Quantum Machine Learning approaches to enhance Classification and Regression problems

1st Md Muballigh Hossain Bhuyain
Department of CSE
BRAC University
Dhaka, Bangladesh
md.muballigh.hossain.bhuyain@g.bracu.ac.bd
www.muballigh.xyz

2nd Jannatun Noor
Department of CSE
BRAC University
Dhaka, Bangladesh
jannatun.noor@bracu.ac.bd

3rd Shahrear Alam
Department of CSE
BRAC University
Dhaka, Bangladesh
shahrear.alam@g.bracu.ac.bd

4th Muhtasim Ahmed
Department of CSE
BRAC University
Dhaka, Bangladesh
muhtasim.ahmed@g.bracu.ac.bd

5th Md. Sakibur Rahman
Department of CSE
BRAC University
Dhaka, Bangladesh
md.sakibur.rahman3@g.bracu.ac.bd

***Abstract*—In this era of big data and machine learning, a huge quantity of data in the magnitude of trillions is processed every day, to make inferences. It is incumbent upon us to process this large aggregate of data for making precise and logical decisions. In order to process such a huge amount of data, large industry-grade components are required, which are very expensive and inaccessible to the masses. Thus, we are looking into the realm of Quantum Computers for better optimization of machine learning models. In this paper, we discuss how, classification and regression machine learning models can be optimized and made efficient by using quantum machine layer.**

***Index Terms*—quantum machine learning, classification, regression, algorithm, simulation, qiskit**

I. Introduction

The study of quantum machine learning integrates machine learning methods with quantum physics concepts. Quantum computers, which are capable of carrying out certain computations far more quickly than conventional computers, are meant to run quantum machine learning algorithms.

The ability to resolve some issues that classical machine learning algorithms find challenging or impossible to resolve is one of the key advantages of quantum machine learning. Quantum machine learning techniques, for instance, can analyze enormous volumes of data more quickly or identify patterns in data that are challenging to locate using conventional algorithms.

However, it's crucial to keep in mind that the subject of quantum machine learning is still young and expanding quickly, and there are numerous obstacles and constraints that must be surmounted in order to fully realize its potential. For instance, using quantum computers for machine-learning tasks is challenging because they are still in their infancy and prone to errors. A lot of research is still being done to create new quantum machine learning algorithms and comprehend their potential and constraints.

II. Hybrid Quantum Layer

In quantum machine learning algorithms, hybrid quantum layers are typically used to process data and perform various machine learning tasks. They are intended to take advantage of quantum computers' unique capabilities, such as their ability to perform certain calculations much faster than classical computers, while also leveraging classical computing's strengths, such as their ability to handle large amounts of data efficiently. A form of artificial neural network layer known as a hybrid quantum layer combines aspects of classical and quantum computing. In a hybrid quantum layer, some calculations are carried out using quantum algorithms and others using conventional ones.

One of the key advantages of hybrid quantum layers is their ability to enhance machine learning algorithms' performance by allowing them to process data more quickly and identify patterns in data that are challenging to find with conventional techniques. Hybrid quantum layers are a fast expanding field of study, but it's vital to keep in mind that there are still a lot of obstacles to be overcome before their full potential can be realized. .

III. Related Works

In the following journal [6] The authors set up a new area of research that has developed around applying machine learning techniques to take advantage of quantum devices and process classical data.. They provide a general review of quantum machine learning in comparison to classical methods in their work. They talked about the many technological contributions, advantages, and resemblances of the research activity in this field, departing from the fundamental ideas of machine learning and quantum computing. They also go into detail on the complexity of various quantum machine learning algorithms, their recent developments, and their use in a variety of disciplines like physics, chemistry, and natural language processing. The authors discuss an interdisciplinary field called quantum machine learning (Quantum ML) that combines machine learning with quantum physics (ML).In this symbiotic relationship, the capability of quantum computers is employed to develop quantum versions of ML algorithms and to use conventional ML algorithms to study quantum systems. More specifically, quantum machine learning is a discipline in which machine learning tasks are performed using the capabilities of quantum computers and the characteristics of quantum physics, and then the results are applied to related domains. Strange and counterintuitive patterns that cannot be produced by any conventional computer are frequently created by quantum systems. Quantum computers would be able to create unique and counterintuitive patterns as well as recognize patterns that no traditional computer can. When employing a quantum machine learning problem, there are numerous levels of complexity. The theory of computational complexity is concerned with the cost and scalability of algorithms on both a general and problem-specific level." Scalability" is the term used to describe the cost in terms of time and/or space needed to increase the volume or complexity of the computation's objective."An algorithm that is O(n 3) is regarded as "harder" than one that is O(n 2) using the Big-O notation because, regardless of how rapidly these operations are carried out., the former will often require more operations to be affected than the latter.If an algorithm with an O complexity is available, a task is considered to be solvable if it can be completed in polynomial time (n p). If not, it is assumed that the issue is not polynomial. Considering that quantum algorithms can maintain a superposition of all the states of a particular system and select a specific state from a list using just one operation, they are also faster than their conventional equivalents. On a traditional machine, O(n) operations are necessary to do the same task. They used quantum machine learning to solve a variety of issues, including regularization, the shor's factorization algorithm, the quantum fourier transform, and linear equations. For each of these tactics, training times will often be lowered from O(n 3) of conventional approaches to O(n 2) while virtually preserving the trained estimator's statistical efficiency.Though the size of contemporary data sets is constantly expanding, the temporal complexity of the order of O(n 2) can still be too difficult for practical applications. In this context, quantum computing may offer the chance to further boost these technologies' effectiveness, allowing them to be significantly scaled up. In fact, by using a range of quantum algorithms for linear algebra, sampling, and optimization methods, they may theoretically achieve speedups that are up to an order of magnitude faster than those achieved by conventional techniques. Also the quick memory access and intricate data structures required by current QML approaches may limit their usefulness in practical situations. The modeling of high-temperature superconductors, choosing molecules for the construction of organic batteries, and modeling and testing of drugs are just a few of the specialized scientific problems that quantum computing can assist in resolving. The actual number of gates needed to implement an algorithm in QML is currently unknown, assuming it even exists at all. The intricacy of these methods in terms of integration is currently just theoretical because they are simply conceptual. It should be noted that although quantum computing shows significant potential in terms of effectiveness and scalability in comparison to traditional computing, it is still unclear whether this can be fully realized in practice. In fact, it is a widely held belief that a classical Turing machine can solve any issue that can be solved using the quantum computing paradigm. In addition, rather than focusing on quantum phenomena, there are many open concerns about how to apply quantum computing to data coming from non-quantum contexts that are common in computer science and consumer applications.

