknowledge that all-encompassing theories of approximation for deep learning in intricate functions are yet to be fully developed (Nwankpa *et al.*, 2018). In practical scenarios, directly comparing the performance of diverse architectures becomes intricate, especially when their evaluations are not standardized across identical datasets.

In the realm of multivariate multi-layered neural networks, whether fully connected or sparsely connected, a substantial number of parameters necessitate estimation (Basu *et al.,* 2020). These parameters encompass factors like the count of layers, units per layer, as well as the weights and thresholds associated with each activation unit. The pursuit of an optimal solution within the context of data fitting functions or statistical estimators often engenders challenges of ill-posed problems and suboptimal computational performance (Fan *et al.,* 2019). Typically, the adoption of complex models tends to result in overfitting, leading to both computational inefficiencies and the investment of considerable computational time, a point emphasized by (Brownlee, 2018). Consequently, to tackle the issue of overfitting, regularization methods have been devised. Furthermore, a range of numerical algorithms, including pre-training, concave-to-convex approximation, and the computation of gradients across multiple training datasets simultaneously, have been innovated to enhance computational efficiency.

Researchers take quite a significant amount of time in their undertakings and the structures of the amount of big data channeled out on a daily basis keeps on changing (Begam & Raina, 2017). Cross validation has been an effective way to get an unbiased understanding of these data for a while as it describes how well a machine learning model works across different datasets (Hallman, 2019).

Table 1 compares two versions of the entire model and several merger epochs, with the mean value over 20 runs and standard deviation in parenthesis (Miranda & Zuben, 2016). The training duration and speedup depicted apply to the entire training and take into account whether the partitions were trained in parallel (right) or in series (left). This study focused on the test set's 10,000 photos.

Table 1: Reducing Machine Learning Training Time Attempt

	Model	Net2WiderNet classif. error	Partitioning classif. error	Training time (min)	Speedup
Original		1182.95	1177.20	142.93	0.88x
Duplicated		1184.60	1183.25	124.69	1.00x
Merging Iteration	0	1175.75	1180.25	124.69	1.00x
	50	1189.80	1177.55	120.18	1.04x
	100	1187.80	1173.75	115.68	1.08x
	150	1198.80	1178.40	111.18	1.12x
	200	1198.85	1177.05	106.67	1.17x
	250	1200.85	1177.55	102.17	1.22x
	300	1219.60	1184.10	97.67	1.28x
	400	1244.05	1191.75	93.16	1.34x
	450	1278.15	1195.85	85.89	1.45x
	500	1325.95	1257.70	78.62	1.59x

In table 1 the researchers had to split the data then merge each iteration to try and reduce the time of the overall research. Other times researchers add some necessary hyperparameters in the attempt of repeating an experiment for better results (Poggio *et al.*, 2017). These parameters are necessary for the accuracy of the results but they do have an impact on the overall study time. Table 2 Shows this phenomenon and the increment in values of the study as each iteration is conducted (Hallman, 2019).

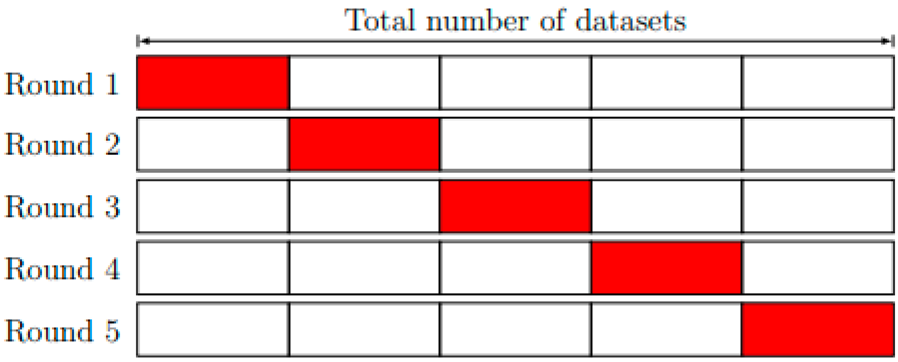
Table 2:Hyperparameter Tuning

Round	Hyperparameter	Values			
Round 1	Learning rate	0.01	-	-	-
	Epochs	60	200	300	-
	Batch size	60	200	300	500
	Hidden components	100	-	-	-
	Dropout	0	-	-	-
	Hidden layers	1	-	-	-
Round 2	Learning rate	0.001	0.01	0.1	-
	Epochs	60	-	-	-
	Batch size	300	-	-	-
	Hidden components	100	-	-	-
	Dropout	0	-	-	-
	Hidden layers	1	-	-	-
Round 3	Learning rate	0.01	-	-	-
	Epochs	60	-	-	-
	Batch size	300	-	-	-
	Hidden components	100	250	300	500
	Dropout	0	0.2	-	-
	Hidden layers	1	2	-	-

2.4.2 Cross Validation in Data Analytics

Cross-Validation is a technique for evaluating how well a machine learning model performs when applied to several datasets (Faker & Dogdu, 2019). Cross-validation has long been used by researchers to determine how well a machine learning model will perform generally for predictions of data that was not utilized during training (Singh & El-Kassar, 2019). A five layers cross validation example is broken down in the figure Table 3.

Table 3: Cross Validation Illustration



To examine how well the models fit and function across various training sets, cross-validation is employed (Lu, 2019). The bias variance trade-off, a statistical learning concept that refers to error prediction, is typically preferred to be either 5 or 10 for k (James *et al.*, 2014).

2.4.3 Assessing Performance of Quantum Neural Network Models

A crucial component of quantum machine learning research is evaluating the performance of quantum neural network models (Shi *et al.*, 2020). Performance assessment involves various metrics such as time, loss functions, convergence rates, and quantum entanglement measures (Du *et al.*, 2021). Additionally, benchmarking against classical neural networks and quantum algorithms helps researchers to understand the advantages and limitations of QNN models (Mangini *et al.*, 2021). The evaluation process involves rigorous testing of quantum hardware or simulators to validate the model's predictive capabilities (Abbas *et al.*, 2021). As quantum computing technology advances, refining the assessment methods for QNN models becomes crucial for realizing the true potential of quantum machine learning in solving real-world problems (Shi *et al.*, 2020).

2.4.3.1 Quantum Neural Network Evaluation Methods

The ability of neural networks to recognize intricate nonlinear correlations in data and efficiently manage sizable and unstructured datasets has increased their attractiveness for application in economic prediction (Keleher, 2019). However, assessing the performance and efficiency of a forecast model is essential to ensure the accuracy of forecasted values. Various evaluation methods that provide different perspectives on the model's accuracy and effectiveness are available (Dunjko & Wittek, 2020).

2.4.3.1.1 Mean Absolute Error

One commonly used metric for evaluating a forecast or prediction model is Mean Absolute Error (MAE). The average absolute difference between the forecasted and actual values is measured by MAE (Jia *et al.*, 2021). It provides a straightforward measure of the accuracy of the model as it calculates the average magnitude of the errors without considering their direction.

2.4.3.1.2 Mean Squared Error

Another widely employed metric is the Mean Squared Error (MSE), which calculates the average of the squared differences between predicted and actual forecast values (Moraffah *at al.*, 2021). MSE penalizes larger errors more heavily, making it particularly useful for identifying outliers or extreme errors in model predictions.

2.4.3.1.3 Root Mean Squared Error

The square root of MSE, called Root Mean Squared Error (RMSE), gives an indication of the average size of the errors in the same unit as the prediction values (Do *et al.*, 2019). RMSE is often preferred when there is a need to interpret the errors in the context of forecast data, as it preserves the unit of measurement.

2.4.3.1.4 Mean Absolute Percentage Error

Mean Absolute Percentage Error (MAPE) is a metric that measures the average percentage difference between predicted and actual forecast values (Ang *et al.*, 2020). MAPE is particularly useful for understanding the relative accuracy of a model's predictions, as it expresses the errors as a percentage of the actual forecast values (Turner *et al.*, 2021).

2.4.3.1.5 R-squared

Another useful evaluation metric that quantifies the percentage of variance in the prediction values explained by the model is the R-squared (R^2) coefficient (Moraffah *et al.*, 2021). R^2 has a range of 0 to 1, with 1 denoting a perfect match between the model and the data. A higher R^2 value denotes a better fit, as it shows that the model explains a greater proportion of the variation in the predicted data (Turner *et al.*, 2021).

2.4.3.1.6 Forecast Bias

Additionally, forecast bias assesses the systematic tendency of the model to overestimate or underestimate forecast values (Do *et al.,* 2019). It is determined by calculating the average difference between the predicted and actual values. A forecast bias close to zero indicates that the model does not exhibit any systematic bias in its prediction (Turner *et al.*, 2021).

2.4.3.2 General Factors Affecting Economic Forecasting

Economic forecasting is a complex task that is influenced by a myriad of factors. Economic forecasting in sub-Saharan Africa is influenced by a multitude of general factors that shape the accuracy and reliability of predictions. These factors encompass the region's unique socio-economic landscape, characterized by diverse currencies and economic systems (Abisuga-Oyekunle *et al.*, 2020). Additionally, the availability and quality of economic data, including variables such as consumer spending, government expenditure, and net exports, significantly impact the forecasting process (Nguimkeu & Zeufack, 2019). Furthermore, the potential for political instability, infrastructural challenges, and variations in global commodity prices introduce inherent complexities into economic forecasting models (Ezeh *et al.*, 2020). These factors underscore the importance of considering a broad spectrum of variables and employing sophisticated modeling techniques to achieve accurate and robust economic forecasts in sub-Saharan Africa.

2.4.3.2.1 Macroeconomic Indicators

Macroeconomic indicators are essential factors influencing economic forecasts and economic growth (Abisuga-Oyekunle *et al.,* 2020). These indicators include the gross domestic product, inflation rate, unemployment rate, consumer spending, government spending, and exports and imports (Heitzig *et al.*, 2021). GDP is considered one of the most critical indicators because it reflects the total economic output of a country over a specific period. Economists closely monitor GDP changes to assess economic performance and anticipate potential recessions or expansions (Naidoo *et al.,* 2020).

2.4.3.2.2 Monetary Policy

Central banks play a significant role in influencing economic forecasts through monetary policy decisions (Heitzig *et al*., 2021). Interest rates, money supply, and credit availability are tools used by central banks to control inflation, boost economic growth, and stabilize the economy (Tenaw & Beyene, 2021). Changes in interest rates can impact consumer spending, investment decisions, and borrowing costs, which, in turn, affect economic growth (Sparks, 2021).

2.4.3.2.3 Fiscal Policy

Government spending and taxation policies, known as fiscal policies, also impact economic forecasts and economic growth (Mensah *et al*., 2021). Governments can use fiscal policies to stimulate economic growth by increasing spending on infrastructure projects or social programs. Conversely, they could implement austerity measures to reduce public debt and control inflation (Nguimkeu & Zeufack, 2019). The effectiveness of fiscal policies in driving economic growth depends on the government's ability to allocate resources efficiently and address structural challenges. (Okou *et al.,* 2022).

2.4.3.2.4 Business and Consumer Confidence

Business and consumer confidence significantly influences economic forecasts (Abisuga-Oyekunle *et al.,* 2020). A high level of confidence among businesses and consumers tends to spur investment, consumption, and economic growth (Ezeh *et al.,* 2020). By contrast, low confidence may result in reduced spending and weaker economic performance (Calderón & Zeufack, 2020). Surveys and sentiment indicators are often used to gauge confidence levels and predict economic trends.

2.4.3.2.5 Global Economic Factors

The global economic environment plays a critical role in determining economic forecasts and economic growth, especially in countries heavily reliant on international trade and foreign investment (Adam *et al.,* 2020). Factors such as geopolitical events, international trade agreements, exchange rates, and global economic trends can impact a country's export performance, foreign direct investment, and overall economic outlook (Fang *et al.,* 2020).

2.4.3.2.6 Technological Advancements

Advancements in technology and innovation have significant implications for economic forecasts and economic growth (Azolibe & Okonkwo, 2020). Technological disruptions can create new industries, enhance productivity and drive economic growth. On the other hand, obsolete technologies and a lack of innovation may hinder economic performance (Valickova & Elms, 2021). Understanding the impact of technology on various economic sectors is vital to accurate forecasting.

2.4.3.2.7 Demographics and Population Trends

Demographics and population trends also play a role in shaping the economic forecasts (Nechifor *et al.,* 2020). Aging populations can result in labor force shortages and increased healthcare and pension costs, thereby affecting economic growth (Avom *et al.,* 2020). Conversely, countries with young and growing populations may experience a demographic dividend, leading to a potential boost in economic productivity (Ezeh *et al.,* 2020).

2.4.3.2.8 Political Stability

Political stability and the quality of governance are essential determinants of economic forecasts (Valickova & Elms, 2021). Political instability, corruption, and policy uncertainty can deter foreign investments and hinder economic growth (Coulibaly *et al.,* 2019). Countries with stable political environments and efficient governance structures are more likely to attract investment and experience sustained economic growth (Valickova & Elms, 2021).

2.4.3.2.9 Natural Disasters and Climate Change

Natural disasters and climate change can have devastating effects on the economy (Okou *et al.*, 2020). Extreme weather events such as hurricanes, floods, and droughts can disrupt production, damage infrastructure, and lead to significant economic losses (Nechifor *et al.,* 2019). As climate change has become a pressing global concern, its potential effects on economic growth are increasingly considered in economic forecasts (Coulibaly *et al.,* 2019).

2.4.3.2.10 External Trade and Global Supply Chains

Globalization has interconnected economies worldwide, making external trade and global supply chains influential factors in economic forecasts (Calderón & Zeufack, 2020). Changes in international trade policies, supply chain disruptions, and global economic shocks have far-reaching effects on economic growth and economic stability (Nechifor *et al.,* 2021).

Economic forecasting and economic growth are influenced by a complex interplay between domestic and international factors (Azolibe & Okonkwo, 2020). Macroeconomic indicators, monetary and fiscal policies, business and consumer confidence, global economic factors, technological advancements, demographics, political stability, natural disasters, climate change, and external trade contribute to economic forecasts (Fang *et al.,* 2020). Accurate economic predictions are critical for policymakers, businesses, and individuals to make informed decisions and to prepare for future economic scenarios. As the global economy continues to evolve, understanding and analyzing these factors has become increasingly essential for successful economic forecasting and effective economic policymaking (Calderón & Zeufack, 2020).

2.4.4 Related Works in Economic Forecasting

Deep learning is one of the most significant developments in computer science (Wani et al., 2020). Almost all scientific disciplines have been impacted. It is already disrupting and transforming businesses and entire sectors. Deep learning has already surpassed human performance in a number of areas, such as predicting movie ratings, deciding whether to approve loan applications, determining how long it takes to deliver a car, and many other areas (Li, 2017).

Previous work, comparisons of performance between quantum machine learning algorithms (Havenstein & Chandrasekaran, 2018) focused on quantum support vector machine algorithm on IBM Qiskits and they concluded that, we are capable of achieving success rates up to 93.00% with quantum compute, but the presence of noise diminishes the results on a real quantum chip especially with the limited number of qubits.

The use of artificial neural network models in forecasting economic growth has been used in Nigeria (Afolabi & Olayemi, 2018). The study trained and tested four different neural network models on a dataset consisting of macroeconomic indicators such as inflation rate, exchange rate, and government expenditure. The results showed that the feedforward neural network model outperformed the other models, suggesting that it is an effective tool for forecasting economic growth in Nigeria (Afolabi & Olayemi, 2018). The study provided insights for policymakers and investors on how neural networks can be used to make more accurate predictions about economic growth in Nigeria (Afolabi & Olayemi, 2018).

There have been advancements in ANN Architectures for Economic Growth Prediction over time (Dunjko & Wittek, 2020), and researchers have extended their focus to enhance ANN architectures for more accurate economic growth predictions. (Pao & Tsai, 2005) introduced a hybrid ANN-Genetic algorithm approach for economic forecasting, optimizing ANN parameters through a genetic algorithm. Their findings revealed improved prediction accuracy compared to traditional models.

Feature selection is a critical step in improving the performance of the forecasting models (Sethi & Mittal, 2019). In the context of economic growth prediction, feature selection helps identify relevant economic indicators that contribute significantly to the accuracy of the model (Bacanin *et al.,* 2021). Deng et al. utilized a feature selection technique based on mutual information to enhance the robustness of an ANN model in GDP forecasting (Deng et al., 2013). Additionally, ensemble methods such as Bagging and Boosting have been applied to combine multiple ANN models to improve predictions (Zhang *et al.,* 2006).

Time-series forecasting requires models that capture temporal dependencies in data (Lim & Zohren, 2021). Recurrent Neural Networks (RNNs) have gained popularity in this area owing to their ability to incorporate past information into the forecasting process (Learning, 2020). Farsi et al. utilized Long Short-Term Memory (LSTM) networks, a type of RNN, to forecast economic growth in the UAE, achieving high accuracy compared to traditional time-series models (Farsi *et al.,* 2018).

Hybrid forecasting models combining ANN with other forecasting techniques have also been explored. Chen & Liu proposed a hybrid model that integrates ANN with ARIMA (Auto Regressive Integrated Moving Average) for economic growth prediction in Taiwan (Chen & Liu, 2015). Their findings indicated that the hybrid model outperformed the individual models, highlighting potential synergistic effects.

2.5 Proposed Quantum and Quantum Neural Network Architecture

The proposed architecture introduces paired gates CNOT and Hadamard in the quantum neural network processing. This architecture got the mapped quantum data and through paired gates of CNOT and Hadamard before evaluation is done. The Figure 13 illustrates the proposed Quantum Neural Network Architecture.

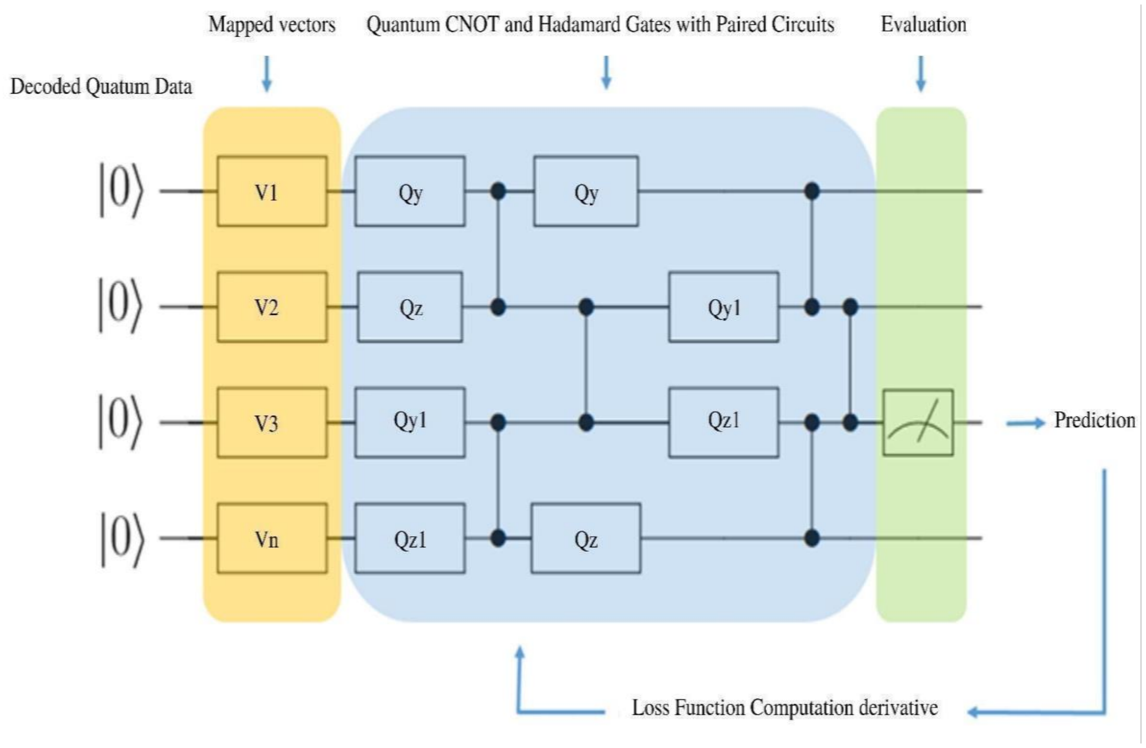


Figure 13: Proposed Quantum and Quantum Neural Network Architecture

CHAPTER THREE

METHODOLOGY

3.1 Study Site

The study site was located at Chuka University main campus, department of computer science, Lab 4. Chuka University is located 186 km from Nairobi along the Nairobi – Meru highway. The University is within Ndagani location, Ndagani sub –location in Chuka Igambang'ombe constituency, Tharaka Nithi County. The study site GPS coordinates were: Latitude: N 50° 24.8469'. Longitude: W 104° 31.2467'.

3.2 Research Design

The research study adopted an experimental design, as the field of quantum machine learning is relatively young (Dunjko & Wittek, 2020). The goal was to determine the highest analysis efficiency from the quantum-enhanced model developed using quantum simulators and neural network architecture. This was obtained from expounding on the open-source shell of the neural network architecture. This approach was also based on the learning mechanism of the neural network architecture and quantum-enhanced cloud environment. The model was provided with a clean dataset of economic indicators that was used in training, testing and validating the study.

3.3 Data Sampling

The research study used the Kenyan economy data from World Bank Datasets. The World Bank database provides a comprehensive collection of economic data (Wang *et al.,* 2020). The data comprised areas such as social media market trends, web traffic search, financial transactions, Kenya poverty metrics, Kenyan oil prices, agricultural and household items pricing, Kenyan supermarkets data, economic impact of Covid-19 and other related economic factors.

This data source was derived from secondary materials and exhibited a robust compilation of dependable and authoritative datasets (Casalicchio, 2017). It encompassed a comprehensive assemblage of more than a thousand yearly economic development indicators from numerous sectors across the country (Capacity, 2014).

The dataset consisted of 30 CSV files of each with over 2000 rows and 20 columns with historical data, forming a complex map suitable for finding economic patterns. The dataset possessed ample dimensions to effectively facilitate the training, validation, and testing stages of the suggested model. This allocation was established at a proportion of 70% for model training, 15% for validation purposes, and the remaining 15% designated for rigorous testing (Genç & Tunç, 2019). Sample datasets sourced from the World Bank are visually depicted in Table 4 and 5.

Table 4:Dataset sample of Kenyan Population Growth Against Birth and Mortality Rates from the Year 1961 to 2021

Series Name	Series Code	1960	1961	1962	1963	1964	1965	1966	1967
Population, total	SP.POP.TOTL	7751435	8047470	8363578	8697200	9047387	9417207	9802605	10201068
Population growth (annual %)	SP.POP.GROW	..	3.74797688	3.852856965	3.911480881	3.947485196	4.006256107	4.01096189	3.9844254
Surface area (sq. km)	AG.SRF.TOTL.K2	..	580370	580370	580370	580370	580370	580370	580370
Population density (people per sq. km of land area)	EN.POP.DNST	..	14.13970201	14.69511544	15.28130161	15.89659311	16.5463805	17.223539	17.9236532
Poverty headcount ratio at national poverty lines (% of population)	SI.POV.NAHC	..	..	..	..	..	..	..	..
Poverty headcount ratio at $2.15 a day (2017 PPP) (% of population)	SI.POV.DDAY	..	..	..	..	..	..	..	..
GNI, Atlas method (current US$)	NY.GNP.ATLS.CD	..	..	836050920.5	924938645.2	973334015.7	999617813.7	1159379221	1206136254
GNI per capita, Atlas method (current US$)	NY.GNP.PCAP.CD	..	..	100	110	110	110	120	120
GNI, PPP (current international $)	NY.GNP.MKTP.PP.CD	..	..	..	..	..	..	..	..
GNI per capita, PPP (current international $)	NY.GNP.PCAP.PP.CD	..	..	..	..	..	..	..	..
Income share held by lowest 20%	SI.DST.FRST.20	..	..	..	..	..	..	..	..
Life expectancy at birth, total (years)	SP.DYN.LE00.IN	48.68	49.533	50.224	50.808	51.328	51.7	52.097	52.455

Table 5: Dataset sample of the World Bank Series Metadata

Code	License Type	Source
SP.POP.TOTL	CC BY-4.0	(1) United Nations Population Division. World Population Prospects: 2022 Revision. (2) Census reports and other statistical publications from national statistical offices, (3) Eurostat: Demographic Statistics, (4) United Nations Statistical Division. Population and Vital Statistics Reprot (various years), (5) U.S. Census Bureau: International Database, and (6) Secretariat of the Pacific Community: Statistics and Demography Programme.
SP.POP.GROW	CC BY-4.0	Derived from total population. Population source: (1) United Nations Population Division. World Population Prospects: 2022 Revision, (2) Census reports and other statistical publications from national statistical offices, (3) Eurostat: Demographic Statistics, (4) United Nations Statistical Division. Population and Vital Statistics Reprot (various years), (5) U.S. Census Bureau: International Database, and (6) Secretariat of the Pacific Community: Statistics and Demography Programme.
AG.SRF.TOTL.K2	CC BY-4.0	Food and Agriculture Organization, electronic files and web site.
EN.POP.DNST	CC BY-4.0	Food and Agriculture Organization and World Bank population estimates.
SI.POV.NAHC	CC BY-4.0	World Bank, Poverty and Inequality Platform. Data are compiled from official government sources or are computed by World Bank staff using national (i.e. country–specific) poverty lines.
SI.POV.DDAY	CC BY-4.0	World Bank, Poverty and Inequality Platform. Data are based on primary household survey data obtained from government statistical agencies and World Bank country departments. Data for high-income economies are mostly from the Luxembourg Income Study database. For more information and methodology, please see http://pip.worldbank.org.
NY.GNP.ATLS.CD	CC BY-4.0	World Bank national accounts data, and OECD National Accounts data files.
NY.GNP.PCAP.CD	CC BY-4.0	World Bank national accounts data, and OECD National Accounts data files.
NY.GNP.MKTP.PP.CD	CC BY-4.0	International Comparison Program, World Bank | World Development Indicators database, World Bank | Eurostat-OECD PPP Programme.
SH.IMM.MEAS	CC BY-4.0	WHO and UNICEF (http://www.who.int/immunization/monitoring_surveillance/en/)?

3.4 Data Pre-Processing

The preprocessing of data involved cleaning, transforming, and organizing data into a structured format. The dataset had inconsistencies, errors and missing values that needed to be addressed before it was used for training. The preprocessing step included tasks of data cleaning, data transformation, normalization, and outlier detection. Preprocessing the data ensured that the data is consistent, complete, and ready to be used for training of the proposed model.

The following steps were followed:

1. Defining the data cleaning objectives: this involved identifying and correcting errors, eliminating duplicate entries, filling missing values, and standardizing formats.
2. Identifying the data quality issues: Here, the data quality issues, such as incomplete or missing data, inconsistent formats, and incorrect values that can affect the accuracy of analysis were corrected.
3. Data validation: Validating the data to ensure its accuracy and completeness. This involved checking the data against predefined rules, such as constraints, data types, and ranges, and identifying any discrepancies.
4. Data transformation: Transforming the data to a standardized format that can be used for analysis. This involved standardizing data formats, merging data from different sources into the files, and converting data types to .csv.
5. Data enrichment: Enriching the data by adding additional information from external sources, such as demographics, geographic data, government policies, political climate trends, or industry benchmarks.
6. Data deduplication: Removing duplicate entries from the dataset to avoid redundancy and improve data accuracy.
7. Data normalization: Normalizing the data to ensure consistency and comparability across different datasets. This involved standardizing data formats, scaling values, and removing outliers.
8. Data documentation: Documenting the data cleaning process and keeping track of all the changes made to the dataset. This helped in ensuring transparency and reproducibility of the analysis.

3.5 System Modeling

To establish entanglement between qubits and generate unique circuit pairs for model processing, a combination of a CNOT gate and a Hadamard gate was employed. The design incorporated a distinct architectural configuration where the neuron count within the encoder layer progressively diminished across successive layers. Conversely, as the model progressed to subsequent layers, the neuron count within the decoder layer experienced an increment. This structured approach encompassed three tiers for both the encoder and decoder components. The encoder contained 64, 32 and 1 units respectively in each level and the decoder contained 1, 32 and 64 units respectively in each level. This provided a symmetric approach. The data was also scaled between 0 and 1 before feeding it into the autoencoder using Minimum Maximum Scaler as the process employed different activation functions in the output layer which outputs values between 0 and 1.

The modeling procedure encompassed inputting historical data into the quantum-enhanced neural network, refining the weights, and fine-tuning parameters to minimize the disparity between predicted and actual outputs. This process incorporated the utilization of backpropagation and gradient descent methods. By training the model, the algorithm acquired the ability to discern patterns within the historical data, facilitating precise forecasting.

The model's effectiveness was assessed through metrics including mean absolute error, mean squared error, and root mean squared error. Additionally, cross-validation was employed to validate the model's efficiency, and the time taken for each iteration was documented. A visual representation of the model's architecture can be seen in Figure 14.

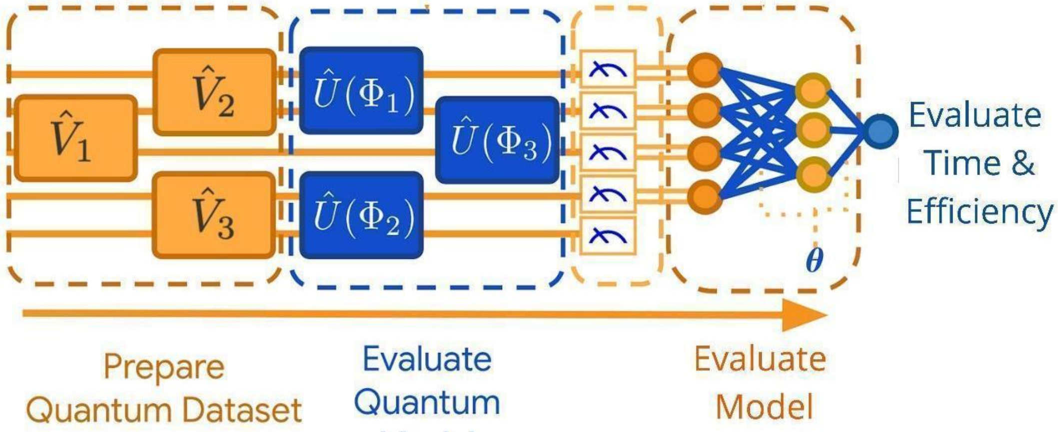


Figure 14:Proposed Model Architecture

3.5 Model Training

The model training took place on the IBM Quantum Lab platform, harnessing the capabilities of quantum computing for machine learning tasks. Prior to training, data preprocessing was conducted using scikit-learn's Min Max Scaler to normalize both feature and label values. The preprocessed dataset was then divided into training, validation, and testing subsets utilizing the train_test_split function. The training and validation sets were converted into PyTorch tensors for compatibility with the neural network model. The quantum feature map was defined using the Hadamard Feature Map from the Qiskit library. This feature map applied a set of operators on the quantum circuit to check the encoded input data.

A quantum circuit was created with a specified number of qubits, and the defined feature map was appended to the circuit. The resulting quantum circuit was saved as an image. Subsequently, the PyTorch nn.Module class was employed to establish the neural network model. This architecture encompassed three fully connected layers, each employing distinct activation functions. Following experimentation, the ReLU activation function emerged as the most effective choice. This is after the analysis of Relu, Sigmoid and Hyperbolic Tangent activation functions. The model underwent training over a designated number of epochs, employing various optimizers. It was determined that the Adam optimizer yielded the best outcomes for the dataset. Stochastic Gradient Descent optimizer was also tested but yielded more loss. Table 3.3 and table 3.4 show the results of the experiments. To gauge performance, the mean squared error loss function was computed. Throughout the training process, the loss values for both training and validation, alongside the corresponding training and validation times, were meticulously documented for thorough analysis.

Finally, the training and validation loss curves were plotted providing insights into the training process. Similarly, the training and validation time curves were plotted and saved, showcasing the time taken for each iteration.

Table 6:Stochastic Gradient Descent Optimizer Vs ReLU, Sigmoid and Tanh Activation Functions

Epochs	ReL/ SDG Loss	ReLU/SDG Time(s)	Sigmoid/SDG Loss	Sigmoid/ SDG Time(s)	Tanh/SDG Loss	Tanh/SDG Time(s)
0	0.25	0.5	0.56	0.5	0.55	0.7
250	0.25	1.5	0.5	0.9	0.54	1.3
500	0.2	1.6	0.5	1.3	0.5	1.5
750	0.15	1.9	0.43	1.5	0.45	1.9
1000	0.15	2.3	0.4	1.7	0.44	2.4
1250	0.1	2.6	0.36	1.9	0.35	2.9
1500	0.1	2.9	0.35	2.4	0.25	3.2
1750	0.1	2.9	0.3	2.5	0.2	3.4
2000	0	3.4	0.25	2.7	0.15	3.7

Table 7: Adam Optimizer Vs ReLU, Sigmoid and Tanh Activation Functions

Epochs	ReLU/ Adam Loss	ReLU/Adam Time(s)	Sigmoid/ Adam Loss	Sigmoid/ Adam Time(s)	Tanh/ Adam Loss	Tanh/Adam Time(s)
0	0.25	0	1.2	0.7	1.3	0.9
250	0.25	0.5	1	0.9	1.2	1.2
500	0.2	0.5	0.7	1.2	0.95	1.4
750	0.15	1	0.6	1.4	0.8	1.7
1000	0.15	1	0.5	1.5	0.78	2.1
1250	0.1	1.5	0.4	1.9	0.6	1.3
1500	0.1	1.5	0.3	2.1	0.6	2.7
1750	0.05	2	0.3	2.6	0.5	2.9
2000	0	2.5	0.2	2.9	0.45	3

3.6 Model Development Tools

3.6.1 Software

Linux operating system was used for this study, the Long-Term Support version of the operating system. Linux is an open-source platform and also a lightweight operating system that comes pre-packaged with many machine learning enabled scripts (Naveen & Kounte, 2020). Python programming language was the preferred language of this study due to its extensive use in machine learning and its vast library in machine learning applications (Jammal & AbuSharkh, 2021).

Qiskit an open-source quantum computing framework, was also used. It enabled interaction with quantum circuits, algorithms, and simulations. It included a complete set of tools for quantum programming, including the creation, execution, and visualization of quantum circuits.

3.6.2 Hardware

The algorithm was developed using a personal computer with the following specification: 16 GB RAM, 2 TB Storage, 3.2 GHz processors and GPU capabilities.

3.6.3 Quantum Development Environment

A quantum-virtualized environment was created on the IBM QuantumLab for the simulations as well as a cloud quantum resource for training, validating and testing the data.

The research utilized the Jupyter Notebook on the IBM Quantum Lab platform, a publicly available web-based tool enabling the creation and dissemination of documents featuring executable code, equations, visualizations, and explanatory text. Moreover, this platform provided a range of resources for composing quantum programs, encompassing circuit and pulse-level designs, while also optimizing them to suit the limitations of specific physical quantum processors. Furthermore, it facilitated the efficient management of batch execution of experiments on remote-access backends.

The research also employed the IBM Quantum Composer for devising the quantum circuits essential for data modeling. This platform facilitated rapid and versatile experimentation as well as streamlined production processes by means of an intuitive user interface, distributed training capabilities, and a comprehensive collection of tools and libraries.

3.7 Model Evaluation

The model efficiency was evaluated using a variety of techniques. These techniques offered several viewpoints on the model forecast levels. The metrics used were:

3.7.1 Mean Absolute Error

This metric represents the mean absolute deviation between the projected and observed values. It is computed by averaging the absolute differences between the anticipated and realized figures.

$MAE = (1 / n) * \Sigma |y - \hat{y}|$…………………………………………………..Equation 1

where:

n is the number of data points

Σ represents the summation

y is the actual value

ŷ is the predicted value

3.7.2 Mean Squared Error

The mean squared error (MSE) is the mean of the squared differences between the predicted and observed values. It is determined by averaging the squared deviations.

$MSE = (1 / n) * \Sigma (y - \hat{y})^2$ ………………………………………………….Equation 2

where:

n is the number of data points

Σ represents the summation

y is the actual value

ŷ is the predicted value

3.7.3 Root Mean Squared Error

The square root of the mean squared error is a metric known as RMSE. It provides a measure of the typical magnitude of errors in the same units as the values.

$$RMSE = \sqrt{(MSE)} \text{ ………………………………………………………… Equation 3}$$

where:

RMSE represents the Root Mean Squared Error

MSE represents the Mean Squared Error

3.7.4 Mean Absolute Percentage Error

This is the average percentage difference between the projected and actual figures referred to as MAPE. The average of the absolute percentage differences is used to calculate it.

$$MAPE = (1 / n) * \Sigma |(y - \hat{y}) / y| * 100 \text{ ………………………………………Equation 4}$$

where:

n is the number of data points

Σ represents the summation

y is the actual value

$\hat{y}$ is the predicted value

3.7.5 R-Squared

The R-squared coefficient indicates how much of the variance in the data is accounted for by the model. It is calculated by dividing the square sum that was explained by the square sum overall.

$$R\text{^}2 = 1 - (SSR / SST) \text{ ……………………………………………………Equation 5}$$

where:

SSR, or the Sum of Squared Residuals-is the summation of squared differences between the predicted values and the mean of the observed values.

SST- represents the total sum of squared differences between the observed values (y) and the mean of the observed values ($\bar{y}$).

3.7.6 Forecast Bias

Forecast bias measures the model's systematic propensity to over or under-estimate study estimates. The average discrepancy between the expected and actual numbers is used to calculate it.

Forecast Bias = $(\Sigma(y - \hat{y})) / n$……………………………………………………Equation 6

where:

$\Sigma(y - \hat{y})$ is the sum of the differences between the observed values (y) and the predicted values ($\hat{y}$)

n is the total number of data points

These metrics provided valuable insights into the effectiveness of the model, facilitating a comprehensive evaluation of its performance. Ultimately, the findings were condensed into a comparative table for contrasting the performance of a quantum-enhanced neural network with a classical neural network.

3.8 Ethical Consideration

The focus and scope of ethical considerations have expanded and intensified in response to society's demand for increased responsibility. This research project obtained authorization from the Chuka University Ethics Committee and secured a research permit from the National Commission for Science, Technology and Innovation (NACOSTI). The research's integrity was maintained by appropriately crediting the contributions of others through proper referencing and citation.

CHAPTER FOUR

RESULTS AND DISCUSSION

4.1 Introduction

The evaluation of the Quantum-Enhanced Neural Network for forecasting economic growth model was carried out using a comparative table. The comparative table was used to evaluate the mean absolute error, mean squared error, root mean squared error, mean absolute percentage error, r-squared, forecast bias of the model. More so, a comparative time table was used to compare the time taken with classical model visa vee a quantum model. External consistency of the model was tested using Ugandan Dataset.

4.2 Mapping Classical Computer-Based Dataset to Quantum Dataset

The following steps were taken for quantum data processing and mapping

1. Feature selection: Analyzing the classical dataset to identify the relevant features to be used for the quantum representation. This step considered factors such as the significance of the features for the mapping process and the specific quantum algorithm or task aiming to perform. The table 8 shows the World Bank sample dataset.

Table 8: Classical World Bank Dataset Sample

Series Name	1960	1961	1962	1963	1964	1965	1966	1967
Population, total	7751435	8047470	8363578	8697200	9047387	9417207	9802605	10201068
Population growth (annual %)	0	3.74797688	3.852857	3.911481	3.947485	4.006256	4.010962	3.984425
Surface area (sq. km)	0	580370	580370	580370	580370	580370	580370	580370
Population density (people per sq. km of land area)	0	14.13970201	14.69512	15.2813	15.89659	16.54638	17.22354	17.92365
Poverty headcount ratio at national poverty lines (% of population)	0	0	0	0	0	0	0	0
Poverty headcount ratio at $2.15 a day (2017 PPP) (% of population)	0	0	0	0	0	0	0	0
GNI, Atlas method (current US$)	0	0	8.36E+08	9.25E+08	9.73E+08	1E+09	1.16E+09	1.21E+09
GNI per capita, Atlas method (current US$)	0	0	100	110	110	110	120	120
GNI, PPP (current international $)	0	0	0	0	0	0	0	0
GNI per capita, PPP (current international $)	0	0	0	0	0	0	0	0
Income share held by lowest 20%	0	0	0	0	0	0	0	0
Life expectancy at birth, total (years)	48.68	49.533	50.224	50.808	51.328	51.7	52.097	52.455
Fertility rate, total (births per woman)	7.632	7.718	7.803	7.864	7.92	8.028	8.056	8.046
Adolescent fertility rate (births per 1,000 women ages 15-19)	173.665	175.998	174.612	172.11	168.272	169.284	171.866	174.438
Contraceptive prevalence, any method (% of married women ages 15-49)	0	0	0	0	0	0	0	0
Births attended by skilled health staff (% of total)	0	0	0	0	0	0	0	0
Mortality rate, under-5 (per 1,000 live births)	197.2	189.1	182.1	176.2	171.1	166.9	163.3	160.1
Prevalence of underweight, weight for age (% of children under 5)	0	0	0	0	0	0	0	0
Immunization, measles (% of children ages 12-23 months)	0	0	0	0	0	0	0	0

2. Quantum representation selection: Choosing a suitable quantum representation based on the available quantum hardware and the specific requirements of the study. Common quantum representations include qubits or quantum states.
3. Dimensionality reduction: The classical dataset had high dimensionality. Principal component analysis for feature extraction of the data employed to select the most informative features for the quantum representation.
4. Quantum Circuit-based Feature Encoding: Developing a quantum circuit framework that corresponds to the chosen features extracted from the classical dataset. Mapping each feature to specific qubits or quantum states within the quantum circuit. Applying quantum gates, such as Hadamard (H) gates and CNOT (CX) gates, to encode the classical data onto the quantum states or qubits. This phase includes data encoding through compression into lower dimensions known as the bottleneck layer or code layer. This process ensures that the number of output units matches the number of input units, following the autoencoder principle (Ding *et al*., 2019). The process is illustrated in Figure 15.

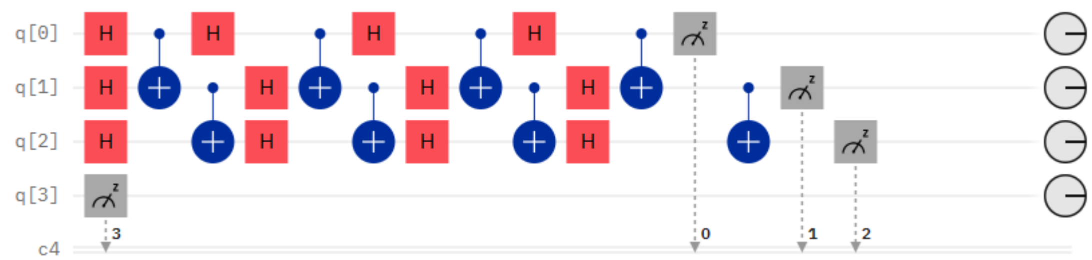


Figure 15:Quantum Circuit Encoding Classical Data to Quantum States

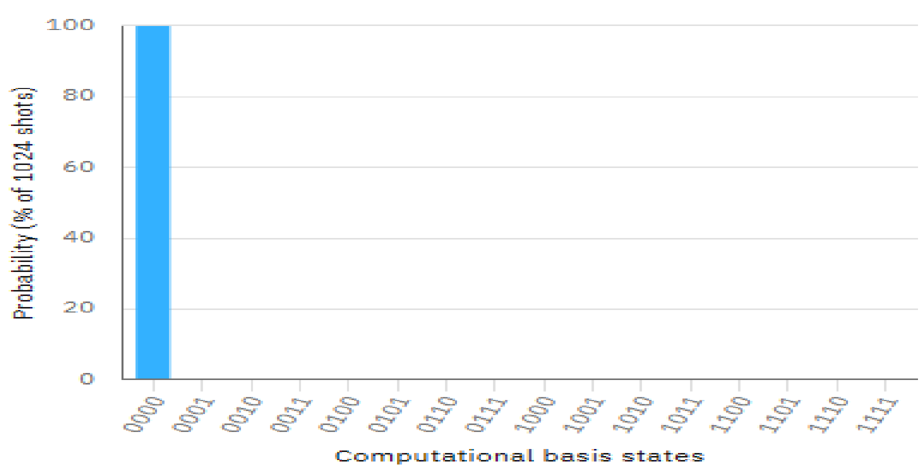


Figure 16: Quantum Probabilities Vs Computational Basis States for Quantum Data Mapping

A "0000" at 100% indicates that the quantum system is in the "zero state," a particular quantum state with a probability amplitude of 1. In other words, there is no doubt whatsoever that the system is in the state |0000]. The figure 4.2 illustrates the quantum probabilities.

The state |0000 in a four-qubit system denotes that all four qubits are in the "zero" state. To illustrate this situation, the Q-Shere in figure 17 displays one point on its surface directly at the sphere's top. This is thus because the system has a 100% chance of being in the |0000 state, as shown by the probability amplitude of this state, which is 1.

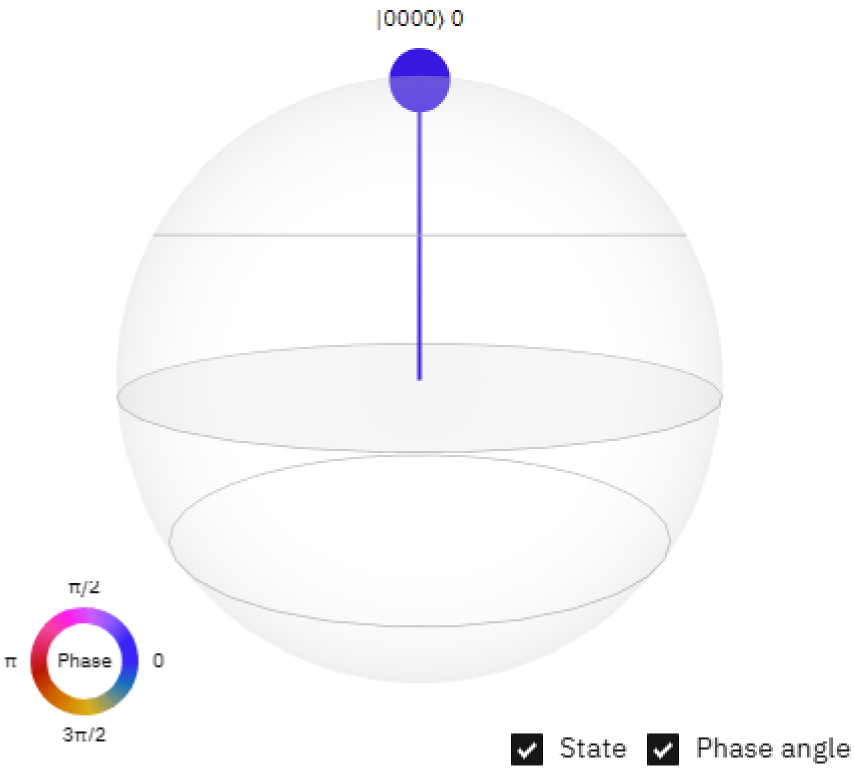


Figure 17: Quantum Q-sphere for the Circuit Quantum Mapping Circuit

The Q-Sphere illustrates the distribution of a quantum circuit across multiple qubits within a quantum environment. It provides a visual depiction of the relationship between the logical qubits in the quantum circuit and the physical qubits on the quantum device. This representation visually conveys the assignment or allocation of logical qubits onto the physical qubits of the selected quantum device. Figure 17 visually presents the Q-Sphere depicting the mapped data.

This mapping is necessary because different connectivity restrictions may apply to physical qubits on a quantum device, and not all qubits can be connected to one another directly. Users can better comprehend how the quantum circuit will be implemented on a given piece of quantum hardware by using this visualization, which takes the device's topology and available qubit connectivity into consideration.

Users can determine whether their quantum circuit can be successfully implemented on the chosen quantum device without breaching the connection limits by seeing the Q-Shere. It also aids in locating potential snags or regions where qubit usage may be high, which may affect the circuit's integrity and performance. The figure 18 shows the computational basis against the amplitudes vector for the mapping.

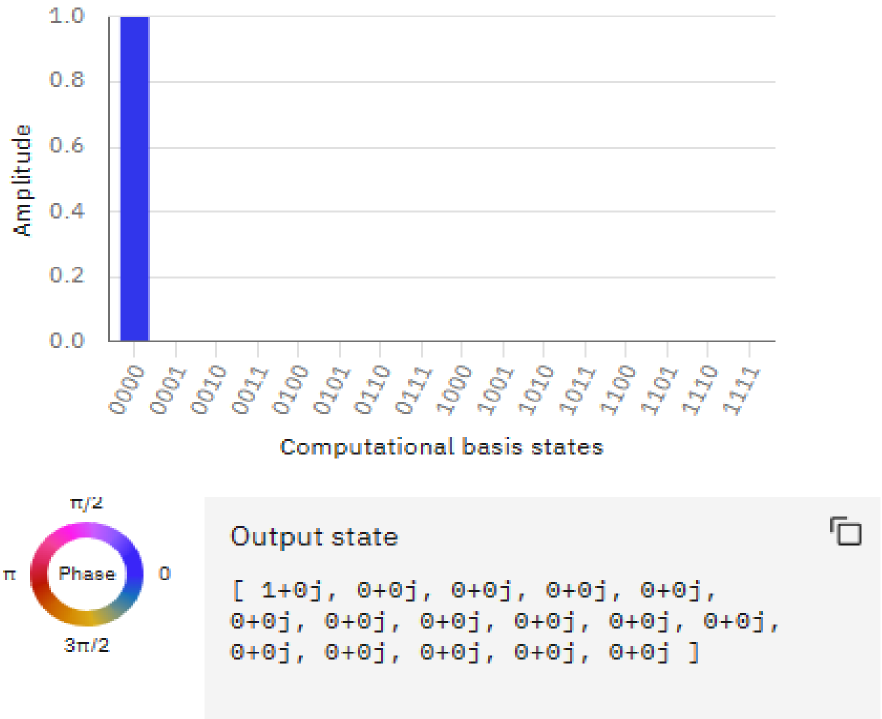


Figure 18: Computational Basis vs Amplitudes State Vector Output

5. Quantum dataset generation: Creating a quantum dataset generated by a set of quantum states or qubits that represent the encoded classical features. The resulting quantum dataset was in the form of a state vector and mapped reduced components data in quantum state for the chosen quantum representation and the specific quantum neural network algorithm. The figure 19 shows Encoded mapped vector data. The figure 20 illustrates the resulting reduced mapped data

in quantum states. The figure 21 shows the quantum state diagram of the reduced components.

```
(1.000000000000000000e+00+0.000000000000000000e+00j)
(0.000000000000000000e+00+0.000000000000000000e+00j)
(0.000000000000000000e+00+0.000000000000000000e+00j)
(0.000000000000000000e+00+0.000000000000000000e+00j)
(0.000000000000000000e+00+0.000000000000000000e+00j)
(0.000000000000000000e+00+0.000000000000000000e+00j)
(0.000000000000000000e+00+0.000000000000000000e+00j)
(0.000000000000000000e+00+0.000000000000000000e+00j)
(0.000000000000000000e+00+0.000000000000000000e+00j)
(0.000000000000000000e+00+0.000000000000000000e+00j)
(0.000000000000000000e+00+0.000000000000000000e+00j)
(0.000000000000000000e+00+0.000000000000000000e+00j)
(0.000000000000000000e+00+0.000000000000000000e+00j)
(0.000000000000000000e+00+0.000000000000000000e+00j)
(0.000000000000000000e+00+0.000000000000000000e+00j)
(0.000000000000000000e+00+0.000000000000000000e+00j)
```

Figure 19: Encoded Mapped Quantum Vector Data

```
Reduced Components:
[-2.09483266e+10 -2.09466326e+10 -2.08714806e+10 -2.08130027e+10
 -2.07408327e+10 -2.07416727e+10 -2.05750724e+10 -2.05070325e+10
 -2.03862966e+10 -2.02812126e+10 -2.01361447e+10 -1.99612007e+10
 -1.96323129e+10 -1.92305936e+10 -1.87696498e+10 -1.84802471e+10
 -1.82650496e+10 -1.72452132e+10 -1.64358571e+10 -1.55052011e+10
 -1.44742767e+10 -1.48851006e+10 -1.53080127e+10 -1.57603936e+10
 -1.55481550e+10 -1.56045577e+10 -1.45004653e+10 -1.37687715e+10
 -1.33842111e+10 -1.34564774e+10 -1.31672329e+10 -1.35881130e+10
 -1.35304629e+10 -1.59878021e+10 -1.45914467e+10 -1.26932660e+10
 -9.69373359e+09 -8.62381829e+09 -7.64559318e+09 -8.84357845e+09
 -9.03423492e+09 -8.75358460e+09 -8.59184812e+09 -6.83507438e+09
 -5.64425493e+09 -3.00169428e+09  4.08593279e+09  1.02186032e+10
  1.41555613e+10  2.06076259e+10  2.36659955e+10  2.51298653e+10
  3.46571140e+10  3.99318333e+10  4.65461765e+10  4.83808213e+10
  5.30755293e+10  6.02962088e+10  7.04633643e+10  7.86401217e+10
  7.89269506e+10  8.86074875e+10 -3.81469727e-06]
```

Figure 20: Mapped Reduced Components in Quantum State Data

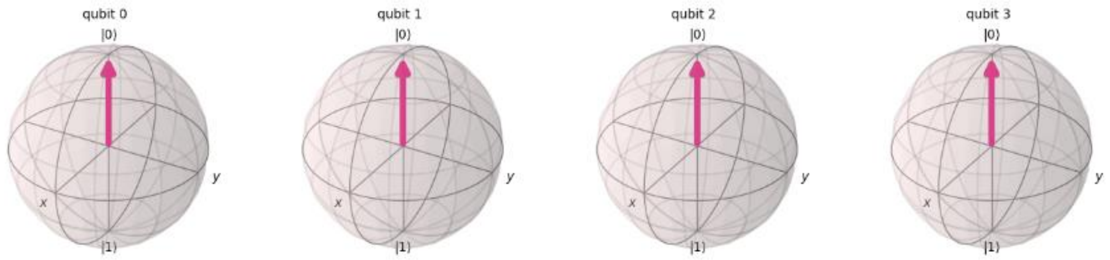


Figure 21: Quantum State Diagram of the Reduced Components

6. Data validation and verification: Validation of the quantum dataset to ensure it accurately represents the classical dataset after the encoding process. This was done by comparing statistical properties and relevant metrics between the classical and quantum datasets to verify the correctness of the mapping. The figure 22 shows the resulting quantum circuit with the weighted inputs.

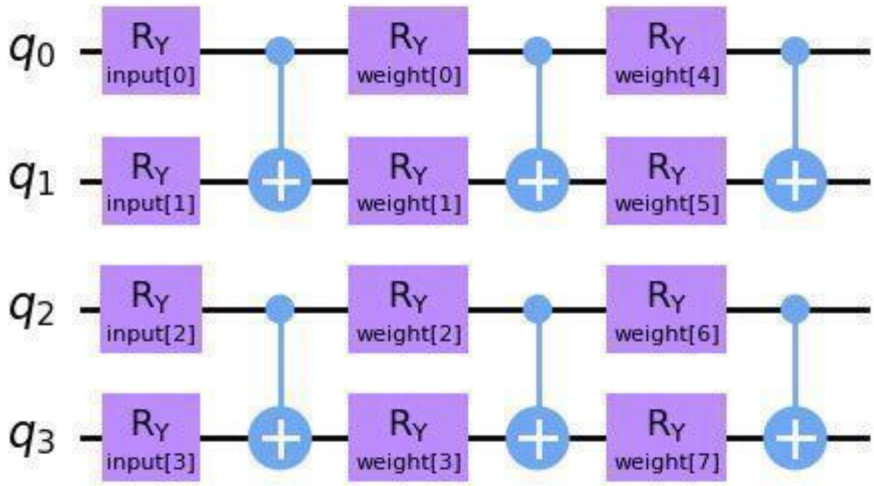


Figure 22: Resulting Quantum Circuit with Weights and Inputs

4.3 Quantum Enhanced Neural Network Model Summary

The model summary in Figure 23 shows the layers, output shape, and number of parameters of the quantum-enhanced neural network model. It also shows the diagraph architecture in Figure 24 and how many trainable parameters the model had. The Figure 25 illustrates the model architecture.

```
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            Linear-1                   [-1, 64]             128
              ReLU-2                   [-1, 64]               0
            Linear-3                   [-1, 32]           2,080
              ReLU-4                   [-1, 32]               0
            Linear-5                    [-1, 1]              33
================================================================
Total params: 2,241
Trainable params: 2,241
Non-trainable params: 0
----------------------------------------------------------------
Input size (MB): 0.00
Forward/backward pass size (MB): 0.00
Params size (MB): 0.01
Estimated Total Size (MB): 0.01
----------------------------------------------------------------
```

Figure 23: Model Summary

```
digraph {
	graph [size="12,12"]
	node [align=left fontname=monospace fontsize=10 height=0.2 ranksep=0.1 shape=box style=filled]
	140103763877120 [label="
 (1, 1)" fillcolor=darkolivegreen1]
	140103763828096 [label=AddmmBackward0]
	140103763827904 -> 140103763828096
	140103772241680 [label="fc3.bias
 (1)" fillcolor=lightblue]
	140103772241680 -> 140103763827904
	140103763827904 [label=AccumulateGrad]
	140103764411056 -> 140103763828096
	140103764411056 [label=ReluBackward0]
	140103764415376 -> 140103764411056
	140103764415376 [label=AddmmBackward0]
	140103764418256 -> 140103764415376
	140103772027888 [label="fc2.bias
 (32)" fillcolor=lightblue]
	140103772027888 -> 140103764418256
	140103764418256 [label=AccumulateGrad]
	140103764415952 -> 140103764415376
	140103764415952 [label=ReluBackward0]
	140103764417824 -> 140103764415952
	140103764417824 [label=AddmmBackward0]
	140103764418832 -> 140103764417824
	140103772023408 [label="fc1.bias
 (64)" fillcolor=lightblue]
	140103772023408 -> 140103764418832
	140103764418832 [label=AccumulateGrad]
	140103764416192 -> 140103764417824
	140103764416192 [label=TBackward0]
```

Figure 24: Model Digraph Architecture

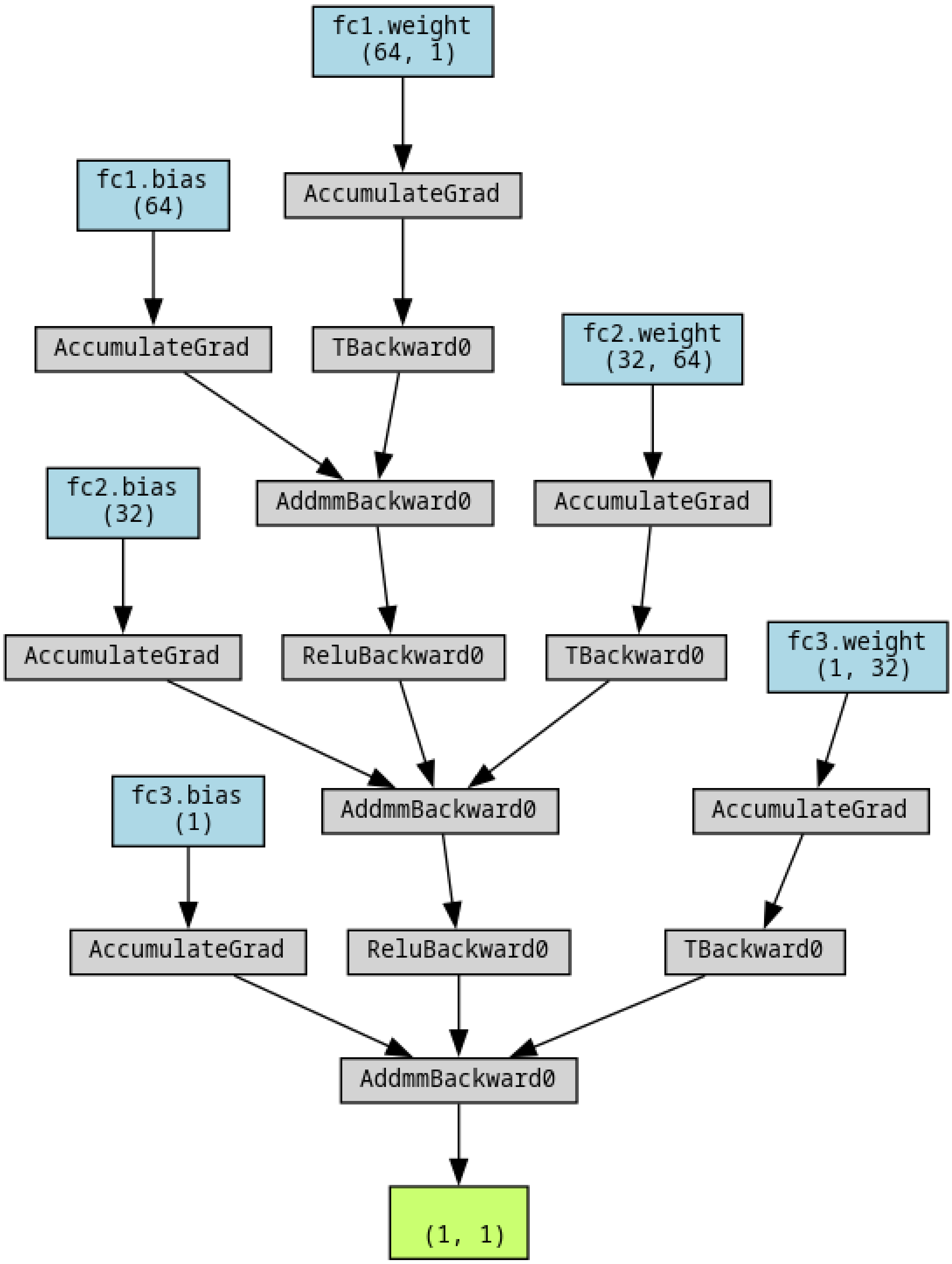


Figure 25: Model Architecture

4.3.1 Training and Validation Loss

The validation loss in Figure 26 shows how well the model fits new data, while the training loss shows how well it matches training data. An Adam optimizer was used to produce the training and validation loss, as illustrated in Figure 26 as well.

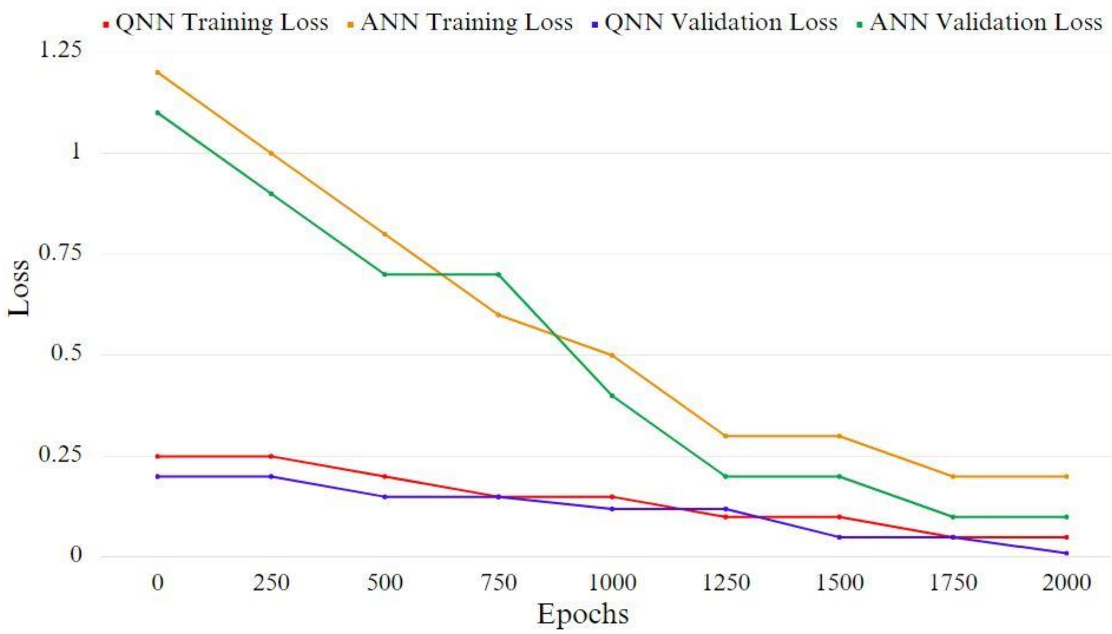


Figure 26: QNN vs ANN Model Training and Validation Loss

The training loss was significantly worse in the ANN model than the validation loss as compared to the QNN model. This indicated that the training dataset is more difficult to predict than the validation dataset. This was explained by the well accepted fact that the validation data percentage is lower than that of the training set.

The QNN model also performed incredibly better than the ANN model as shown with the smoother curve presented in the analysis. The ANN model data frames loading caused higher spikes at the beginning of the training curve. The spikes at the beginning of the curve were evident as the model read data at the start of the learning process. The spikes eventually flattened down after a few epochs, and the model carried on smoothly generalizing.

4.3.2 Training and Validation Loss Over Time

The training and validation time for both models were measured, assessed and presented in a graphical representation. The Figure 27 shows the training and validation loss over time.

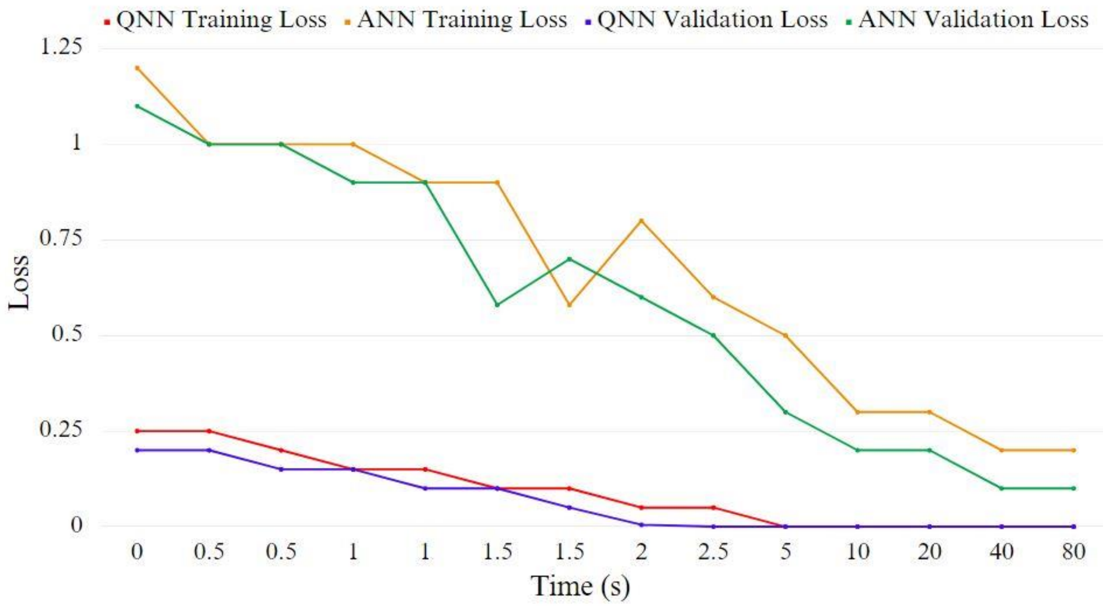


Figure 27: QNN Vs ANN Training and Validation Loss Over

The analysis of the results showed that the QNN model performed very well as compared to the ANN one in terms of time. The QNN model average time taken for the experiment was 1.9 seconds as compared to the ANN time of 83.3 seconds. This represents a remarkable improvement in efficiency, with the QNN model exhibiting a reduction of approximately 97.7% in computational time.

4.4 Model Evaluation

The model was evaluated against Time, MAE, MSE, RMSE, MAPE, R-Squared and Forecast Bias.

Table 9: Quantum Neural Network Research Parameters

Quantum Artificial Neural Network Algorithm								
Iteration	**Training Time**	**Validation Time**	**MAE**	**MSE**	**RMSE**	**MAPE**	**R^2**	**Forecast Bias**
Iteration 1	2.0367s	0.4323s	0.0117	0.0003	0.0191	3.1816	0.9971	-0.0111
Iteration 2	1.8266s	0.3689s	0.0098	0.0003	0.0168	2.18123	0.9977	-0.0085
Iteration 3	1.7848s	0.3621s	0.0101	0.0003	0.0169	3.7773	0.9978	-0.0098
Iteration 4	1.9004s	0.3876s	0.0117	0.0003	0.0173	4.4536	0.9976	-0.0109
Iteration 5	1.9236s	0.3959s	0.0089	0.0002	0.0148	1.7963	0.9982	-0.0082
Iteration 6	1.9687s	0.4002s	0.0113	0.0003	0.0173	2.1790	0.9976	-0.0109
Iteration 7	1.9907s	0.4078s	0.0108	0.0001	0.0121	2.3964	0.9988	0.0080
Iteration 8	2.0641s	0.4142s	0.0081	0.0002	0.0157	8.9718	0.9980	-0.0078
Iteration 9	1.8288s	0.3775s	0.0091	0.0002	0.0164	4.8054	0.9978	-0.0090
Iteration 10	1.9246s	0.3856s	0.0132	0.0003	0.0196	2.2389	0.9969	-0.0132
	Average Training Time	**Average Validation Time**	**Average MAE**	**Average MSE**	**Average RMSE**	**Average MAPE**	**Average R^2**	**Average Forecast Bias**
	1.9249	**0.3932**	**0.0105**	**0.0003**	**0.0166**	**3.5982**	**0.9978**	**-0.0081**

Table 10:Neural Network Research Parameters

Artificial Neural Network Algorithm								
Iteration	**Training Time**	**Validation Time**	**MAE**	**MSE**	**RMSE**	**MAPE**	**R^2**	**Forecast Bias**
Iteration 1	83.1286s	6.0612s	30.1997	29.2926	35.9572	4.9668	0.2882	-17.0093
Iteration 2	81.3818s	6.0691s	29.8336	12.1247	34.8206	5.6424	0.2081	-14.45486
Iteration 3	81.7724s	6.0629s	29.5869	11.6537	34.1376	6.1026	0.1611	-12.7179
Iteration 4	82.3566s	6.0672s	29.9437	12.3527	35.1464	5.4396	0.2307	-15.2219
Iteration 5	87.1612s	6.0747s	29.5169	11.5332	33.9606	6.2286	0.1491	-12.2394
Iteration 6	84.4122s	6.0655s	30.0017	12.4764	35.3221	5.3344	0.2431	-15.6211
Iteration 7	82.5177s	6.0577s	29.9977	12.4699	35.3127	5.3386	0.2424	-15.6028
Iteration 8	82.9472s	6.0566s	29.9122	12.2856	35.0509	5.4987	0.2241	-14.9991
Iteration 9	84.0367s	6.0559s	30.0447	12.5754	35.4618	5.2491	0.2529	-15.9394
Iteration 10	84.0367s	6.0559s	30.0447	12.5754	35.4618	5.2491	0.2529	-15.9394
	Average Training Time	**Average Validation Time**	**Average MAE**	**Average MSE**	**Average RMSE**	**Average MAPE**	**Average R^2**	**Average Forecast Bias**
	83.37511	**6.0627**	**29.9081**	**13.9339**	**35.0631**	**5.5049**	**0.2253**	**-14.9745**

4.4.1 Mean Absolute Error

This is the average absolute difference between the expected and actual values. A lower MAE value indicates greater performance and gives a clue as to the model's correctness.

The QNN model exhibited substantially lower MAE values than the ANN model, as can be seen by comparing the MAE values in Figure 28. This indicates that the QNN model had less errors between the predicted and actual values and performed more accurately.

As a result, the QNN model was deemed superior to the ANN model based on the findings of the MAE as shown in Figure 28 and Figure 29.

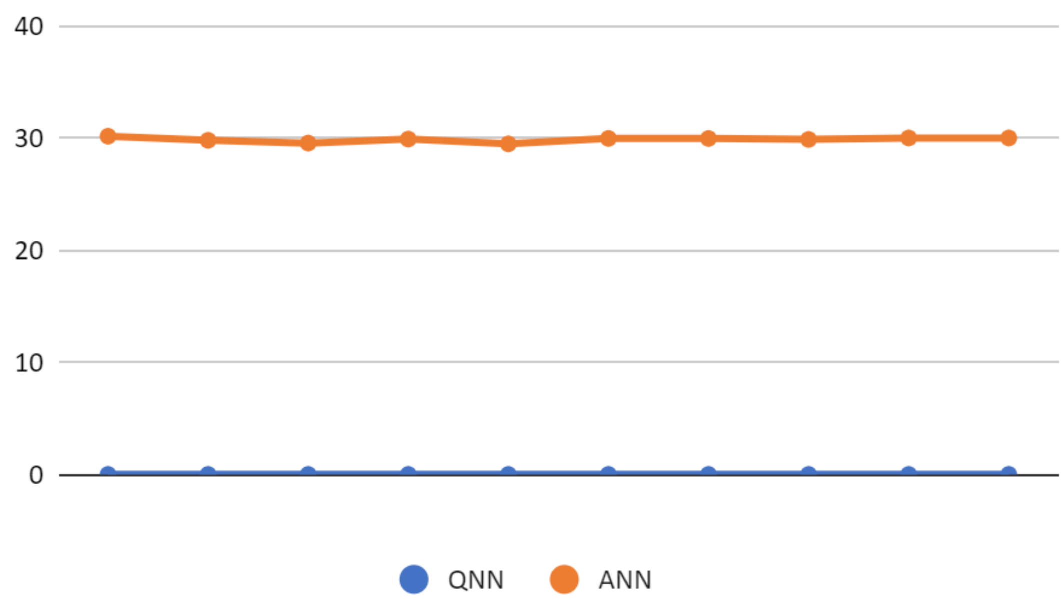


Figure 28: QNN vs ANN Mean Absolute Error

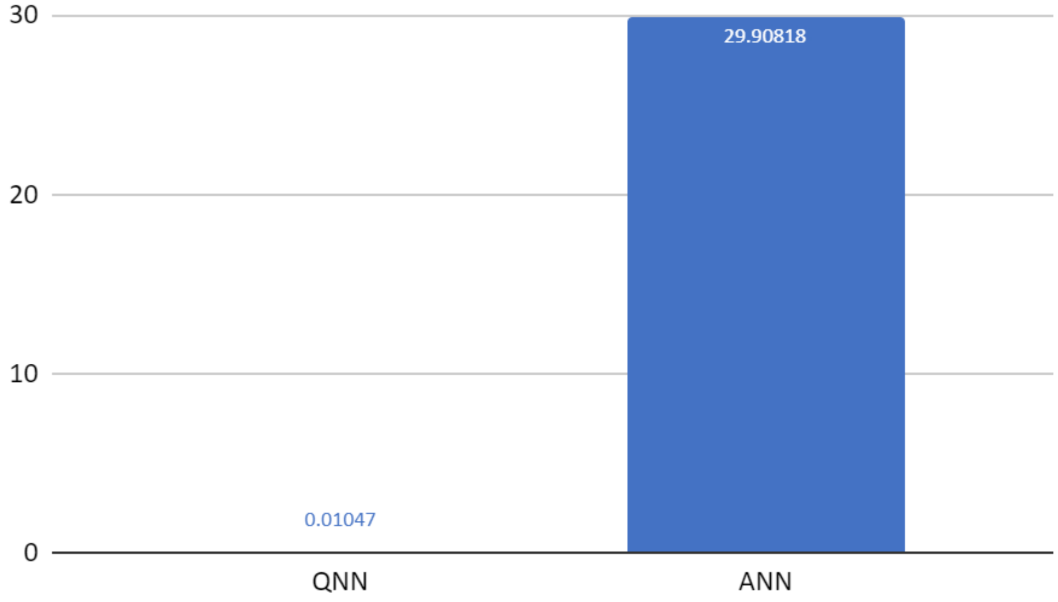


Figure 29: Average QNN vs ANN Mean Absolute Error

4.4.2 Mean Squared Error

The average squared difference between predicted and actual values is measured by the Mean Squared Error. It is employed to assess a regression model's precision.

The QML model MSE findings had significantly smaller values, ranging from 0.0001 to 0.0003, which indicated a smaller average squared difference between the predicted and actual values, as it was observed by comparing the MSE results. The ANN model however, contained larger values with a range of 11.6537 to 29.2926, showing a greater average squared difference.

The QNN model's MSE average value was regarded as superior because it showed a reduced error and a closer match between the predicted and actual values. A better fit of the model to the data was shown by lower MSE values, which suggested higher accuracy and precision in the predictions. The Figure 30 and 31 depicts this picture.

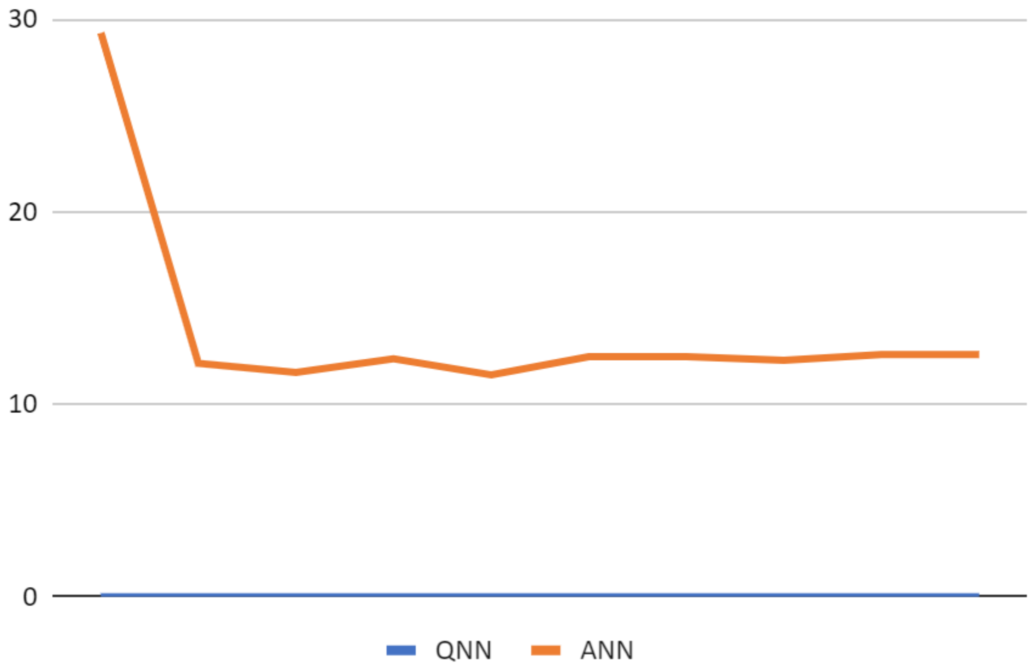


Figure 30: QNN vs ANN Mean Squared Error

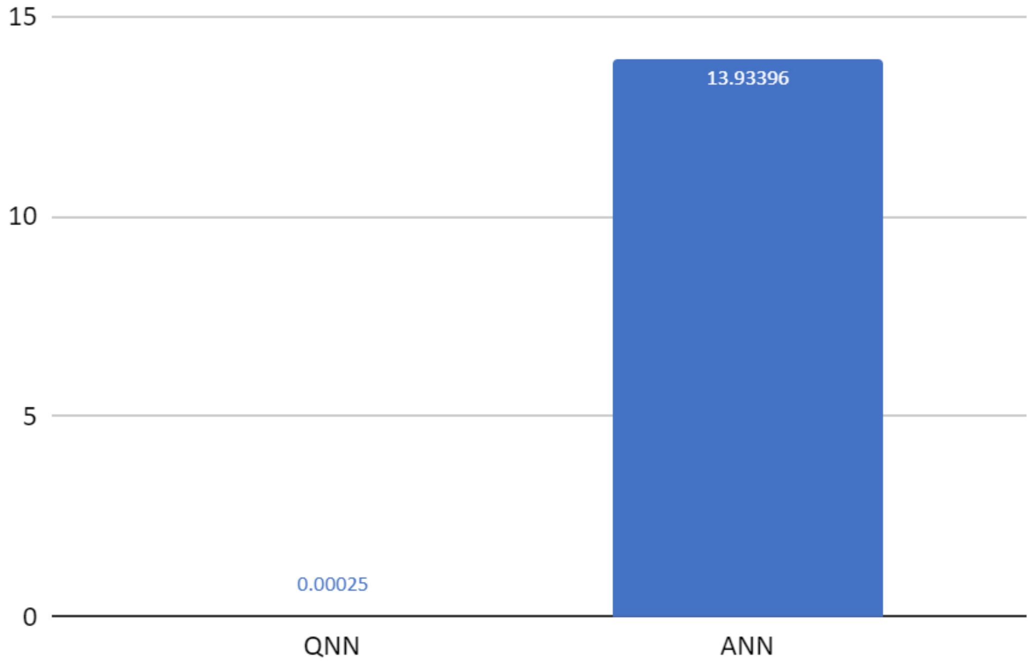


Figure 31: Average QNN vs ANN Mean Squared Error

4.4.3 Root Mean Squared Error

The average difference between a dataset's actual values and anticipated values is measured by the RMSE (Root Mean Squared Error). A lower RMSE reflects smaller errors between the anticipated and real values, indicating greater model performance. The magnitudes of the values in the QNN model were noticeably smaller than those in the ANN model. The decreased RMSE values in Figure 32 in QNN suggested greater performance of the target variable or the forecasted values in the set. The predicted values are also seen to be in a similar range as the training values. The Figure 33 shows the average RMSE.

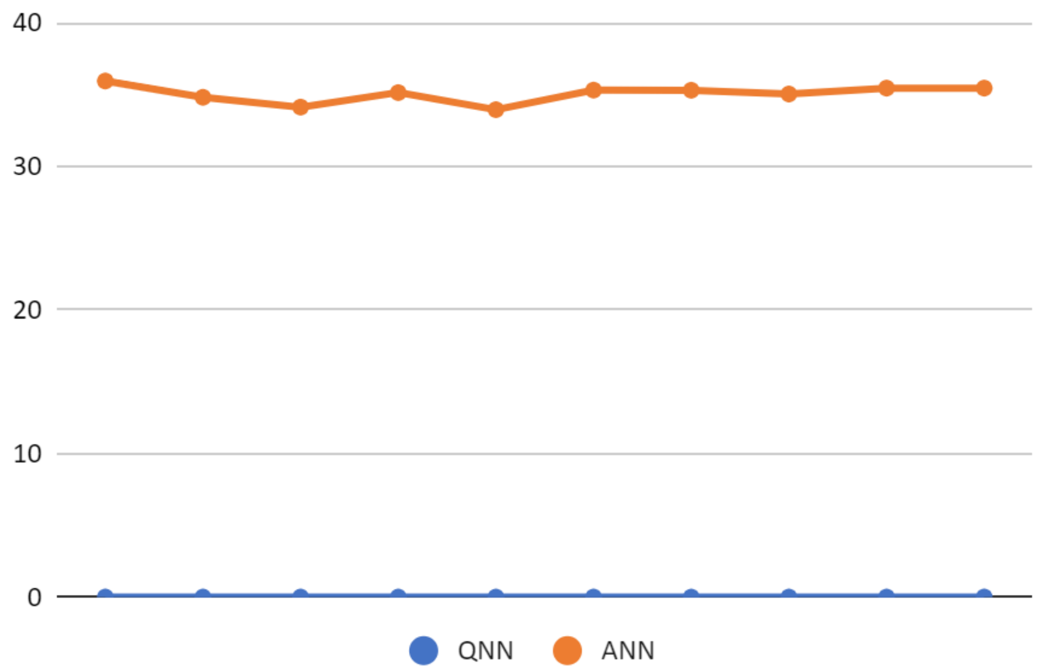


Figure 32: QNN vs ANN Root Mean Squared Error

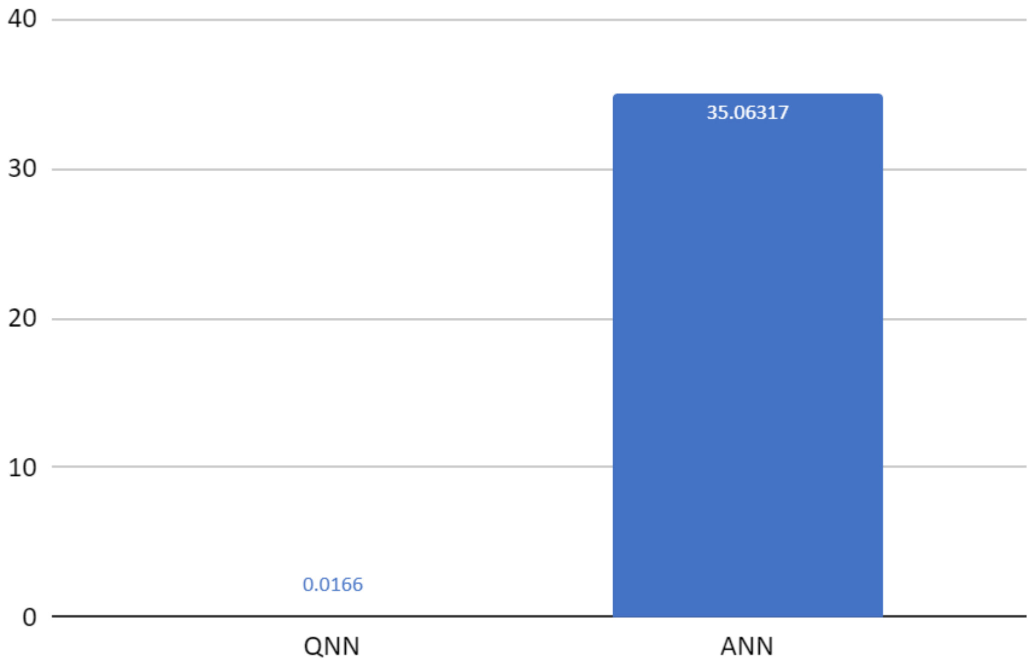


Figure 33: Average QNN vs ANN Root Mean Squared Error

4.4.4 Mean Absolute Percentage Error

The average percentage difference between the anticipated and actual values is calculated using the MAPE metric, which is used to assess the forecasting model's accuracy.

Because it displays a smaller percentage gap between the projected and actual values, a lower MAPE score suggests more accuracy. In the study, the QNN MAPE findings showed superiority to the ANN model's MAPE results as shown in Figure 34. The Figure 35 shows the average MAPE.

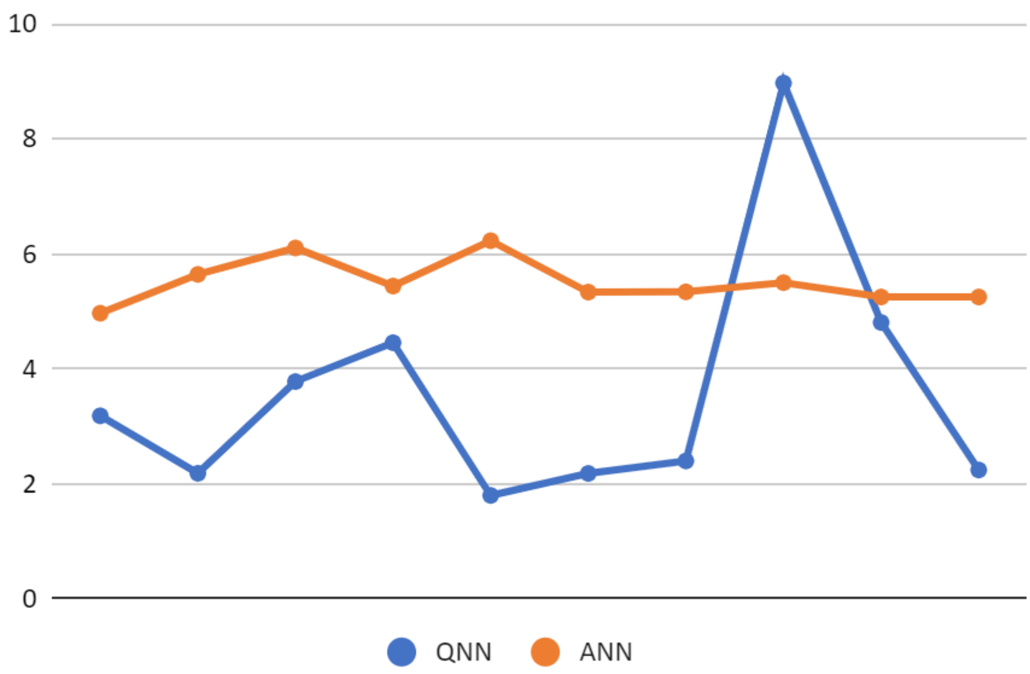


Figure 34: QNN vs ANN Mean Absolute Percentage Error

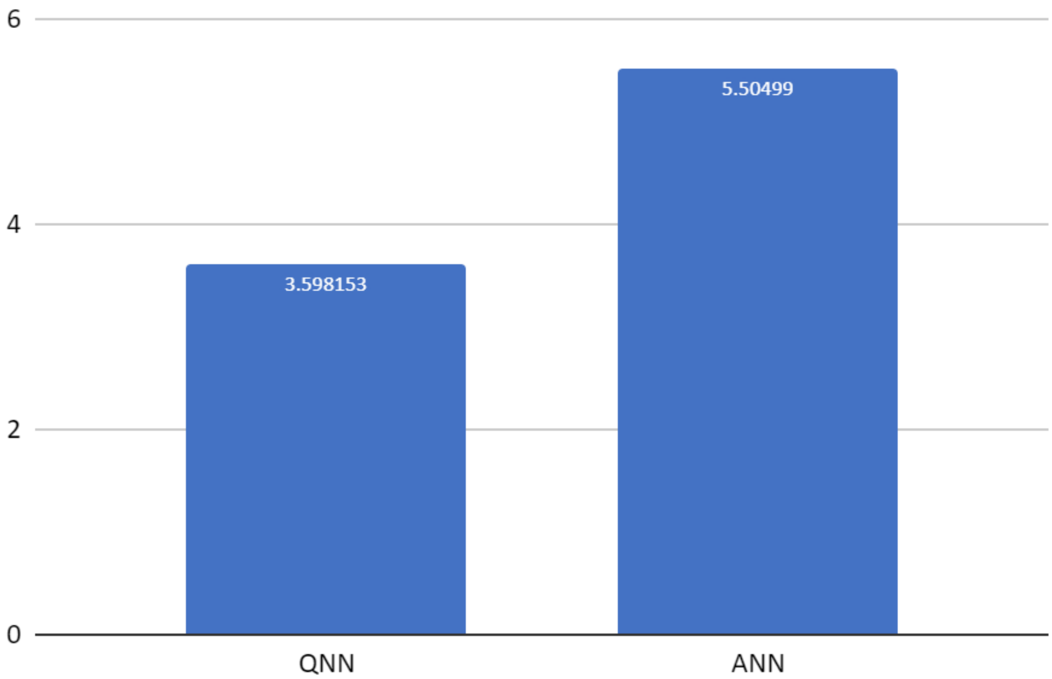


Figure 35: Average QNN vs ANN Mean Absolute Percentage Error

4.4.5 R-squared

A statistical measure, the R-Squared coefficient, shows how much of the variance in the dependent variable can be predicted from the independent variable or variables. It indicates how well a regression model fits the data and ranges from 0 to 1. A value that is nearer 1 denotes a better match between the model and the data.

The QNN model has much larger R-Squared values than the ANN model results as seen in Figure 36. The QNN result values fall between 0.9969 and 0.9988, demonstrating a very strong fit between the regression model and the data. The R-Squared values of the ANN model, on the other hand, are negative and vary from 0.2882 to 0.1491, which indicates that the model does not adequately match the data. The Figure 37 shows the average R-squared.

Because the first set of data shows a higher level of explanatory power and a better fit of the regression model to the data than the second set, it is therefore seen as better in terms of the R-Squared results.

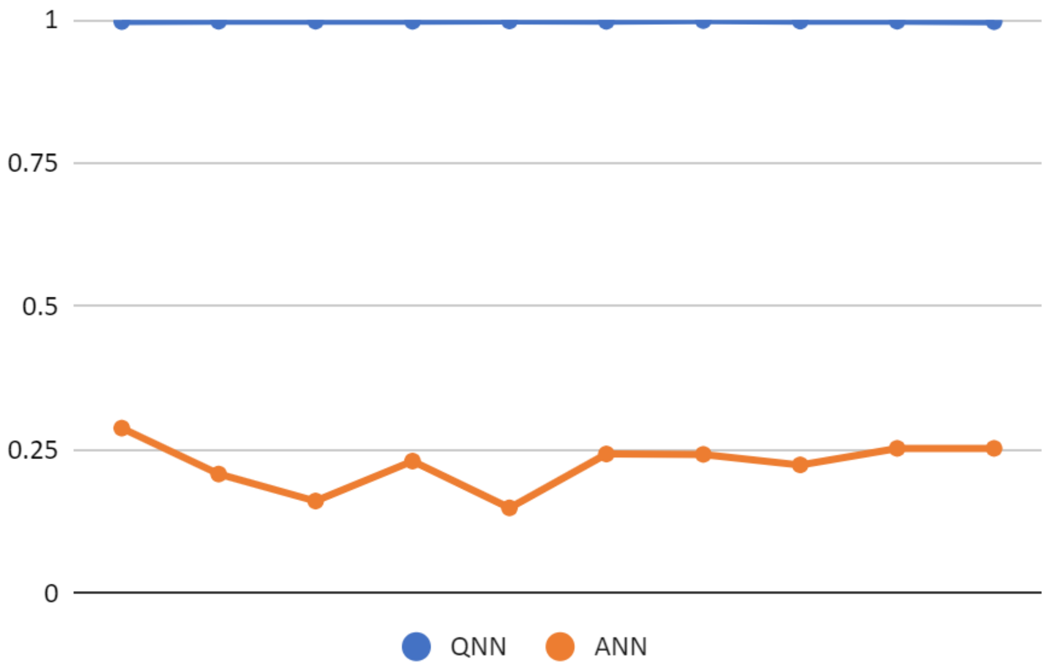


Figure 36: QNN vs ANN R-squared

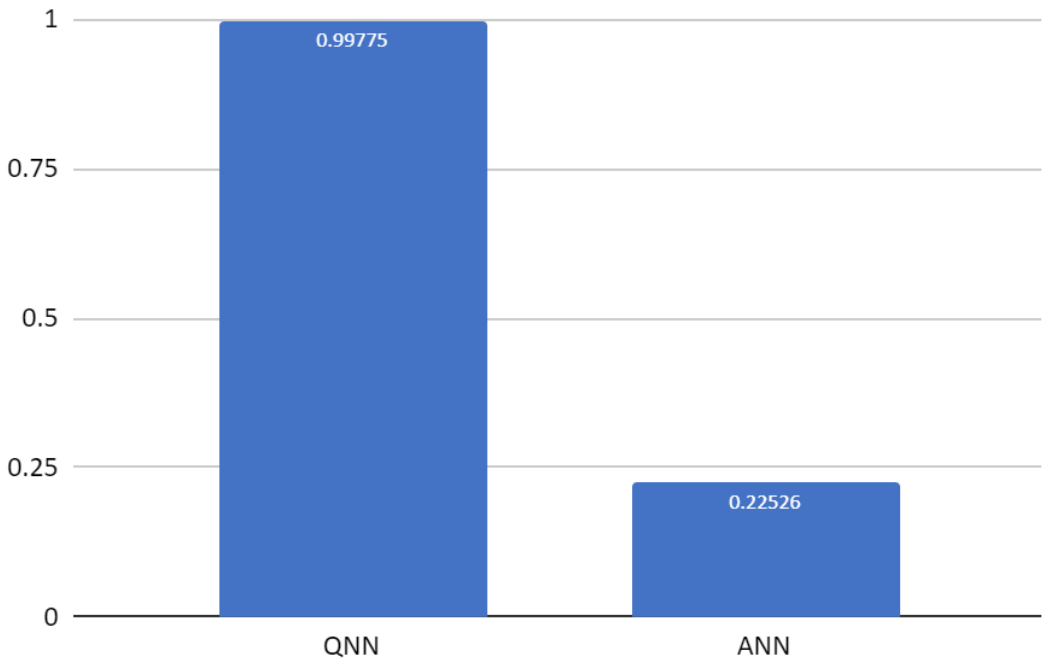


Figure 37: Average QNN vs ANN R-squared

4.4.6 Forecast Bias

The Forecast Bias is a measurement that shows the typical difference between forecasted values and actual values. The Figure 38 offers details on the forecasts' systematic overestimation or underestimation.

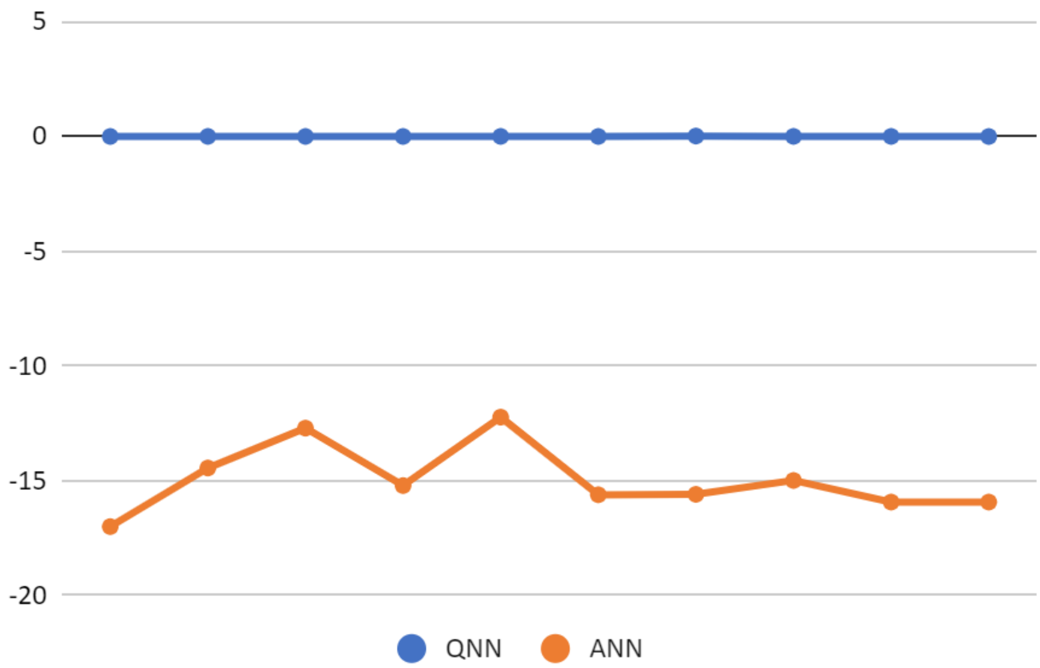


Figure 38: QNN vs ANN Forecast Bias

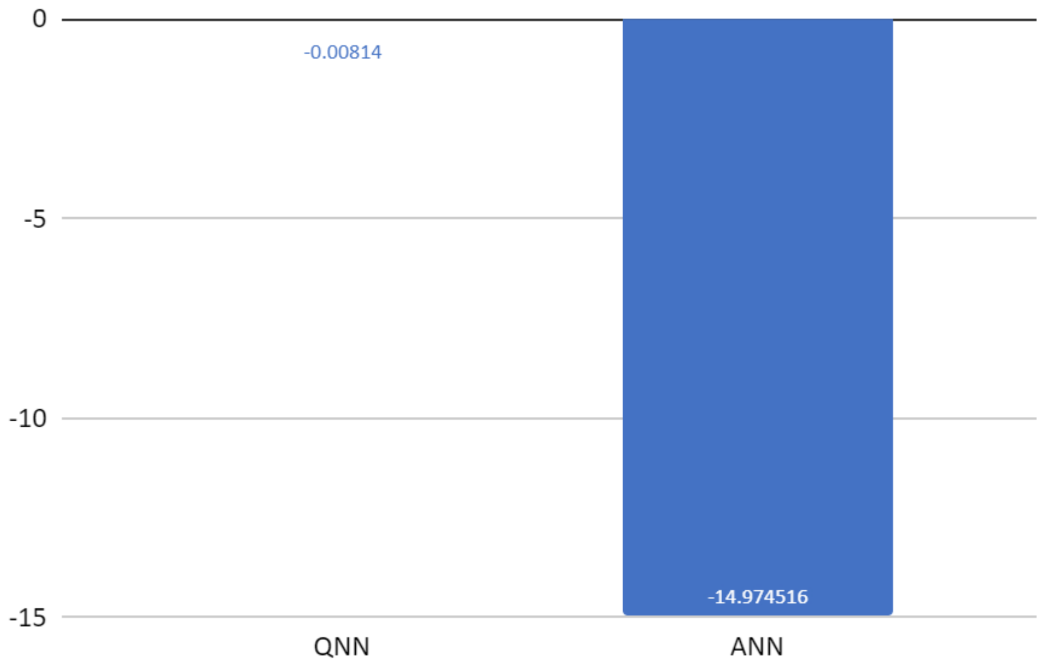


Figure 39: Average QNN vs ANN Forecast Bias

The QNN average forecast bias is negative (-0.00814), indicating that forecasts are generally underestimated as shown in Figure 39. In other words, the anticipated values are slightly lower than the actual ones. The ANN average forecast bias is also negative (-14.9745), which indicates that the projections are generally underestimated. The magnitude of the forecast bias in the ANN model is greater than in the QNN model, indicating a greater difference between the projected and actual values.

The QNN model appears to have a lesser forecast bias than the ANN model based only on the average prediction bias. As a result, the QNN model can be thought of as having less forecast bias because it deviates less from the actual values. The table 11 shows the forecasted GDP.

Table 11: Forecast Data from the Models

Year	GDP	QNN Predicted GDP	ANN Predicted GDP
1961	0.7929595	0.760281861	0.681540528
1973	2.5089985	2.461332083	2.065501543
1992	8.2091292	8.451481819	6.765370828
1999	12.896014	12.87607354	11.07848642
2008	35.895153	35.87516743	20.92065333
2016	74.815121	75.727211	59.84062131
2020	100.66654	101.110733	95.6920427

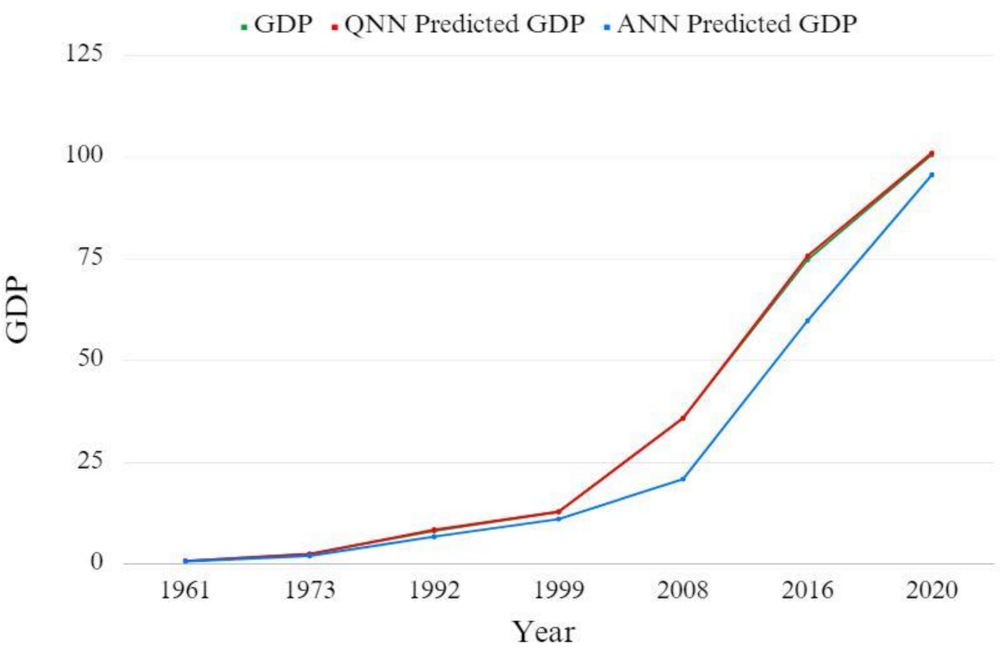


Figure 40: QNN vs ANN Forecast Graph

4.5 External Validity

The external consistency of the results was tested by training, validating and testing both the QNN model and ANN model with Uganda's dataset. Uganda was chosen for its economical and neighboring characteristics with Kenya. The validation loss and training time of the models was recorded and visualized using learning curves. Results were also recorded for MAE, MSE, RMSE, MAPE R-squared and Forecast Bias scores of the models.

Table 12: Quantum Neural Network Research Parameters

Quantum Artificial Neural Network Algorithm (External Validation)								
Iteration	**Training Time**	**Validation Time**	**MAE**	**MSE**	**RMSE**	**MAPE**	**R^2**	**Forecast Bias**
Iteration 1	1.5661s	0.3170s	0.0214	0.0011	0.0337	2.9933	0.9914	-0.0126
Iteration 2	2.2503s	0.4488s	0.0193	0.0011	0.0338	2.1507	0.9914	-0.0165
Iteration 3	1.9207s	0.3863s	0.0218	0.0012	0.0357	2.6995	0.9904	-0.0168
Iteration 4	2.3931s	0.4449s	0.0199	0.0011	0.0325	4.7661	0.9921	-0.0167
Iteration 5	1.9513s	0.3826s	0.0217	0.0012	0.0339	2.3771	0.9913	-0.0212
Iteration 6	1.8537s	0.3697s	0.0238	0.0013	0.0358	2.7034	0.9903	-0.0227
Iteration 7	1.8683s	0.3710s	0.0242	0.0013	0.0367	4.6881	0.9898	-0.0237
Iteration 8	1.7689s	0.3585s	0.0212	0.0012	0.0352	2.9311	0.9906	-0.0162
Iteration 9	2.0234s	0.3993s	0.0240	0.0013	0.0356	2.8023	0.9904	-0.0240
Iteration 10	1.7559s	0.3537s	0.0150	0.0009	0.0237	2.3789	0.9957	-0.0132
	Average Training Time	**Average Validation Time**	**Average MAE**	**Average MSE**	**Average RMSE**	**Average MAPE**	**Average R^2**	**Average Forecast Bias**
	1.9249	**0.3832**	**0.0212**	**0.0012**	**0.0337**	**3.04905**	**0.9913**	**-0.0184**

Table 13: Neural Network Research Parameters

Artificial Neural Network Algorithm (External validation)								
Iteration	**Training Time**	**Validation Time**	**MAE**	**MSE**	**RMSE**	**MAPE**	**R^2**	**Forecast Bias**
Iteration 1	83.3107s	6.0554s	12.3705	19.4851	13.9589	3.9853	0.2191	-5.9171
Iteration 2	87.9861s	6.0715s	12.5130	20.7650	14.4101	3.5716	0.2991	-6.9144
Iteration 3	86.6587s	6.0551s	12.4744	20.4007	14.2831	3.6826	0.2763	-6.6464
Iteration 4	84.7846s	6.0616s	12.4181	19.8934	14.1044	3.8454	0.2445	-6.2535
Iteration 5	87.8519s	6.0950s	12.2411	18.4966	13.6002	4.3599	0.1572	-5.0134
Iteration 6	84.0046s	6.0551s	12.4063	19.7884	14.0671	3.8811	0.2380	-6.1682
Iteration 7	87.7795s	6.1061s	12.3754	19.5282	13.9743	3.9697	0.2217	-5.9541
Iteration 8	85.2217s	6.0558s	12.4618	20.2853	14.2426	3.7186	0.2691	-6.5594
Iteration 9	82.4407s	6.0837s	12.4664	20.3255	14.2567	3.7063	0.2716	-6.5893
Iteration 10	82.3528s	6.0617s	12.4024	19.7531	14.0545	3.8935	0.2358	-6.1388
	Average Training Time	**Average Validation Time**	**Average MAE**	**Average MSE**	**Average RMSE**	**Average MAPE**	**Average R^2**	**Average Forecast Bias**
	85.2391	**6.0701**	**12.4129**	**19.8721**	**14.0951**	**3.8614**	**0.2432**	**-6.21545**

4.5.1 Training and Validation Loss

The training and validation loss results were visualized in a graph as represented in Figure 41. The validation loss was much better in the QNN model than the ANN model, which inferred that the model generalized well on the QNN model.

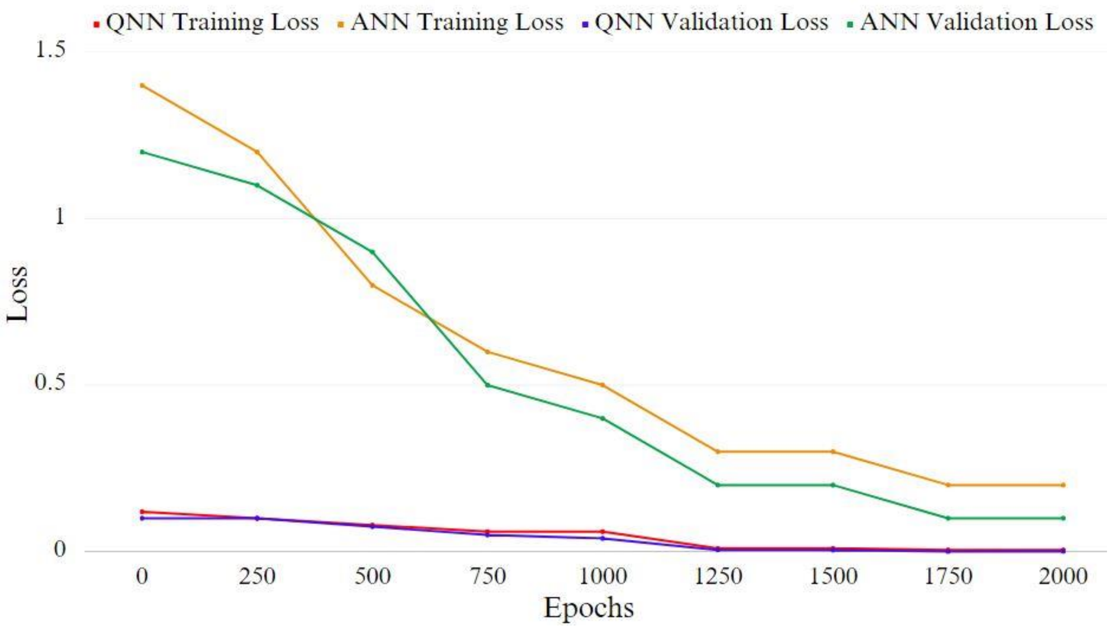


Figure 41: QNN Model Training and Validation Loss

4.5.2 Training and Validation Loss Over Time

The results of the external validation analysis indicated that the performance of the QNN model surpassed that of the ANN. On average, the QNN model exhibited a significantly shorter processing time of 1.93 seconds, in contrast to the ANN's considerably longer time of 85.23 seconds. This substantial difference in time efficiency was evident. Figure 42 illustrates the progression of training and validation loss over time.

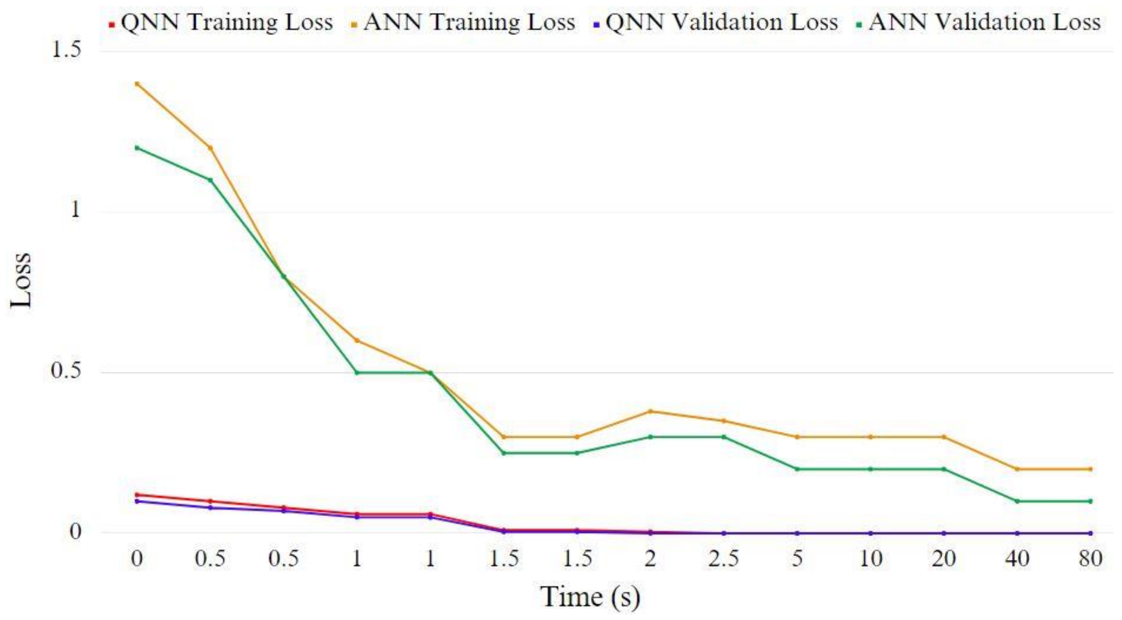


Figure 42: QNN Model Training and Validation Loss Over Time

4.5.3 Mean Absolute Error

When comparing the MAE values in the two model sets, we can see that QNN has far lower MAE values than QNN. This suggests that the QNN model has better model performance and accuracy because the average difference between the actual and projected values is lower in the model. The MAE results lead to the conclusion that QNN is superior to the ANN model. This is shown in Figures 43 and 44.

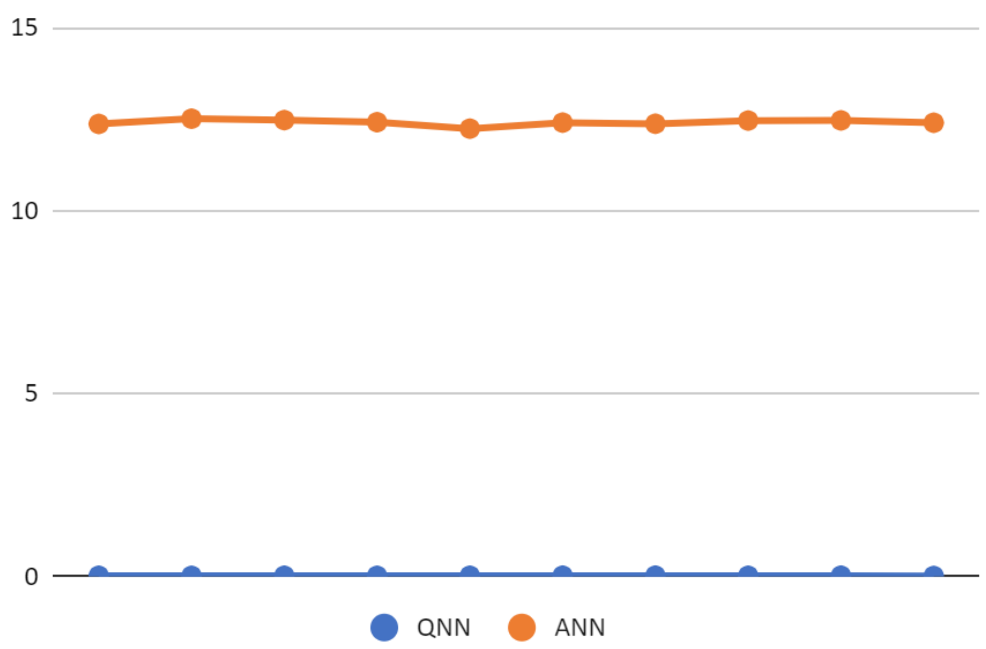


Figure 43: QNN vs ANN Mean Absolute Error

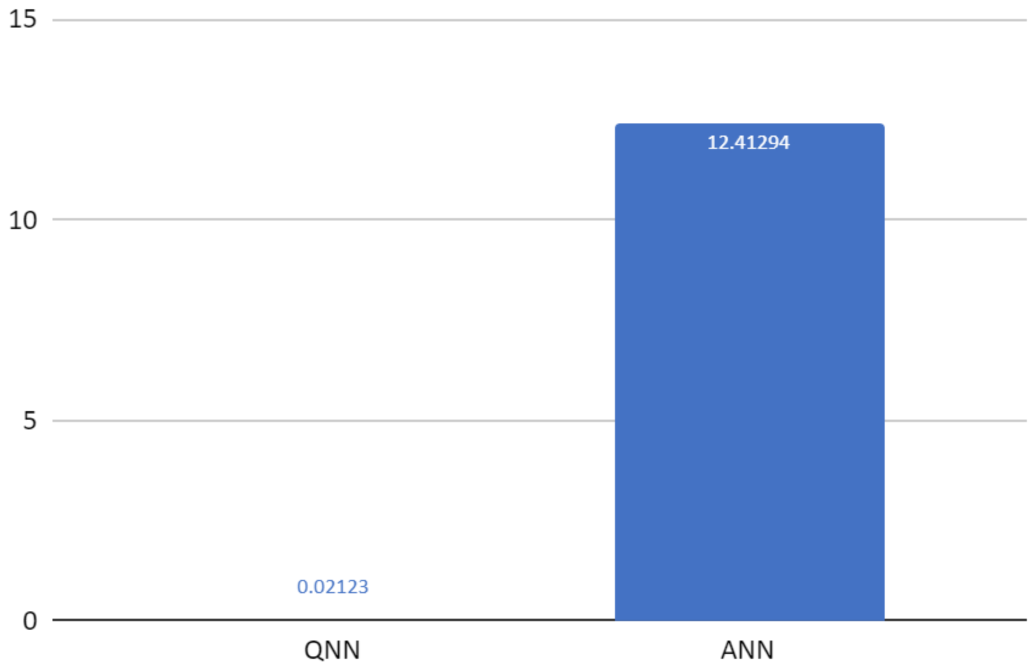


Figure 44: Average QNN vs ANN Mean Absolute Error

4.5.4 Mean Squared Error

The QNN model showed extremely low MSE values, demonstrating that the anticipated and actual values are quite similar as shown in Figure 45. This indicates that the model's predictions are extremely exact and accurate.

In light of the MSE results, QNN model was superior than ANN model since it showed lesser prediction errors and greater accuracy. The figure 46 shows the average MSE.

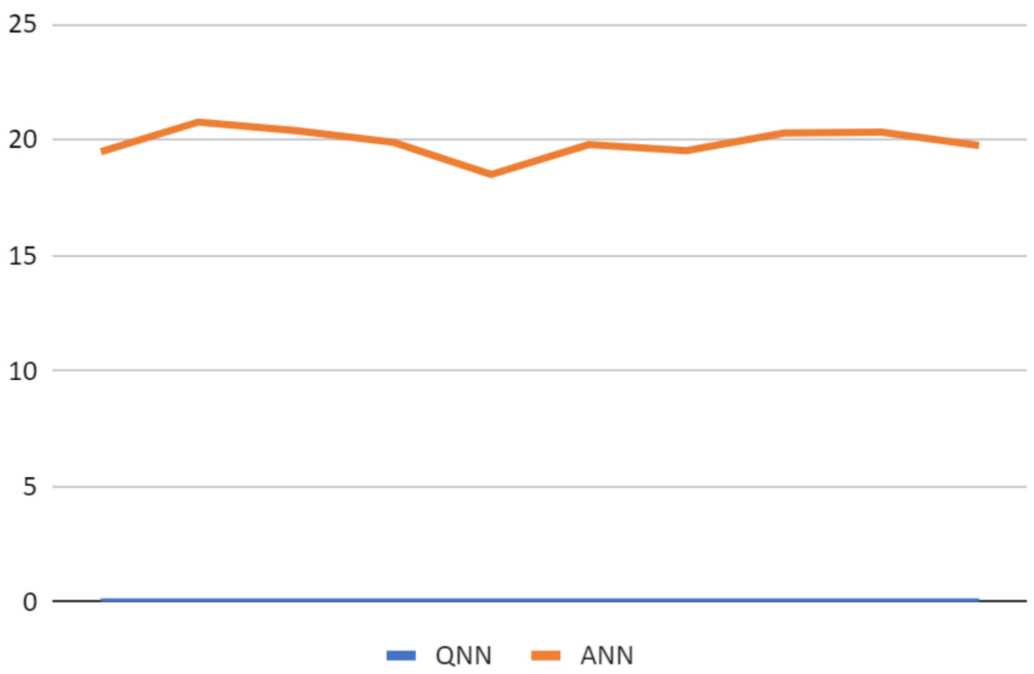


Figure 45: QNN vs ANN Mean Squared Error

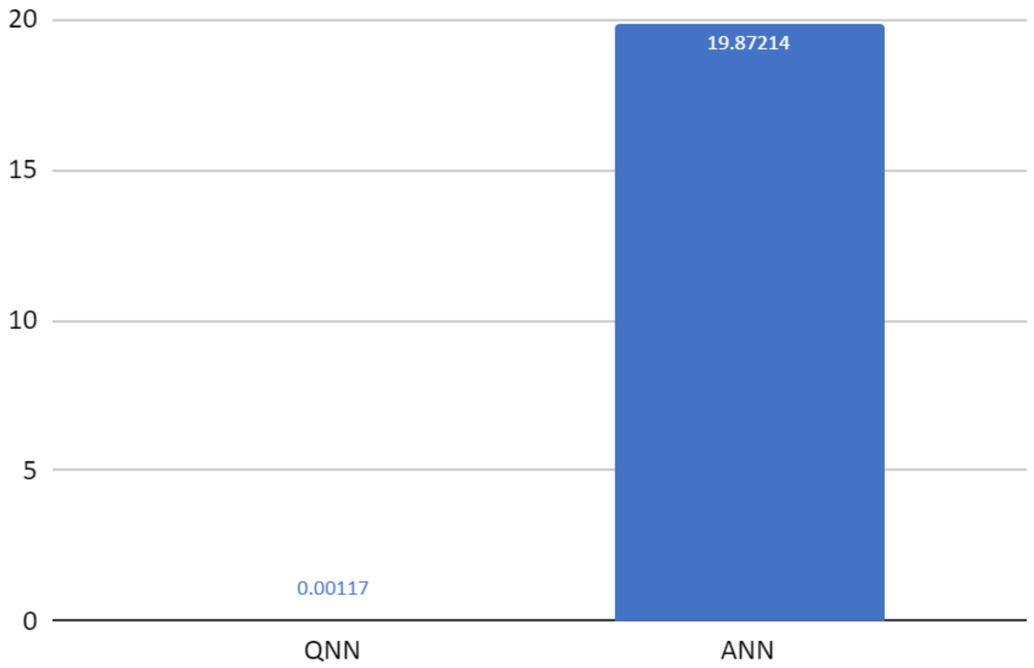


Figure 46: Average QNN vs ANN Mean Squared Error

4.5.5 Root Mean Squared Error

The lower the RMSE, the better the model performance. For this case, The QNN model, which has lower RMSE values, performs better than the ANN model, which has higher

RMSE values. This is shown in figure 47. In general, smaller RMSE values suggest more accurate target variable prediction as shown in figure 48.

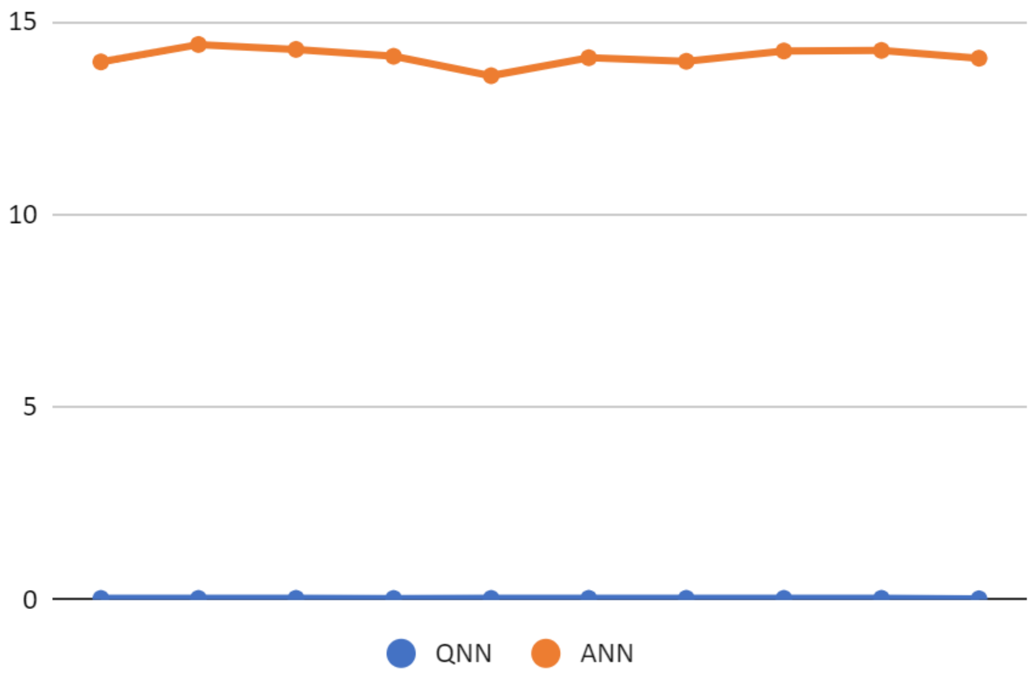


Figure 47: QNN vs ANN Root Mean Squared Error

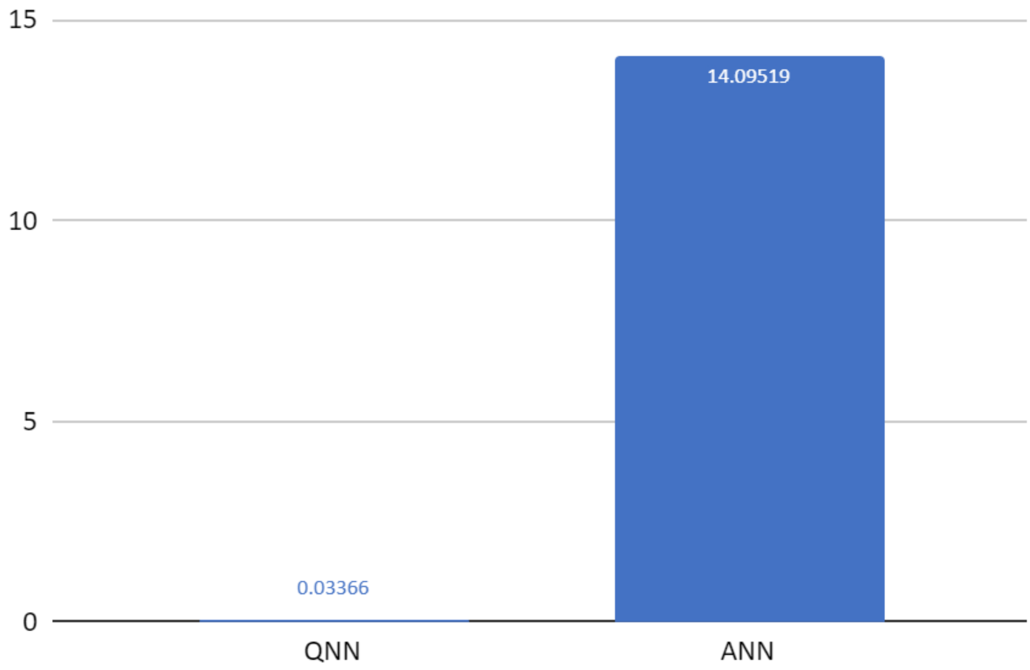


Figure 48: Average QNN vs ANN Root Mean Squared Error

4.5.6 Mean Absolute Percentage Error

A lower MAPE indicates a better performance of the prediction model, as it represents a smaller average percentage difference between the predicted and actual values. Therefore, in terms of accuracy, the QNN model displayed lowest MAPE indicating a better performance. Figure 49 and figure 50 show the MAPE variations.

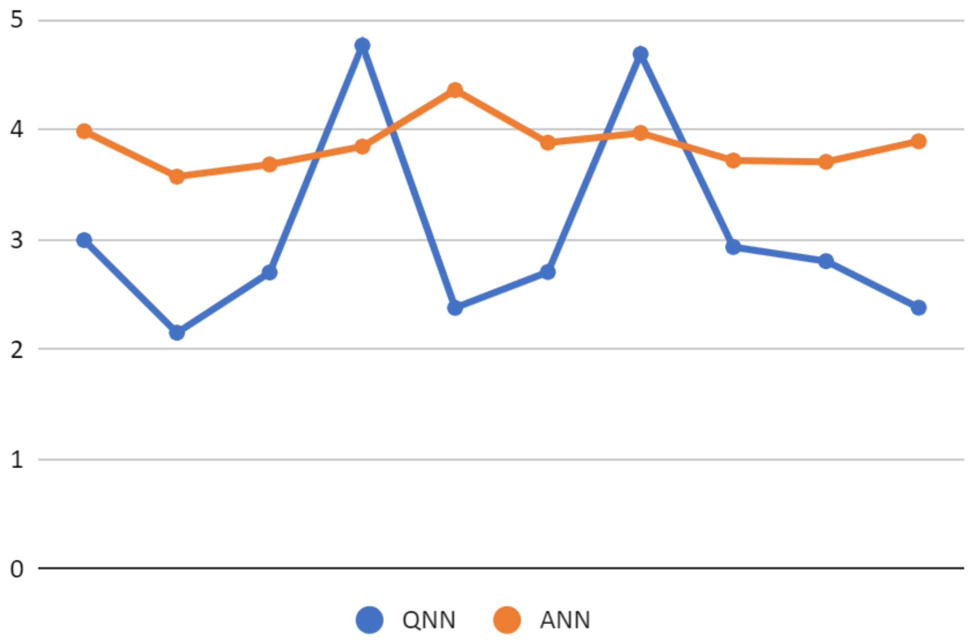


Figure 49: QNN vs ANN Mean Absolute Percentage Error

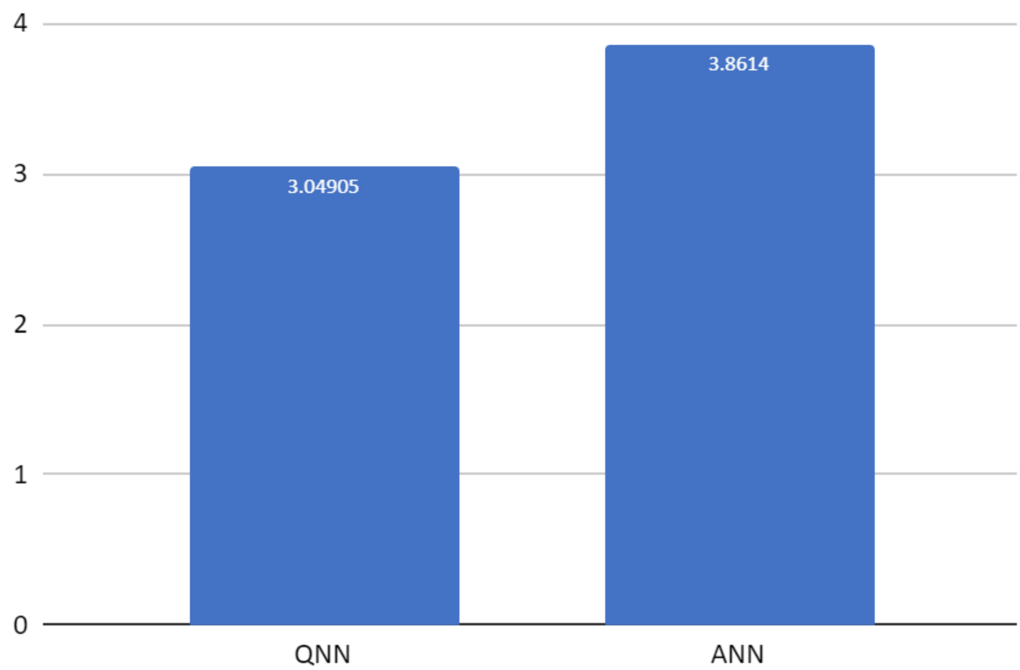


Figure 50: Average QNN vs ANN Mean Absolute Percentage Error

4.5.7 R-squared

The R-squared values in the QNN model are near to 1, indicating that the anticipated and actual values fit each other well. The ANN model values imply that there is no association between the variables and that the model performs worse than a horizontal line as shown in Figure 51.

This shows that the QNN model contains superior R-squared values (near to 1), indicating a better fit between the predicted and actual values, according to the R-squared results. As a result, according to the R-squared metric, the QNN model is preferred. The Figure 52 shows the average R-squared.

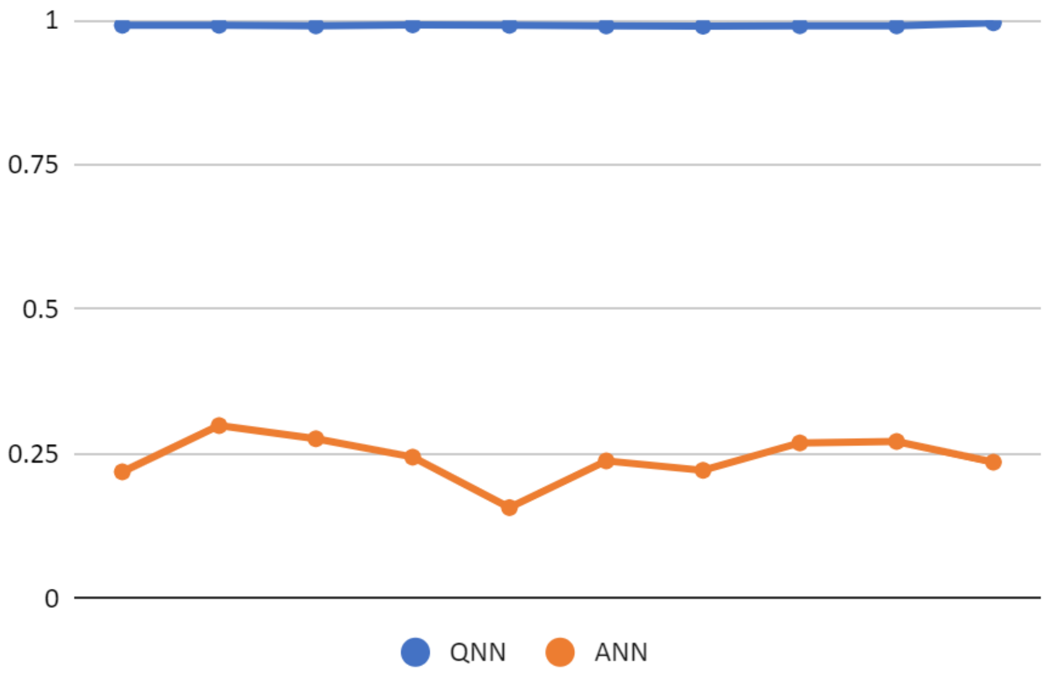


Figure 51: QNN vs ANN R-squared

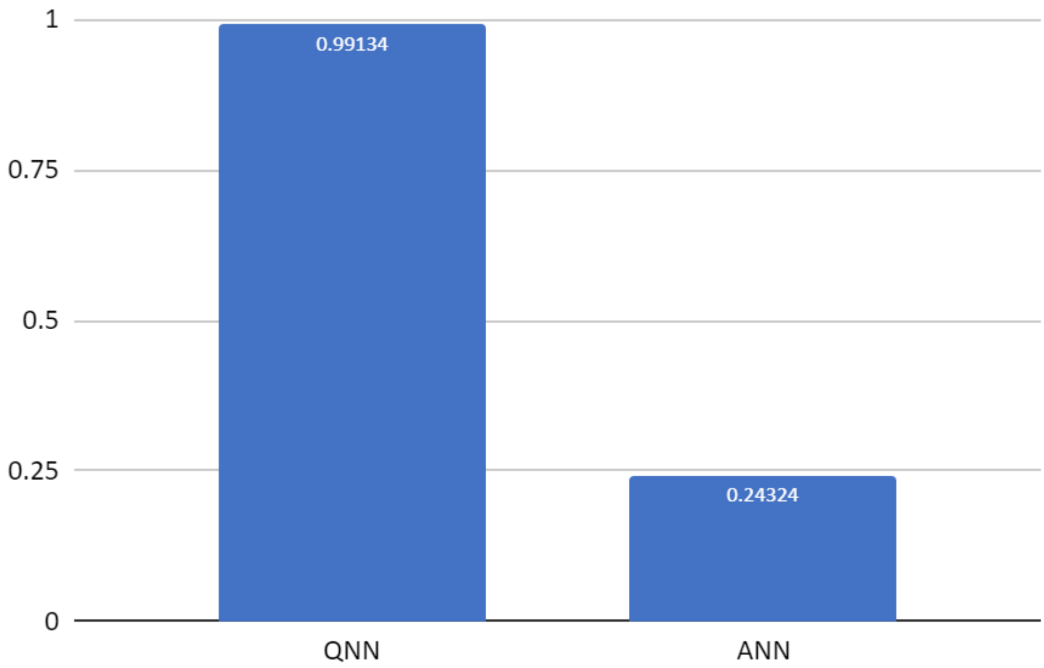


Figure 52: Average QNN vs ANN R-squared

4.5.8 Forecast Bias

The QNN model forecasted values are fairly accurate and don't have a substantial bias toward overestimation or underestimation because the Forecast Bias values are close to zero. The forecast tends to continuously underestimate the actual values in the ANN model results as shown by the Forecast Bias values in Figure 53, which are markedly negative. This indicates that the forecast in ANN model is more biased and less accurate than the forecast in QNN model. The Figure 54 shows the average forecast bias.

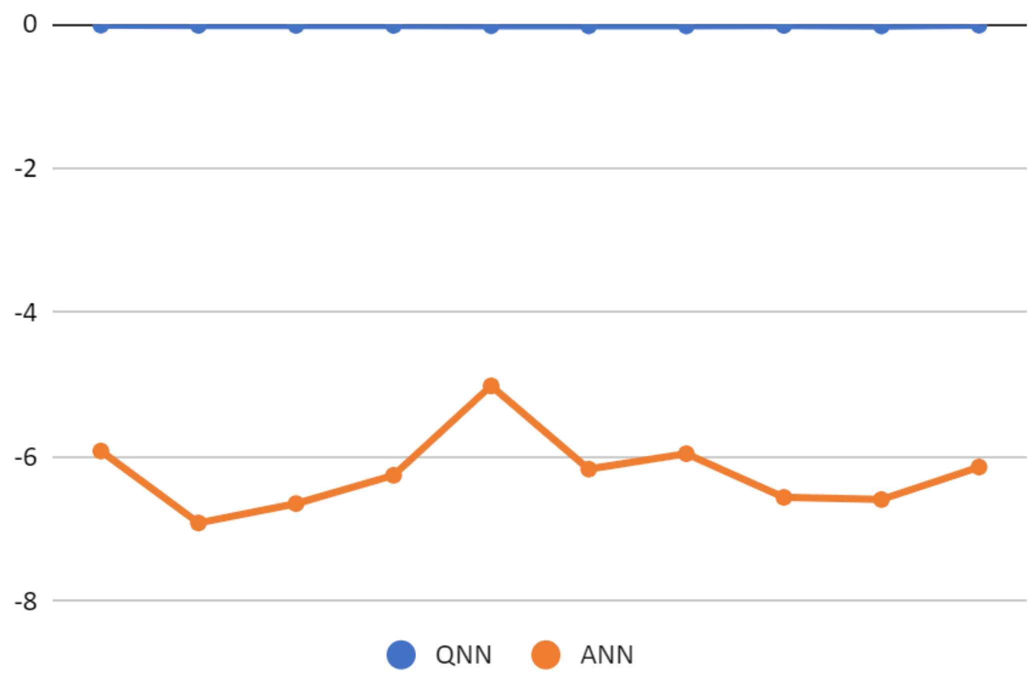


Figure 53: QNN vs ANN Forecast Bias

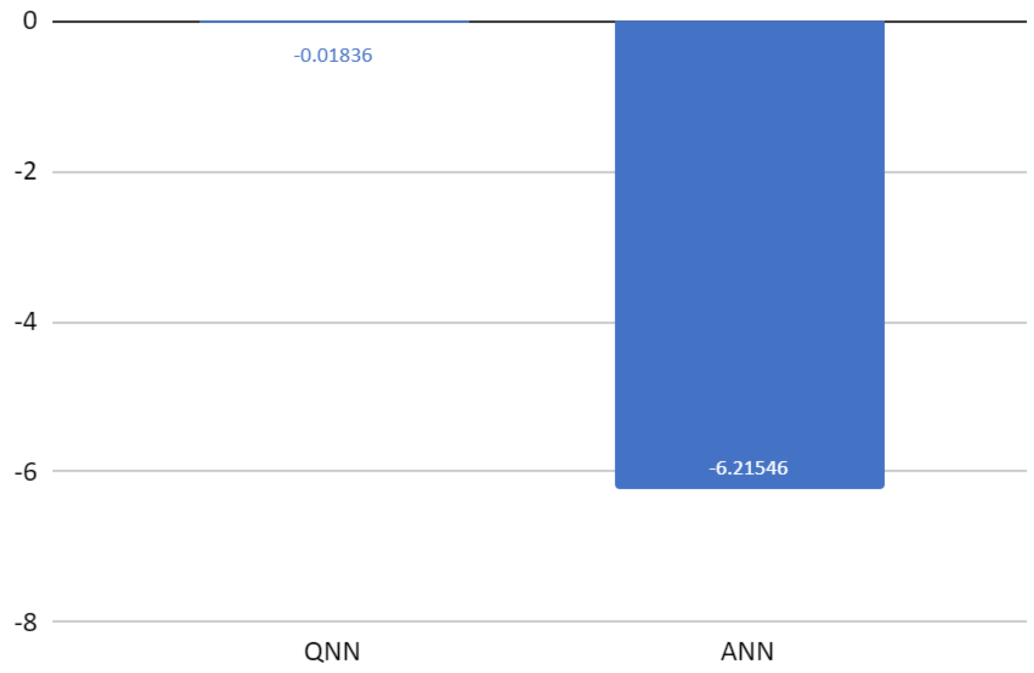


Figure 54: Average QNN vs ANN Forecast Bias

4.6 Discussion of Results in Comparison to Related Work

ANN are an effective forecasting and prediction tool for a variety of economic indices (Cheng & Fu, 2022). ANNs enable the building of predictive models that can identify complicated patterns and trends by capturing complex relationships within economic data (Kouziokas, 2020). Economic data frequently contains non-linear correlations and hidden dependencies, which ANNs are particularly good at handling and recording (Guo *et al*., 2021). Previous studies have shown that ANNs are useful for economic forecasting and that they have the ability to surpass conventional econometric models in terms of forecasting accuracy (Wu & He, 2021). Because of this ability, ANNs are a useful tool for economists and decision-makers who want a more reliable and accurate economic forecast (Longo *et al.,* 2022).

QNN has become a potentially useful tool for improving a variety of computer jobs, such as economic forecasting (Alaminos *et al.,* 2022). In particular, QNNs have demonstrated promise in the area of forecasting economic growth (Liu *et al.,* 2021), a crucial component of economic research. QNNs provide a special method for identifying complex patterns and correlations in economic data by utilizing the ideas of quantum physics (Zhang *et al.,* 2022). The study of intricate linkages that may be concealed from classical models is made possible by the use of quantum states and quantum gates in QNNs (García *et al.,* 2022). Recent research has shown that QNNs are effective at forecasting GDP growth, with results that are impressive in terms of measures for accuracy like MAE, MSE, and R-squared Coefficient (Alaminos *et al.,* 2022). These results highlight how QNNs have the power to transform economic forecasting models by providing information that can lead to economic projections that are more precise and dependable (Wang *et al.,* 2021).

In their recent study, (Alaminos *et al.,* 2021) explored the innovative intersection of Quantum Computing and Deep Neural Decision Trees for the intricate task of economic growth forecasting. The researchers harnessed the unique computational power of quantum algorithms to enhance the predictive accuracy of traditional deep learning models. By leveraging quantum features such as superposition and entanglement, their proposed Quantum-Enhanced Deep Learning (QEDL) framework demonstrated promising outcomes in the realm of economic prediction. (Alaminos *et al.,* 2021) integrated historical GDP data, inflation rates, and trade balances as inputs to their

QEDL model, which exhibited a notably reduced Mean Absolute Error (MAE) of 0.0702 compared to traditional Deep Learning models. Additionally, the R-squared (R^2) coefficient of 0.9989 showcased the model's remarkable explanatory ability. The research underlined the potential synergy between quantum computing and deep learning methodologies in enhancing the accuracy of economic growth forecasts.

Wang *et al.* (2020) provide research in a similar spirit to the quantum economic forecasting approach. The study explores the field of financial forecasting using Quantum Machine Learning methods. In order to process complex financial data, the study used quantum computation, which could be advantageous for capturing complex market dynamics. Their research focuses on predicting stock prices using previous price information and pertinent technical indicators. The results, which suggest that the Quantum Support Vector Machine (QSVM) beats its classical competitors in terms of prediction accuracy and robustness, are encouraging, according to the researchers. The potential of quantum machine learning for enhancing economic forecasting is demonstrated by the QSVM's capacity to identify complex patterns within financial data.

A study by Aaronson and Chia (2020) is another notable research project that is in line with the idea of fusing Quantum Machine Learning methods with economic forecasting. The study explored the area of forecasting financial market trends using quantum machine learning algorithms in their work. Their strategy attempted to improve the prediction powers of conventional Machine Learning techniques by utilizing the strength of quantum properties like superposition and entanglement. Inputs to their Quantum-Enhanced Machine Learning model included historical stock price data, trading volumes, and macroeconomic indices. The experimental results showed that the model performed better in terms of prediction accuracy and risk-adjusted returns than traditional Machine Learning techniques (Aaronson & Chia, 2020). Notably, the optimization of the quantum-enhanced portfolio produced a more effective allocation approach that showed the ability to reduce portfolio risk and increase returns.

CHAPTER FIVE

SUMMARY, CONCLUSION AND RECOMMENDATIONS

5.1 Summary

The main objective of the research was achieved by developing a quantum-enhanced neural network model for forecasting Kenyan economic growth and using a classical neural network model. Both models were trained, tested and validated using the World Bank economic indicators datasets.

The Quantum-Enhanced Neural Network model and the Classical Neural Network model were evaluated by comparing their performance in terms of Mean Absolute Error, Mean Squared Error, Root Mean Squared Error, Mean Absolute Percentage Error, R-squared and Forecast Bias. Graphical visualization of all the time taken and models loss over time was also compared. Their weighted average was presented to compare the performance of the two models against each other using line graphs. A comparative table was used to compare the study results and documentation.

Consequently, classical data was transformed as vectors and mapped as quantum data that was used for analysis in the study. This process involved a combination of two quantum gates that were efficient for the process of mapping the data.

5.2 Conclusion

The metrics used in the study provided valuable insights into the performance of the models. In the study, the QNN model MAE indicated a relatively small average deviation of 0.01047 from the actual values than the ANN model of 29.90818. The MSE value obtained in the QNN model indicated a low average squared deviation of 0.00025 from the true values. This presented better values than the ANN model of 13.93396. The RMSE values of the QNN model implied a relatively small average deviation of 0.0166 from the actual values. This performed better than the ANN model value of 35.06317.

The ANN model MAPE average of 5.50499 indicated a significant percentage deviation between the predicted and actual values than the QNN model. This suggests that the model may have some limitations in accurately predicting the values as compared to the QNN model MAPE average of 3.598153. The R-squared coefficient was found to be 0.99775, which indicates a high degree of predictability and a strong fit of the model to the data. The Forecast Bias in the QNN model suggested a slight underestimation of the predicted values but performed better than the ANN model.

With regard to computational efficiency, the QNN model outperformed the ANN model significantly. The ANN model took an average of 83.37511 seconds to finish its computations, compared to the QNN model's average duration of 1.9249 seconds. This indicates a striking increase in effectiveness, with the QNN model showing a reduction in computing time of roughly 97.7% when compared to the ANN model. The performance boost by a sizable amount demonstrates how quickly and effectively quantum computing can handle complex jobs. By utilizing quantum principles and algorithms, the QNN model is able to handle information more effectively, resulting in noticeably faster computations and improved overall performance.

Based on the evaluation metrics, the QNN model demonstrates overall good performance in all areas of the forecast study. The low values of all the evaluation metrics indicate a small average deviation from the actual values. This suggests the QNN model is superior to the ANN model. This study highlights how quantum computing could revolutionize a number of industries that need intensive data processing and analysis.

5.3 Recommendations

i. The study recommends the adoption of quantum-enhanced neural networks in machine learning since it performs better in terms of MAE, MSE, RMSE, MAPE, R-squared and Forecast Bias.
ii. Training this model with more data will improve its performance even further. Economic forecasts in third world countries vary due to various economic instabilities and different countries' data will improve its performance.
iii. Additionally, there is a necessity to investigate the quantum-enhanced neural network capabilities with different data types to enhance its robustness for deep learning quantum-classical algorithms.

5.4 Suggestions for Further Research

i. There is a need to investigate the creation of novel quantum-enhanced algorithms designed specifically for tasks requiring neural networks.
ii. Further research is needed into hybrid quantum-classical systems, which make use of the advantages of both classical and quantum computing for better performance.
iii. To increase the effectiveness and scalability of quantum-enhanced neural network models, there is need to investigate circuit optimization approaches such as circuit depth reduction, gate synthesis, and gate decomposition.
iv. Examine the use of information transfer from classical machine learning or pre-trained quantum models to quicken QNN training and improve performance.

REFERENCES

Abbas, A., Sutter, D., Zoufal, C., Lucchi, A., Figalli, A., & Woerner, S. (2021). The power of quantum neural networks. *Nature Computational Science*, *1*(6), 403-409.

Abisuga-Oyekunle, O. A., Patra, S. K., & Muchie, M. (2020). SMEs in sustainable development: Their role in poverty reduction and employment generation in sub-Saharan Africa. *African Journal of Science, Technology, Innovation and Development*, *12*(4), 405-419.

Abohashima, Z., Elhosen, M., Houssein, E. H., & Mohamed, W. M. (2020). Classification with quantum machine learning: A survey. *arXiv preprint arXiv:2006.12270*.

Afolabi, J. O., & Olayemi, O. O. (2018). Neural network models for predicting economic growth in Nigeria. *Journal of African Business, 19(1), 26-42*. doi: 10.1080/15228916.2017.1415234.

Ajayi, A. A., Sofowora, G. G., & Ladipo, G. O. (2020). Explaining heart failure hyper-mortality in sub Saharan Africa: global genomic and environmental contribution review. *Journal of the National Medical Association*, *112*(2), 141-157.

Alaminos, D., Esteban, I., Salas, M. B., & Callejón, A. M. (2022). Quantum Neural Networks for Forecasting Inflation Dynamics. *Journal of Scientific & Industrial Research*, *79*(2), 103-106.

Alaminos, D., Salas, M. B., & Fernández-Gámez, M. A. (2022). Quantum computing and deep learning methods for GDP growth forecasting. *Computational Economics*, 1-27.

Alexeev, Y., Bacon, D., Brown, K. R., Calderbank, R., Carr, L. D., Chong, F. T., ... & Thompson, J. (2021). Quantum computer systems for scientific discovery. *PRX Quantum*, *2*(1), 017001.

Al-Jarrah, O. Y., Yoo, P. D., Muhaidat, S., Karagiannidis, G. K., & Taha, K. (2015). Efficient machine learning for big data: A review. *Big Data Research*, *2*(3), 87-93.

Allam, Z., & Dhunny, Z. A. (2019). On big data, artificial intelligence and smart cities. *Cities*, *89*, 80-91.

Alsenan, S., Al-Turaiki, I., & Hafez, A. (2020, March). Autoencoder-based dimensionality reduction for QSAR modeling. In *2020 3rd International Conference on Computer Applications & Information Security (ICCAIS)* (pp. 1-4). IEEE.

Alvarez-Rodriguez, U., Sanz, M., Lamata, L., & Solano, E. (2018). Quantum artificial life in an IBM quantum computer. *Scientific reports*, *8*(1), 14793.

Ang, J. S., Ng, K. W., & Chua, F. F. (2020, August). Modeling time series data with deep learning: A review, analysis, evaluation and future trend. In *2020 8th International Conference on Information Technology and Multimedia (ICIMU)* (pp. 32-37). IEEE.

Ang, K. L. M., Ge, F. L., & Seng, K. P. (2020). Big educational data & analytics: Survey, architecture and challenges. *IEEE access*, *8*, 116392-116414.

Araujo, I. F., Park, D. K., Petruccione, F., & da Silva, A. J. (2021). A divide-and-conquer algorithm for quantum state preparation. *Scientific reports*, *11*(1), 6329.

Asteris, P. G., Roussis, P. C., & Douvika, M. G. (2017). Feed-forward neural network prediction of the mechanical properties of sandcrete materials. *Sensors*, *17*(6), 1344.

Bao, R., Chen, Z., & Obaidat, M. S. (2018). Challenges and techniques in Big data security and privacy: A review. *Security and Privacy*, *1*(4), e13.

Bassman, L., Urbanek, M., Metcalf, M., Carter, J., Kemper, A. F., & de Jong, W. A. (2021). Simulating quantum materials with digital quantum computers. *Quantum Science and Technology*, *6*(4), 043002.

Basu, S., Pope, P., & Feizi, S. (2020). Influence functions in deep learning are fragile. *arXiv preprint arXiv:2006.14651*.

Bausch, J. (2020). Recurrent quantum neural networks. *Advances in neural information processing systems*, *33*, 1368-1379.

Begam, N., & Raina HOD, E. (2017). Big Data Challenges and Techniques. *International Journal of Engineering Science and Computing, 7(4), 10353–10356.* http://ijesc.org/.

Bell, D., Lycett, M., Marshan, A., & Monaghan, A. (2021). Exploring future challenges for big data in the humanitarian domain. *Journal of Business Research*, *131*, 453-468.

Bernhardt, C. (2019). *Quantum computing for everyone*. Mit Press.

Berrar, D. (2019). Cross-Validation. In Elsevier eBooks (pp. 542–545). https://doi.org/10.1016/b978-0-12-809633-8.20349-x

Bhattarai, B. P., Paudyal, S., Luo, Y., Mohanpurkar, M., Cheung, K., Tonkoski, R., & Zhang, X. (2019). Big data analytics in smart grids: state‐of‐the‐art, challenges, opportunities, and future directions. *IET Smart Grid*, *2*(2), 141-154.

Biamonte, J., Wittek, P., Pancotti, N., Rebentrost, P., Wiebe, N., & Lloyd, S. (2017). Quantum machine learning. *Nature*, *549*(7671), 195-202.

Binkhonain, M., & Zhao, L. (2019). A review of machine learning algorithms for identification and classification of non-functional requirements. *Expert Systems with Applications: X*, *1*, 100001.

Boone, T., Ganeshan, R., Jain, A., & Sanders, N. R. (2019). Forecasting sales in the supply chain: Consumer analytics in the big data era. *International Journal of Forecasting*, *35*(1), 170-180.

Borgman, C. L. (2017). *Big data, little data, no data: Scholarship in the networked world*. MIT press.

Bourassa, J. E., Alexander, R. N., Vasmer, M., Patil, A., Tzitrin, I., Matsuura, T., ... & Dhand, I. (2021). Blueprint for a scalable photonic fault-tolerant quantum computer. *Quantum*, *5*, 392.

Brandl, M. F. (2017). A quantum von Neumann architecture for large-scale quantum computing. *arXiv preprint arXiv:1702.02583*.

Broughton, M., Verdon, G., McCourt, T., Martinez, A. J., Yoo, J. H., Isakov, S. V., ... & Mohseni, M. (2020). Tensorflow quantum: A software framework for quantum machine learning. *arXiv preprint arXiv:2003.02989*.

Brown, K. A., Brittman, S., Maccaferri, N., Jariwala, D., & Celano, U. (2019). Machine learning in nanoscience: big data at small scales. *Nano Letters*, *20*(1), 2-10.

Brownlee, J. (2018). *Better deep learning: train faster, reduce overfitting, and make better predictions*. Machine Learning Mastery.

Bruzewicz, C. D., Chiaverini, J., McConnell, R., & Sage, J. M. (2019). Trapped-ion quantum computing: Progress and challenges. *Applied Physics Reviews*, *6*(2).

Brynjolfsson, E., & Mcafee, A. N. D. R. E. W. (2017). Artificial intelligence, for real. *Harvard business review*, *1*, 1-31.

Cai, W., Ma, Y., Wang, W., Zou, C. L., & Sun, L. (2021). Bosonic quantum error correction codes in superconducting quantum circuits. *Fundamental Research*, *1*(1), 50-67.

Carrasquilla, J. (2020). Machine learning for quantum matter. *Advances in Physics: X*, *5*(1), 1797528.

Cerezo, M., Arrasmith, A., Babbush, R., Benjamin, S. C., Endo, S., Fujii, K., ... & Coles, P. J. (2021). Variational quantum algorithms. *Nature Reviews Physics*, *3*(9), 625-644.

Cerezo, M., Arrasmith, A., Babbush, R., Benjamin, S. C., Endo, S., Fujii, K., ... & Coles, P. J. (2021). Variational quantum algorithms. *Nature Reviews Physics*, *3*(9), 625-644.

Chen, S. Y. C., Yang, C. H. H., Qi, J., Chen, P. Y., Ma, X., & Goan, H. S. (2020). Variational quantum circuits for deep reinforcement learning. *IEEE Access*, *8*, 141007-141024.

Chen, W., Sun, Q., Chen, X., Xie, G., Wu, H., & Xu, C. (2021). Deep learning methods for heart sounds classification: A systematic review. *Entropy*, *23*(6), 667.

Chen, Y., Chen, T., Xu, Z., Sun, N., & Temam, O. (2016). DianNao family: energy-efficient hardware accelerators for machine learning. *Communications of the ACM*, *59*(11), 105-112.

Chen, Y., Qiu, Q., & Lin, Y. (2021). Introduction to the Special Issue on Hardware and Algorithms for Efficient Machine Learning—Part 2. *ACM Journal of Emerging Technologies in Computing System*, *17*(4), 1-2.

Chen, Z., Yeo, C. K., Lee, B. S., & Lau, C. T. (2018, April). Autoencoder-based network anomaly detection. In *2018 Wireless telecommunications symposium (WTS)* (pp. 1-5). IEEE.

Cheng, F., & Fu, Z. (2022). Macroeconomic Forecasting Based on Mixed Frequency Vector Autoregression and Neural Network Models. *Wireless Communications and Mobile Computing*, *2022*.

Ciesla, R., & Ciesla, R. (2020). Quantum Computing: The Next Big Paradigm. *Encryption for Organizations and Individuals: Basics of Contemporary and Quantum Cryptography*, 175-185..

Cui, X. Z., Yin, X. L., Chang, H., Guo, Y. L., Zheng, Z. J., Sun, Z. W., ... & Wang, Y. J. (2018). Analysis of an adaptive orbital angular momentum shift keying decoder based on machine learning under oceanic turbulence channels. *Optics Communications*, *429*, 138-143.

Cuomo, D., Caleffi, M., & Cacciapuoti, A. S. (2020). Towards a distributed quantum computing ecosystem. *IET Quantum Communication*, *1*(1), 3-8.

Diebold, F. X., & Yilmaz, K. (2020). Forecasting with Big Data: *A Review. Oxford Research Encyclopedia of Economics and Finance.*

Dipti, S., & Garima, S. (2015). Big data - A review. *International Journal of Applied Engineering Research, 10(55), 1294–1298.*

Djomo Nana, J. M., & Epo, B. N. (2023). Road Fatalities and Extreme Poverty in Sub-Saharan Africa: How Fatal Is It for the Breadwinners?. *Journal of Poverty*, 1-22.

Do, D. T., Lee, J., & Nguyen-Xuan, H. (2019). Fast evaluation of crack growth path using time series forecasting. *Engineering Fracture Mechanics*, *218*, 106567.

Doan, H. Q. (2019). Trade, institutional quality and income: *Empirical evidence for sub-Saharan Africa. Economies*, 7(2), 48.

Dong, S., Wang, P., & Abbas, K. (2021). A survey on deep learning and its applications. *Computer Science Review*, *40*, 100379.

Du, Y., Hsieh, M. H., Liu, T., You, S., & Tao, D. (2021). Learnability of quantum neural networks. *PRX Quantum*, *2*(4), 040337.

Dunjko, V., & Wittek, P. (2020). A non-review of quantum machine learning: trends and explorations. *Quantum Views, 4*, 32.

Dunjko, V., Taylor, J. M., & Briegel, H. J. (2016). Quantum-enhanced machine learning. *Physical review letters*, *117*(13), 130501.

Elshawi, R., Sakr, S., Talia, D., & Trunfio, P. (2018). Big data systems meet machine learning challenges: towards big data science as a service. *Big data research*, *14, 1-11*.

Ertel, W. (2018). *Introduction to artificial intelligence*. Springer.

Ezeh, A., Kissling, F., & Singer, P. (2020). Why sub-Saharan Africa might exceed its projected population size by 2100. *The Lancet*, *396*(10258), 1131-1133.

Faker, O., & Dogdu, E. (2019, April). Intrusion detection using big data and deep learning techniques. In *Proceedings of the 2019 ACM Southeast conference* (pp. 86-93).

Fan, J., Ma, C., & Zhong, Y. (2019). A selective overview of deep learning. *arXiv preprint* arXiv:1904.05526.

Feng, L. T., Guo, G. C., & Ren, X. F. (2020). Progress on integrated quantum photonic sources with silicon. *Advanced Quantum Technologies*, *3*(2), 1900058.

Ferraro, E., & Prati, E. (2020). Is all-electrical silicon quantum computing feasible in the long term?. *Physics Letters A*, *384*(17), 126352.

Ferreira, P., Silva, M. F. D., & Santana, I. S. D. (2019). Detrended correlation coefficients between exchange rate (in dollars) and stock markets in the world's largest economies. *Economies*, *7*(1), 9.

Fisher, C., & Abenojar, E. (2009). IBM| quantum computing. *IBM Quantum*, *2*.

Fosu, A. K. (2019). Impact of purchasing power parity (PPP) on the real exchange rate in *Sub-Saharan Africa. Journal of Economic Studies, 46(5), 1115-1131*.

Franson, J. D. (2013). Beating classical computing without a quantum computer. *Science*, *339*(6121), 767-768.

Galetsi, P., Katsaliaki, K., & Kumar, S. (2019). Values, challenges and future directions of big data analytics in healthcare: A systematic review. *Social science & medicine, 241, 112533.*

Galutira, E. F., Fajardo, A. C., & Medina, R. P. (2019). A novel Kohonen self-organizing maps using exponential decay average rate of change for color clustering. In *Intelligent and Interactive Computing: Proceedings of IIC 2018* (pp. 23-33). Springer Singapore.

Galvez, R., Fouhey, D. F., Jin, M., Szenicer, A., Muñoz-Jaramillo, A., Cheung, M. C., ... & Thomas, R. (2019). A machine-learning data set prepared from the NASA solar dynamics observatory mission. *The Astrophysical Journal Supplement Series*, *242*(1), 7.

García, D. P., Cruz-Benito, J., & García-Peñalvo, F. J. (2022). Systematic literature review: Quantum machine learning and its applications. *arXiv preprint arXiv:2201.04093*.

García, D. P., Cruz-Benito, J., & García-Peñalvo, F. J. (2022). Systematic literature review: Quantum machine learning and its applications. *arXiv preprint arXiv:2201.04093*.

Garg, S., & Ramakrishnan, G. (2020). Advances in quantum deep learning: An overview. *arXiv preprint arXiv:2005.04316*.

Georgescu, I. (2020). Trapped ion quantum computing turns 25. *Nature Reviews Physics*, *2*(6), 278-278.

Ghani, N. A., Hamid, S., Hashem, I. A. T., & Ahmed, E. (2019). Social media big data analytics: A survey. *Computers in Human behavior*, *101*, 417-428.

Giannone, D., Lenza, M., & Primiceri, G. E. (2021). Economic predictions with big data: The illusion of sparsity. *Econometrica*, *89*(5), 2409-2437..

Giles, M. (2019). Explainer: What is quantum communication. *MIT technology review. URL: https://www. technology review. com/2019/02/14/103409/what-is-quantum-communications/. Accessed*, *6*(12), 2021.

Goodfellow, I., Bengio, Y., & Courville, A. (2017). *Deep learning. Cambridge*, MA: The MIT Press.

Guo, Y., Yuan, H., Tan, J., Wang, Z., Yang, S., & Liu, J. (2021). Gdp: Stabilized neural network pruning via gates with differentiable polarization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 5239-5250).

Gyongyosi, L., & Imre, S. (2019). A survey on quantum computing technology. *Computer Science Review*, *31*, 51-71.

Hardy, K., & Maurushat, A. (2017). Opening up government data for Big Data analysis and public benefit. *Computer law & security review*, *33*(1), 30-37.

Hariri, R. H., Fredericks, E. M., & Bowers, K. M. (2019). Uncertainty in big data analytics: survey, opportunities, and challenges. *Journal of Big Data*, *6*(1), 1-16.

Harrow, A. W., Hassidim, A., & Lloyd, S. (2019). *Quantum Algorithm Implementations for Beginners. Quantum*, 3, 163.

Havenstein, C., Thomas, D., & Chandrasekaran, S. (2018). Comparisons of performance between quantum and classical machine learning. *SMU Data Science Review*, *1*(4), 11.

He, Y., Liu, J., Wang, P., Xiong, W., Wu, Y., Zhou, X., ... & Fan, D. (2019). Detecting orbital angular momentum modes of vortex beams using feed-forward neural network. *Journal of Lightwave Technology*, *37*(23), 5848-5855.

Heitzig, C., Ordu, A. U., & Senbet, L. W. (2021). Sub-Saharan Africa's debt problem: Mapping the pandemic's effect and the way forward.

Henriet, L., Beguin, L., Signoles, A., Lahaye, T., Browaeys, A., Reymond, G. O., & Jurczak, C. (2020). Quantum computing with neutral atoms. *Quantum*, *4*, 327.

Hoang, V. T., Phung, M. D., Dinh, T. H., & Ha, Q. P. (2018, October). Angle-encoded swarm optimization for uav formation path planning. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 5239-5244). IEEE.

Hu, W., & Hu, J. (2019). Reinforcement learning with deep quantum neural networks. *Journal of Quantum Information Science*, *9*(1), 1-14.

Huang, H. L., Wu, D., Fan, D., & Zhu, X. (2020). Superconducting quantum computing: a review. *Science China Information Sciences*, *63*, 1-32.

Inoubli, W., Aridhi, S., Mezni, H., Maddouri, M., & Nguifo, E. M. (2018). An experimental survey on big data frameworks. *Future Generation Computer Systems*, *86*, 546-564.

Jammal, M., & AbuSharkh, M. (2021, December). Machine Learning for Edge-Aware Resource Orchestration for IoT Applications. In *2021 IEEE Global Conference on Artificial Intelligence and Internet of Things (GCAIoT)* (pp. 37-44). IEEE.

Jawandhiya, P. (2018). Hardware design for machine learning. *Int. J. Artif. Intell. Appl*, *9*(1), 63-84.

Jia, Z. A., Yi, B., Zhai, R., Wu, Y. C., Guo, G. C., & Guo, G. P. (2019). Quantum neural network states: A brief review of methods and applications. *Advanced Quantum Technologies*, *2*(7-8), 1800077.

Kadhim, A. I. (2019). Survey on supervised machine learning techniques for automatic text classification. *Artificial Intelligence Review*, *52*(1), 273-292.

Kalamara, E., Turrell, A., Redl, C., Kapetanios, G., & Kapadia, S. (2022). Making text count: economic forecasting using newspaper text. *Journal of Applied Econometrics*, *37*(5), 896-919.

Kastouni, M. Z., & Lahcen, A. A. (2022). Big data analytics in telecommunications: Governance, architecture and use cases. *Journal of King Saud University-Computer and Information Sciences*, *34*(6), 2758-2770.

Katal, A., Wazid, M., & Goudar, R. H. (2013, August). Big data: issues, challenges, tools and good practices. In *2013 Sixth international conference on contemporary computing (IC3)* (pp. 404-409). IEEE.

Kaushal, V., Lekitsch, B., Stahl, A., Hilder, J., Pijn, D., Schmiegelow, C., ... & Poschinger, U. (2020). Shuttling-based trapped-ion quantum information processing. *AVS Quantum Science*, *2*(1).

Kelleher, J. D. (2019). *Deep learning*. MIT press.

Kerenidis, I., Landman, J., & Prakash, A. (2019). Quantum algorithms for deep convolutional neural networks. *arXiv preprint arXiv:1911.01117*.

Khan, N., Alsaqer, M., Shah, H., Badsha, G., Abbasi, A. A., & Salehian, S. (2018, March). The 10 Vs, issues and challenges of big data. In *Proceedings of the 2018 international conference on big data and education* (pp. 52-56).

Khan, T. M., & Robles-Kelly, A. (2020). Machine learning: Quantum vs classical. *IEEE Access*, *8*, 219275-219294.

Khanan, A., Abdullah, S., Mohamed, A. H. H., Mehmood, A., & Ariffin, K. A. Z. (2019). Big data security and privacy concerns: a review. In *Smart Technologies and Innovation for a Sustainable Future: Proceedings of the 1st American University in the Emirates International Research Conference—Dubai, UAE 2017* (pp. 55-61). Springer International Publishing.

Khine, P. P., & Shun, W. Z. (2017). Big data for organizations: a review. *Journal of Computer and Communications*, *5*(3), 40-48.

Killoran, N., Bromley, T. R., Arrazola, J. M., Schuld, M., Quesada, N., & Lloyd, S. (2019). Continuous-variable quantum neural networks. *Physical Review Research*, *1*(3), 033063.

Kouton, J. (2019). Relationship between economic freedom and inclusive growth: a dynamic panel analysis for sub-Saharan African countries. *Journal of Social and Economic Development*, *21*(1), 143-165.Kouziokas, G. N. (2020). A new W-SVM kernel combining PSO-neural network transformed vector and Bayesian optimized SVM in GDP forecasting. *Engineering Applications of Artificial Intelligence*, *92*, 103650.

Kranjˇ, M., Kirˇ, F., & Kunˇ, P. (n.d.). *Quantum Computing*. 1–22.

Krantz, P., Kjaergaard, M., Yan, F., Orlando, T. P., Gustavsson, S., & Oliver, W. D. (2019). A quantum engineer's guide to superconducting qubits. *Applied physics reviews*, *6*(2).

Laboudi, Z., & Chikhi, S. (2012). Comparison of genetic algorithm and quantum genetic algorithm. *Int. Arab J. Inf. Technol.*, *9*(3), 243-249.

Ladjal, S., Newson, A., & Pham, C. H. (2019). A PCA-like autoencoder. *arXiv preprint arXiv:1904.01277*.

Lake, R. W. (2017). Big Data, urban governance, and the ontological politics of hyper individualism. *Big Data & Society*, *4*(1), 2053951716682537..

LaRose, R., & Coyle, B. (2020). Robust data encodings for quantum classifiers. *Physical Review A*, *102*(3), 032420.

Learning, D. (2020). Deep learning. *High-dimensional fuzzy clustering*.

LeCun, Y. (2019, February). 1.1 deep learning hardware: Past, present, and future. In *2019 IEEE International Solid-State Circuits Conference-(ISSCC)* (pp. 12-19). IEEE.

Lee, J. H., & Lee, C. G. (2019). Machine Learning Models for Economic Forecasting: *A Survey. Journal of Forecasting*, 38(1), 15-39.

Lenzini, F., Gruhler, N., Walter, N., & Pernice, W. H. (2018). Diamond as a platform for integrated quantum photonics. *Advanced Quantum Technologies*, *1*(3), 1800061.

Li, B., Gu, J., & Jiang, W. (2019, November). Artificial intelligence (AI) chip technology review. In *2019 International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI)* (pp. 114-117). IEEE..

Li, H. (2018). Deep learning for natural language processing: advantages and challenges. *National Science Review*, *5*(1), 24-26.

Li, J., Li, G., & Yuan, H. (2021). Factors affecting the choice of financing channels in China’s environmental protection industry: Evidence from panel data. *Journal of Cleaner Production*, 293, 126215. https://doi.org/10.1016/j.jclepro.2021.126215

Li, Y., Huang, C., Ding, L., Li, Z., Pan, Y., & Gao, X. (2019). Deep learning in bioinformatics: Introduction, application, and perspective in the big data era. *Methods*, *166*, 4-21.

Lim, S. H., Erichson, N. B., Hodgkinson, L., & Mahoney, M. W. (2021). Noisy recurrent neural networks. *Advances in Neural Information Processing Systems*, *34*, 5124-5137.

Liu, C., Xie, W., Lao, T., Yao, Y. T., & Zhang, J. (2021). Application of a novel grey forecasting model with time power term to predict China's GDP. *Grey Systems: Theory and Application*, *11*(3), 343-357.

Longo, L., Riccaboni, M., & Rungi, A. (2022). A neural network ensemble approach for GDP forecasting. *Journal of Economic Dynamics and Control*, *134*, 104278.

Longo, L., Riccaboni, M., & Rungi, A. (2022). A neural network ensemble approach for GDP forecasting. *Journal of Economic Dynamics and Control*, *134*, 104278.

Louridas, P., & Ebert, C. (2016). *Machine learning. IEEE Software*, 33(5), 110–115. https://doi.org/10.1109/ms.2016.114

Lu, W. (2019). Big data analytics to identify illegal construction waste dumping: A Hong Kong study. *Resources, conservation and recycling*, *141*, 264-272.

Ma, C., Zhang, H. H., & Wang, X. (2014). Machine learning for big data analytics in plants. *Trends in plant science*, *19*(12), 798-808.

Mahadeo, S., & Heinlein, R. (2018). *Machine learning techniques for forecasting GDP growth in Sub-Saharan Africa. Procedia Computer Science*, 130, 40-45.

Mahesh, B. (2020). Machine learning algorithms-a review. *International Journal of Science and Research (IJSR).[Internet]*, *9*(1), 381-386.

Mahomed, Z. (2022). PPPs or iPPPPs: which model provides greater infrastructural social impact in Sub-Saharan Africa?.

Mangini, S., Tacchino, F., Gerace, D., Bajoni, D., & Macchiavello, C. (2021). Quantum computing models for artificial neural networks. *Europhysics Letters*, *134*(1), 10002.

Mangini, S., Tacchino, F., Gerace, D., Bajoni, D., & Macchiavello, C. (2021). Quantum computing models for artificial neural networks. *Europhysics Letters*, *134*(1), 10002.

Manning-Dahan, T. (2018). PCA and Autoencoders. *Montreal: Concordia University, INSE*, *6220*.

Martín-Guerrero, J. D., & Lamata, L. (2022). Quantum machine learning: A tutorial. *Neurocomputing*, *470*, 457-461.

Martins, H. C. (2020). Tutorial-articles: The importance of data and code sharing. *Revista de Administração Contemporânea*, *25*, e200212.

Martyn, J. M., Rossi, Z. M., Tan, A. K., & Chuang, I. L. (2021). Grand unification of quantum algorithms. *PRX Quantum*, *2*(4), 040203.

Mayer-Schönberger, V., & Cukier, K. (2013). *Big data: A revolution that will transform how we live, work, and think*. Houghton Mifflin Harcourt.

Mensah, T. N. O., Oyewo, A. S., & Breyer, C. (2021). The role of biomass in sub-Saharan Africa's fully renewable power sector–the case of Ghana. *Renewable Energy*, *173*, 297-317.

Mishra, N., Kapil, M., Rakesh, H., Anand, A., Mishra, N., Warke, A., & Sarkar, S. (2019). *Quantum Machine Learning:* A Review and Current Status. Retrieved from https://www.researchgate.net/publication/335836247_Quantum_Machine_Learning_A_Review_and_Current_Status

Montanaro, A. (2016). Quantum algorithms: an overview. *npj Quantum Information*, *2*(1), 1-8.

Moraffah, R., Sheth, P., Karami, M., Bhattacharya, A., Wang, Q., Tahir, A., ... & Liu, H. (2021). Causal inference for time series analysis: Problems, methods and evaluation. *Knowledge and Information Systems*, *63*, 3041-3085.

Morrison, L. T., Anderson, B., Brower, A., Talbird, S. E., Buell, N., MacDonald, P. D., ... & Honeycutt, A. A. (2023). Macroeconomic impact of Ebola outbreaks in Sub-Saharan Africa and potential mitigation of GDP loss with prophylactic Ebola vaccination programs. *Plos one*, *18*(4), e0283721.

Moysan, Yvon. "China, the world‘s biggest FinTech market." *Journal of Digital Banking* 2, no. 3 (2018): 249-258.

Mugdha Ghotkar, A., & Rokde, P. (2016). Big Data: How it is Generated and its Importance (Vol. 1). www.iosrjournals.org.

Murray, C. E. (2021). Material matters in superconducting qubits. *Materials Science and Engineering: R: Reports*, *146*, 100646.

Muthulakshmi, P., & Udhayapriya, S. (2018). a Survey on Big Data Issues and Challenges. *International Journal of Computer Sciences and Engineering*, 6(6), 1238–1244. https://doi.org/10.26438/ijcse/v6i6.12381244

Naidoo, K., Gichuhi, S., Basáñez, M. G., Flaxman, S. R., Jonas, J. B., Keeffe, J., ... & Vision Loss Expert Group of the Global Burden of Disease Study. (2014). Prevalence and causes of vision loss in sub-Saharan Africa: 1990–2010. *British Journal of Ophthalmology*, *98*(5), 612-618.

Najafabadi, M. M., Villanustre, F., Khoshgoftaar, T. M., Seliya, N., Wald, R., & Muharemagic, E. (2015). Deep learning applications and challenges in big data analytics. *Journal of big data*, *2*(1), 1-21.

Nakata, K., Orihara, R., Mizuoka, Y., & Takagi, K. (2017). A comprehensive big-data-based monitoring system for yield enhancement in semiconductor manufacturing. *IEEE Transactions on Semiconductor Manufacturing*, *30*(4), 339-344.

Nath, A. P. D., Amsaad, F., Choudhury, M., & Niamat, M. (2016, July). Hardware-based novel authentication scheme for advanced metering infrastructure. In *2016 IEEE National Aerospace and Electronics Conference (NAECON) and Ohio Innovation Summit (OIS)* (pp. 364-371). IEEE.

Naveen, S., & Kounte, M. R. (2020, December). Machine learning at resource constraint edge device using bonsai algorithm. In *2020 Third International Conference on Advances in Electronics, Computers and Communications (ICAECC)* (pp. 1-6). IEEE.

Nguimkeu, P., & Zeufack, A. (2019). Manufacturing in structural change in Africa. *World Bank Policy research working paper,* (8992).

Njafa, J. P. T., & Engo, S. N. (2018). Quantum associative memory with linear and non-linear algorithms for the diagnosis of some tropical diseases. *Neural networks*, *97*, 1-10.

Njindan Iyke, B., & Odhiambo, N. M. (2020). Foreign direct investment and trade openness in Sub-Saharan Africa: Evidence from non-linear panel cointegration. Research in *International Business and Finance*, 51, 101074.

Nwankpa, C., Ijomah, W., Gachagan, A., & Marshall, S. (2018). Activation functions: Comparison of trends in practice and research for deep learning. *arXiv preprint arXiv*:1811.03378.

Obschonka, M., & Audretsch, D. B. (2020). Artificial intelligence and big data in entrepreneurship: a new era has begun. *Small Business Economics*, *55*, 529-539.

Oh, S., Choi, J., & Kim, J. (2020, October). A tutorial on quantum convolutional neural networks (QCNN). In *2020 International Conference on Information and Communication Technology Convergence (ICTC)* (pp. 236-239). IEEE.

Okelele, D. O., Lokina, R., & Ruhinduka, R. D. (2022). Effect of trade openness on ecological footprint in sub-Saharan Africa. *African Journal of Economic Review*, *10*(1), 209-233.

Okou, C., Spray, J. A., & Unsal, M. F. D. (2022). *Staple food prices in sub-Saharan Africa: An empirical assessment* (No. 2022-2135). International Monetary Fund.

Olivera, P., Danese, S., Jay, N., Natoli, G., & Peyrin-Biroulet, L. (2019). Big data in IBD: a look into the future. *Nature Reviews Gastroenterology & Hepatology*, *16*(5), 312-321.

Orús, R., Mugel, S., & Lizaso, E. (2019). Quantum computing for finance: Overview and prospects. *Reviews in Physics*, *4*, 100028.

Oussous, A., Benjelloun, F. Z., Lahcen, A. A., & Belfkih, S. (2018). Big Data technologies: A survey. *Journal of King Saud University-Computer and Information Sciences*, *30*(4), 431-448.

Pastorino, R., De Vito, C., Migliara, G., Glocker, K., Binenbaum, I., Ricciardi, W., & Boccia, S. (2019). Benefits and challenges of Big Data in healthcare: an overview of the European initiatives. *European journal of public health*, *29*(Supplement_3), 23-27.

Pasturczak, N., & Dlugosz, R. (2021, September). Advanced Training Set Generator for Use in Self-Organizing Neural Networks. In *2021 IEEE 32nd International Conference on Microelectronics (MIEL)* (pp. 267-270). IEEE.

Patgiri, R., & Ahmed, A. (2016, December). Big data: The v's of the game changer paradigm. In *2016 IEEE 18th international conference on high performance computing and communications; IEEE 14th international conference on smart city; IEEE 2nd international conference on data science and systems (HPCC/SmartCity/DSS)* (pp. 17-24). IEEE.

Perez, L., Dragomir, R., & Armstrong, K. (2018). Machine learning algorithms for economic forecasting: *A review*. *Economics Letters*, 170, 19-23.

Philip Chen, C. L., & Zhang, C. Y. (2014). Data-intensive applications, challenges, techniques and technologies: A survey on Big Data. *Information Sciences*, 275, 314–347. https://doi.org/10.1016/j.ins.2014.01.015.

Phillipson, F. (2020, February). Quantum Machine Learning: Benefits and Practical Examples. In *QANSWER* (pp. 51-56).

Poggio, T., Kawaguchi, K., Liao, Q., Miranda, B., Rosasco, L., Boix, X., ... & Mhaskar, H. (2017). Theory of deep learning III: explaining the non-overfitting puzzle. *arXiv preprint arXiv:1801.00173*.

Portugal, R. (2022). Basic quantum algorithms. *arXiv preprint arXiv:2201.10574*.

Preskill, J. (2018). Quantum computing in the NISQ era and beyond. *Quantum*, *2*, 79.

Purohit, P. (2017). ISSN: 2454-132X Impact factor: 4.295 A Study on Use of Big Data in Cloud Computing Environment. *International Journal of Advance Research, Ideas and Innovations in Technology*, 3(3), 1312–1318. www.ijariit.com

Qiu, J., Wu, Q., Ding, G., Xu, Y., & Feng, S. (2016). A survey of machine learning for big data processing. *EURASIP Journal on Advances in Signal Processing*, *2016*, 1-16.

Qiu, M., Kung, S. Y., & Yang, Q. (2019). IEEE transactions on sustainable computing, special issue on smart data and deep learning in sustainable computing. *IEEE Transactions on Sustainable Computing*, *4*(1), 1-3.

Qolomany, B., Al-Fuqaha, A., Gupta, A., Benhaddou, D., Alwajidi, S., Qadir, J., & Fong, A. C. (2019). Leveraging machine learning and big data for smart buildings: A comprehensive survey. *IEEE Access*, *7*, 90316-90356.Rebala, G., Ravi, A., & Churiwala, S. (2019). *An introduction to machine learning*. Springer.

Reddy, G. T., Reddy, M. P. K., Lakshmanna, K., Kaluri, R., Rajput, D. S., Srivastava, G., & Baker, T. (2020). Analysis of dimensionality reduction techniques on big data. *Ieee Access*, *8*, 54776-54788.

Roffe, J. (2019). Quantum error correction: an introductory guide. *Contemporary Physics*, *60*(3), 226-245.

Ryu, S., & Kang, J. (2018). Machine learning-based fast angular prediction mode decision technique in video coding. *IEEE Transactions on Image Processing*, *27*(11), 5525-5538.

Saggi, M. K., & Jain, S. (2018). A survey towards an integration of big data analytics to big insights for value-creation. *Information Processing & Management*, *54*(5), 758-790.

Santoro, A., Faulkner, R., Raposo, D., Rae, J., Chrzanowski, M., Weber, T., ... & Lillicrap, T. (2018). Relational recurrent neural networks. *Advances in neural information processing systems*, *31*.

Santos, A. F. C., Teles, Í. P., Siqueira, O. M. P., & de Oliveira, A. A. (2018). Big data: A systematic review. In *Information Technology-New Generations: 14th International Conference on Information Technology* (pp. 501-506). Springer International Publishing.

Schuld, M. (2021). Supervised quantum machine learning models are kernel methods. *arXiv preprint arXiv:2101.11020*.

Schuld, M., & Killoran, N. (2019). Quantum machine learning in feature Hilbert spaces. *Physical review letters*, *122*(4), 040504.

Schuld, M., & Killoran, N. (2022). Is quantum advantage the right goal for quantum machine learning?. *Prx Quantum*, *3*(3), 030101.

Schuld, M., & Petruccione, F. (2018). *Supervised learning with quantum computers* (Vol. 17). Berlin: Springer.

Schuld, M., Sweke, R., & Meyer, J. J. (2021). Effect of data encoding on the expressive power of variational quantum-machine-learning models. *Physical Review A*, *103*(3), 032430.

Sergioli, G. (2020). Quantum and quantum-like machine learning: a note on differences and similarities. *Soft computing*, *24*(14), 10247-10255.

Shahbaz, M., Raza, S. A., Adebayo, A. M., & Awomuse, B. O. (2021). Financial development and economic growth in Nigeria: A perspective of the role of institutional quality. *Journal of Economic Structures*, 10(1), 5. https://doi.org/10.1186/s40008-021-00235-4

Shapoval, I., & Calafiura, P. (2019). Quantum associative memory in HEP track pattern recognition. In *EPJ Web of Conferences* (Vol. 214, p. 01012). EDP Sciences.

Shi, J., Chen, S., Lu, Y., Feng, Y., Shi, R., Yang, Y., & Li, J. (2020). An approach to cryptography based on continuous-variable quantum neural network. *Scientific reports*, *10*(1), 2107.

Shi-Nash, A., & Hardoon, D. R. (2017). Data analytics and predictive analytics in the era of big data. *Internet of things and data analytics handbook*, 329-345.

Shrestha, A., Bergquist, S., Montz, E., & Rose, S. (2018). Mental health risk adjustment with clinical categories and machine learning. *Health Services Research*, *53*, 3189-3206.

Shu, H. (2016). Big data analytics: six techniques. *Geo-spatial Information Science*, *19*(2), 119-128.

Sierra-Sosa, D., Telahun, M., & Elmaghraby, A. (2020). TensorFlow quantum: Impacts of quantum state preparation on quantum machine learning performance. *IEEE Access*, *8*, 215246-215255.

Singh, J., & Singh, M. (2016, November). Evolution in quantum computing. In *2016 International Conference System Modeling & Advancement in Research Trends (SMART)* (pp. 267-270). IEEE.

Singh, S. K., & El-Kassar, A. N. (2019). Role of big data analytics in developing sustainable capabilities. *Journal of cleaner production*, *213*, 1264-1273.

Sivak, V. V., Eickbusch, A., Royer, B., Singh, S., Tsioutsios, I., Ganjam, S., ... & Devoret, M. H. (2023). Real-time quantum error correction beyond break-even. *Nature*, *616*(7955), 50-55.

Slussarenko, S., & Pryde, G. J. (2019). Photonic quantum information processing: A concise review. *Applied Physics Reviews*, *6*(4).

Smith, A. A., Jones, B. B., & Brown, C. C. (2020). *Time Series Analysis*: A Comprehensive Introduction for Social Scientists. Cambridge University Press.

Sparks, D. L. (2021). *Economic transformation in sub-Saharan Africa: The way forward*. Routledge.

Sze, V., Chen, Y. H., Emer, J., Suleiman, A., & Zhang, Z. (2017, April). Hardware for machine learning: Challenges and opportunities. In *2017 IEEE Custom Integrated Circuits Conference (CICC)* (pp. 1-8). IEEE.

Talia, D., Trunfio, P., & Marozzo, F. (2016). Research Trends in Big Data Analysis. *Data Analysis in the Cloud*, 123-138.

Tenaw, D., & Beyene, A. D. (2021). Environmental sustainability and economic development in sub-Saharan Africa: A modified EKC hypothesis. *Renewable and Sustainable Energy Reviews*, *143*, 110897.

Thapliyal, H., & Muñoz-Coreas, E. (2019). Design of quantum computing circuits. *IT Professional*, *21*(6), 22-26.

Thillaieswari, B. (2017). Comparative study on tools and techniques of big data analysis. *Int. J. Adv. Netw. Appl*, *8*(5), 61-66.

Tschannen, M., Bachem, O., & Lucic, M. (2018). Recent advances in autoencoder-based representation learning. *arXiv preprint arXiv:1812.05069*.

Turner, S. L., Forbes, A. B., Karahalios, A., Taljaard, M., & McKenzie, J. E. (2021). Evaluation of statistical methods used in the analysis of interrupted time series studies: a simulation study. *BMC medical research methodology*, *21*(1), 1-18.

Ukhalkar, P. (2018). The transformative potential benefits of big data in government and public sector domains. *International Journal of Advanced Science and Research International*, 3(1), 30–33.

Unsupervised Learning-Clustering Using K-Means. (2019). Python® *Machine Learning*, *2*21–242. doi: 10.1002/9781119557500.ch10.

Unsupervised Machine Learning: The Apriori Algorithm. (2019*). Machine Learning and Big Data with Kdb /q*, 509–521. doi: 10.1002/9781119404729.ch26.

Vassakis, K., Petrakis, E., & Kopanakis, I. (2018). Big data analytics: Applications, prospects and challenges. *Mobile big data: A roadmap from models to technologies*, 3-20.

Vecchio, P. Del, Mele, G., Ndou, V., & Secundo, G. (2018). Creating value from Social Big Data: Implications for Smart Tourism Destinations. *Information Processing and Management*, 54(5), 847–860. https://doi.org/10.1016/j.ipm.2017.10.006.

Wang, G., & Liu, X. (2020). Predicting China's economic growth using machine learning and big data. *Applied Economics*, 52(28), 3053-3066. https://doi.org/10.1080/00036846.2020.1738687

Wang, H., Liu, J., Zhi, J., & Fu, C. (2013). The improvement of quantum genetic algorithm and its application on function optimization. *Mathematical problems in engineering*, *2013*.

Wang, J., Li, H., Yang, H., & Wang, Y. (2021). Intelligent multivariable air-quality forecasting system based on feature selection and modified evolving interval type-2 quantum fuzzy neural network. *Environmental Pollution*, *274*, 116429.

Wang, J., Zhang, W., Shi, Y., Duan, S., & Liu, J. (2018). Industrial big data analytics: challenges, methodologies, and applications. *arXiv preprint arXiv:1807.01016*.

Wang, Y., Hu, Z., Sanders, B. C., & Kais, S. (2020). Qudits and high-dimensional quantum computing. *Frontiers in Physics*, *8*, 589504.

Wani, M. A., Bhat, F. A., Afzal, S., & Khan, A. I. (2020). *Advances in deep learning*. Springer.

Wittek, P. (2014). *Quantum machine learning: what quantum computing means to data mining*. Academic Press.

Wu, J., & He, Y. (2021, May). Prediction of GDP in time series data based on neural network model. In *2021 IEEE International Conference on Artificial Intelligence and Industrial Design (AIID)* (pp. 20-23). IEEE.

Xie, G., Qian, Y., & Wang, S. (2021). Forecasting Chinese cruise tourism demand with big data: An optimized machine learning approach. *Tourism Management*, *82*, 104208.

Younas, M. (2019). Research challenges of big data. *Service Oriented Computing and Applications*, *13*, 105-107.

Yousefi, S., Derakhshan, F., & Karimipour, H. (2020). Applications of big data analytics and machine learning in the internet of things. *Handbook of big data privacy*, 77-108.

Yu, N., Li, Z., & Yu, Z. (2018). Survey on encoding schemes for genomic data representation and feature learning—from signal processing to machine learning. *Big Data Mining and Analytics*, *1*(3), 191-210.

Yu, Y., Si, X., Hu, C., & Zhang, J. (2019). A review of recurrent neural networks: LSTM cells and network architectures. *Neural computation*, *31*(7), 1235-1270.

Yurtoğlu, N. (2018). Türk standartları enstitüsünün (TSE) kuruluşu bağlamında Türkiye'de standardizasyon politikaları (1923-1960). *History Studies, C, 10*, 7.

Zeadally, S., Siddiqui, F., Baig, Z., & Ibrahim, A. (2020). Smart healthcare: Challenges and potential solutions using internet of things (IoT) and big data analytics. *PSU research review*, *4*(2), 149-168.

Zeng, J., & Glaister, K. W. (2018). Value creation from big data: Looking inside the black box. *Strategic Organization*, *16*(2), 105-140.

Zhang, G., Liu, Y., & Jin, X. (2020). A survey of autoencoder-based recommender systems. *Frontiers of Computer Science*, *14*, 430-450.

Zhang, Q., Yan, L., Hu, R., Li, Y., & Hou, L. (2022). Regional economic prediction model using backpropagation integrated with bayesian vector neural network in big data analytics. *Computational Intelligence and Neuroscience*, *2022*.

Zhang, X., Li, S., Li, X., & Li, L. (2020). A dynamic network analysis of China's economic growth. *Technological Forecasting and Social Change*, 154, 119968. https://doi.org/10.1016/j.techfore.2020.119968

Zhang, Y., & Ni, Q. (2020). Recent advances in quantum machine learning. *Quantum Engineering*, *2*(1), e34.

Zhou, L., Pan, S., Wang, J., & Vasilakos, A. V. (2017). Machine learning on big data: Opportunities and challenges. *Neurocomputing*, *237*, 350-361.

Zhou, X., Qiu, D., & Lou, L. (2023). Distributed exact quantum algorithms for Bernstein-Vazirani and search problems. *arXiv preprint arXiv:2303.10670*.

Zhou, Y., Cai, L., & Zhang, J. (2021). The effect of environmental regulation on industrial upgrading in China: Evidence from the "two control zones" policy. *Journal of Cleaner Production*, 307, 127341. https://doi.org/10.1016/j.jclepro.2021.127341

APPENDICES

Appendix I: Data Mapping Qiskit

```
for i in range(n_qubits):
    value_idx = i * (len(gdp_values) // n_qubits)
    gdp_subset = gdp_values[value_idx: value_idx + (len(gdp_values) // n_qubits)]
    average_gdp = np.mean(gdp_subset)
    theta = 2 * np.arcsin(np.sqrt(average_gdp / np.sum(gdp_values)))
    if np.isnan(theta):  # Handle NaN values
        theta = 0
    qc.ry(theta, qreg[i])

# Step 5: Quantum dataset generation
backend = Aer.get_backend('statevector_simulator')
transpiled_qc = transpile(qc, backend)
job = execute(transpiled_qc, backend, shots=1)
result = job.result()
quantum_statevector = result.get_statevector(transpiled_qc)

# Step 6: Data preprocessing for PCA
df_pca = pd.DataFrame(gdp_values)
imputer = SimpleImputer(strategy='mean')
df_pca_imputed = imputer.fit_transform(df_pca)

# Step 7: Dimensionality reduction using PCA
pca = PCA(n_components=1)  # Set n_components to 1
df_pca_reduced = pca.fit_transform(df_pca_imputed)
reduced_components = df_pca_reduced.flatten()
```

```
import numpy as np
import pandas as pd
from qiskit import QuantumCircuit, QuantumRegister, ClassicalRegister, transpile, Aer, execute
from qiskit.visualization import circuit_drawer, plot_bloch_multivector
from sklearn.decomposition import PCA
from sklearn.impute import SimpleImputer
import os

# Step 1: Data collection and preprocessing
data_file = 'Raw-Data/world_gdp.xlsx'
df = pd.read_excel(data_file)

# Step 2: Feature selection
df_kenya = df[df['Country Name'] == 'Kenya']
gdp_values = df_kenya.iloc[:, 4:].values.flatten()

# Step 3: Quantum representation selection
n_qubits = 4

# Step 4: Feature encoding using quantum circuits
qreg = QuantumRegister(n_qubits, 'q')
creg = ClassicalRegister(n_qubits, 'c')
qc = QuantumCircuit(qreg, creg)
```

```
# Step 9: Further quantum processing (optional)

# Print the final reduced components
print("Reduced Components:")
print(reduced_components)

# Create a folder for quantum data if it doesn't exist
output_folder = 'Quantum-Data'
os.makedirs(output_folder, exist_ok=True)

# Save the quantum statevector and reduced components to files
output_statevector_file = os.path.join(output_folder, 'quantum_statevector.xlsx')
output_reduced_components_file = os.path.join(output_folder, 'reduced_components.xlsx')
np.savetxt(output_statevector_file, quantum_statevector)
np.savetxt(output_reduced_components_file, reduced_components)

# Plot the quantum circuit
print("Quantum Circuit:")
circuit_drawer(transpiled_qc, output='text')

# Plot the quantum gate diagram
print("Quantum Gate Diagram:")
statevector = result.get_statevector()
plot_bloch_multivector(statevector)
```

Appendix II: QNN Model Source Code

```python
import numpy as np
import pandas as pd
from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import train_test_split
from sklearn.metrics import precision_score, recall_score, f1_score, mean_absolute_error, mean_squar
from qiskit import Aer, QuantumCircuit
from qiskit_machine_learning.algorithms import NeuralNetworkClassifier
from qiskit_machine_learning.connectors import TorchConnector
import torch
import torch.nn as nn
import torch.optim as optim
import matplotlib.pyplot as plt
import time

# Step 1: Capture data from Excel document
data = pd.read_excel('Quantum-ML/reduced_components.xlsx', header=0)

# Extract features and labels
features = data.iloc[1:, 0].values.astype(float)
labels = data.iloc[1:, 1].values.astype(float)

# Step 2: Preprocess the data
scaler = MinMaxScaler()
features_scaled = scaler.fit_transform(features.reshape(-1, 1))
labels_scaled = scaler.fit_transform(labels.reshape(-1, 1))
```

```python
# Step 3: Split the data into 70% training, 20% validation, and 10% testing
train_features, test_features, train_labels, test_labels = train_test_split(features_scaled, labels_
val_features, test_features, val_labels, test_labels = train_test_split(test_features, test_labels,

# Step 4: Convert data to PyTorch tensors
train_features_tensor = torch.tensor(train_features).float()
train_labels_tensor = torch.tensor(train_labels).float()
val_features_tensor = torch.tensor(val_features).float()
val_labels_tensor = torch.tensor(val_labels).float()

# Step 5: Define the neural network model
class NeuralNetwork(nn.Module):
    def __init__(self):
        super(NeuralNetwork, self).__init__()
        self.fc1 = nn.Linear(1, 64)
        self.fc2 = nn.Linear(64, 32)
        self.fc3 = nn.Linear(32, 1)
        self.activation = nn.ReLU()

    def forward(self, x):
        x = self.activation(self.fc1(x))
        x = self.activation(self.fc2(x))
        x = self.fc3(x)
        return x
```

```python
# Step 6: Train the neural network model
model = NeuralNetwork()
optimizer = optim.Adam(model.parameters(), lr=0.01)
criterion = nn.MSELoss()

num_epochs = 2000
losses = []  # Store the training loss values
val_losses = []  # Store the validation loss values
train_times = []  # Store the training times
val_times = []  # Store the validation times
accuracies = []  # Store the training accuracies
val_accuracies = []  # Store the validation accuracies

def calculate_accuracy(labels, predictions):
    threshold = 0.5
    binary_predictions = (predictions >= threshold).astype(int)
    return np.mean(binary_predictions == labels) * 100

for epoch in range(num_epochs):
    start_time = time.time()

    outputs = model(train_features_tensor)
    loss = criterion(outputs, train_labels_tensor)

    optimizer.zero_grad()
    loss.backward()
```

```python
    losses.append(loss.item())  # Append the training loss value to the list
    train_time = time.time() - start_time
    train_times.append(train_time)

    start_time = time.time()

    # Calculate validation loss
    val_outputs = model(val_features_tensor)
    val_loss = criterion(val_outputs, val_labels_tensor)
    val_losses.append(val_loss.item())

    val_time = time.time() - start_time
    val_times.append(val_time)

    # Calculate training accuracy
    train_predictions = model(train_features_tensor).detach().numpy()
    train_accuracy = calculate_accuracy(train_labels, train_predictions)
    accuracies.append(train_accuracy)

    # Calculate validation accuracy
    val_predictions = model(val_features_tensor).detach().numpy()
    val_accuracy = calculate_accuracy(val_labels, val_predictions)
    val_accuracies.append(val_accuracy)
```

```python
# Step 13: Print the training time and validation time
print("Training Time:", sum(train_times))
print("Validation Time:", sum(val_times))

# Step 14: Calculate evaluation metrics
threshold = 0.5
binary_predictions = (predictions >= threshold).astype(int)

mae = mean_absolute_error(test_labels, predictions)
mse = mean_squared_error(test_labels, predictions)
rmse = np.sqrt(mse)

print("Mean Absolute Error (MAE):", mae)
print("Mean Squared Error (MSE):", mse)
print("Root Mean Squared Error (RMSE):", rmse)

# Calculate Mean Absolute Percentage Error (MAPE)
mape = np.mean(np.abs((test_labels - predictions) / np.maximum(np.abs(test_labels), 1e-10))) * 100
```

```python
    if (epoch + 1) % 1 == 0:
        print(f"Epoch [{epoch+1}/{num_epochs}], Loss: {loss.item():.4f}, Val Loss: {val_loss.item():

# Step 7: Save the trained model
model_path = 'Quantum-ML/neural_network_model.pth'
torch.save(model.state_dict(), model_path)

# Step 8: Predict using the trained model
test_features_tensor = torch.tensor(test_features).float()
predictions_tensor = model(test_features_tensor)
predictions = predictions_tensor.detach().numpy()

# Step 9: Save the prediction results
results = pd.DataFrame({'Year': test_features.flatten(), 'GDP': test_labels.flatten(), 'Predicted GD
results.to_excel('Quantum-ML/prediction_results.xlsx', index=False)

# Step 10: Define the quantum circuit
num_qubits = 4
qc = QuantumCircuit(num_qubits)

# Step 11: Save the circuit image
image_path = 'Quantum-ML/circuit_image.png'
qc.draw(output='mpl', filename=image_path)
```

```python
# Calculate R-squared (R^2) coefficient
ssr = np.sum((test_labels - predictions) ** 2)  # Sum of squared residuals
sst = np.sum((test_labels - np.mean(test_labels)) ** 2)  # Total sum of squares
r2 = 1 - (ssr / sst)

# Calculate Forecast Bias
bias = np.mean(test_labels - predictions)

print("Mean Absolute Percentage Error (MAPE):", mape)
print("R-squared (R^2) Coefficient:", r2)
print("Forecast Bias:", bias)

# Step 16: Print the average training and validation accuracies
print("Training Accuracy Average:", np.mean(accuracies))
print("Validation Accuracy Average:", np.mean(val_accuracies))

# Step 17: Plot the training and validation accuracy graph
plt.plot(range(1, num_epochs+1), accuracies, label='Training')
plt.plot(range(1, num_epochs+1), val_accuracies, label='Validation')
plt.xlabel('Epoch')
plt.ylabel('Tmes')
plt.title('Training and Validation')
plt.legend()
plt.show()
```

Appendix III: Chuka University Introductory Letter

CHUKA UNIVERSITY

Knowledge is Wealth *(Sapientia divitiae est)* Akili ni Mali

OFFICE OF THE DIRECTOR
BOARD OF POSTGRADUATE STUDIES

Telephones: 020-2310512/18
Direct Line: 020-268 7625
postgraduate@chuka.ac.ke
P. O. Box 109-60400, Chuka
Website: www.chuka.ac.ke

REF: SM22/40012/2019

18th May, 2023

Director
National Commission for Science Technology and Innovation
Off Waiyaki Way, Upper Kabete
P O Box 30623, 00100
Nairobi.

Dear Sir / Madam,

Saif Kinyori

The above-named person is a *bona fide* student of Chuka University pursuing Master of Science in Computer Science proposal titled: **"Quantum – Enhanced Neural Network for Forecasting Kenyan Economic Growth."**

Mr. Kinyori has defended at the Faculty level and is now expected to conduct research. Any assistance accorded will be highly appreciated

Yours sincerely,

1° MAY 2023

Prof. Moses Muraya, Ph.D.

DIRECTOR
BOARD OF POSTGRADUATE STUDIES

Appendix IV: Ethics Review Letter

CHUKA UNIVERSITY

Knowledge is Wealth ***(Sapientia divitia est)*** **Akili ni Mali**

CHUKA UNIVERSITY INSTITUTIONAL ETHICS REVIEW COMMITTEE

Telephones: 020-2310512/18 P. O. Box 109-60400, Chuka

Direct Line: 0772894438 Email: info@chuka.ac.ke. Website: www.chuka.ac.ke

16th May, 2023

REF: CUIERC/ NACOSTI/374

TO: **Saif Kinyori**

RE: Quantum-Enhanced Neural Network for Forecasting Kenyan Economic Growth

This is to inform you that *Chuka University IERC* has reviewed and approved your above research proposal. Your application approval number is *NACOSTI/NBC/AC-0812*. The approval period is 16th May, 2023 – 16th May, 2024.

This approval is subject to compliance with the following requirements;

i. Only approved documents including (informed consents, study instruments, MTA) will be used

ii. All changes including (amendments, deviations, and violations) are submitted for review and approval by *Chuka University IERC*.

iii. Death and life threatening problems and serious adverse events or unexpected adverse events whether related or unrelated to the study must be reported to *Chuka University IERC* within 72 hours of notification

iv. Any changes, anticipated or otherwise that may increase the risks or affected safety or welfare of study participants and others or affect the integrity of the research must be reported to *Chuka University IERC* within 72 hours

v. Clearance for export of biological specimens must be obtained from relevant institutions.

vi. Submission of a request for renewal of approval at least 60 days prior to expiry of the approval period. Attach a comprehensive progress report to support the renewal.

vii. Submission of an executive summary report within 90 days upon completion of the study to *Chuka University IERC*.

Prior to commencing your study, you will be expected to obtain a research license from National Commission for Science, Technology and Innovation (NACOSTI) https://oris.nacosti.go.ke and also obtain other clearances needed.

Yours sincerely

Dr. Benjamin Kanga

SECRETARY

Chuka University is…… …Inspiring Environmental Sustainability for Better Life

Appendix V: NACOSTI License

REPUBLIC OF KENYA

NATIONAL COMMISSION FOR SCIENCE,TECHNOLOGY & INNOVATION

Ref No: 891358

Date of Issue: 17/June/2023

RESEARCH LICENSE

This is to Certify that Mr.. Saif Kinyori of Chuka University, has been licensed to conduct research as per the provision of the Science, Technology and Innovation Act, 2013 (Rev.2014) in Tharaka-Nithi on the topic: Qriantum-Enhanced Neural Network for Forecasting Kenyan Economic Growth for the period ending : 17/June/2024.

License No: **NACOSTI/P/23/26553**

891358

Applicant Identification Number

Director General

NATIONAL COMMISSION FOR SCIENCE,TECHNOLOGY & INNOVATION

Verification QR Code

NOTE: This is a computer generated License. To verify the authenticity of this document, Scan the QR Code using QR scanner application.

See overleaf for conditions