In this paper [7] the author briefly discuss about machine learning (ML) appears to be one of the promising "killer" applications for quantum computing technologies as they get closer to the period of commercialization and quantum supremacy .They suggested putting more emphasis on regions where ML researchers are struggling rather than the well-liked and more manageable supervised learning techniques, such as generative models in unsupervised and semi-supervised learning. They also draw attention to the situation where quantum models might be better appropriate for classical datasets with potential quantum-like statistical correlations .They concentrated on hybrid quantum-classical techniques and highlighted some of the major obstacles they see in front of them in the near future. In this article, they offered and emphasized three strategies to increase the likelihood of discovering game-changing applications for near-term quantum computers: 1. Concentrate on issues that the ML community currently finds difficult and intractable. 2.Focus on datasets with potential intrinsic quantum-like correlations, which makes quantum computers necessary. These will provide the most compact and efficient model representation, and even at the level of 50-100 qubit devices, there might be a sizable quantum advantage. 3. The problematic step of the conventional ML algorithmic pipeline should focus on hybrid algorithms that execute a quantum routine. With 16×16 continuous valued pixels and 10 binary variables representing the class in0, , 9 ¼ . artificial data was produced by a QAHM implemented on the D-Wave 2000Q and trained on a subsampled version of the MNIST dataset. 266 visible variables and two layers of 120 and 60 hidden variables each make up the recognition and generator networks.By first employing the D-Wave 2000Q to sample the deepest layer, and then processing the signals through the conventional generator network component, the samples are created from the final model. The D-Wave 2000Q quantum annealer, which has 1644 qubits, is used in these investigations. It would be challenging for these algorithms to maintain the reported speedup because they inherit the Harrow-Hassidim-Lloyd (HHL) algorithm's restrictions, as was mentioned in .The authors mainly focus shifts to the implementation of algorithms with possible quantum advantages in near-term quantum devices, they raised the bar even higher in this instance .Some of the issues that will typically impact any QAML algorithm include the limited qubit connectivity, the finite dynamic range of the parameters determined by the intrinsic energy scale of the interactions in the device, as well as intrinsic noise in the gadget causing decoherence in the qubits and uncertainty in the programmable parameters. They focused on a few of these practical difficulties, paying particular attention to the execution and application of hybrid QAML algorithms in short-term hardware. The hybrid quantum-classical ML framework known as the quantum-assisted Helmholtz machine (QAHM) has the capacity to operate with real-world datasets. Here they propose to describe the most abstract representation of the data, i.e., the deepest layers of the generator network, using a quantum device. Starting with the raw input, the recognition network (left) performs a bottom up step to infer hidden variables. Either a quantum device or a classical layer (near-term) yields the most abstract representation (future implementations). From samples collected from a quantum device, the generator network creates samples of the observable variables via a top-down pass. When quantum hardware is characterized using a gray-box approach,An implicit density model is the end outcome. Two hidden layers, each with 120 and 60 variables, were present in both networks, totaling 266 variables that could be seen.. They were a classical recognition network and a quantum-assisted generator network. In DWave 2000Q, the deepest layer of 60 variables was mapped to 1644 qubits. They used the quantum-assisted generator network to create samples while running the wake-sleep algorithm for 1000 iterations. The fake data frequently resembles numbers typed by people, despite the fact that these preliminary findings cannot compare to state-of-the-art ML. They observed that the simulated images produced by the network are not simply copies of the training set, but rather, in certain cases, contain variances and uniqueness, reflecting the model's capacity for generalization. While the synthetic data might also appear hazy, other strategies are also impacted by this issue. While the synthetic data might also appear hazy, other strategies are also impacted by this issue. Only recently developed GANs have produced artificial images that are noticeably crisper. Their innovative method attempts to address some of the most urgent problems associated with processing industrial-scale datasets that contain a sizable number of continuous variables. It is driven by the notion that, after removing the data that can be handled traditionally, only the more abstract representation of the data should be handled by a quantum computer.

In this paper [8], the authors mainly discuss the future of ML. This column has highlighted the technologies being studied as potential candidates to reboot computers for the past two years.This article makes the claim that a three-technology hybrid computing method could produce answers to a wide range of issues that are sufficiently improved that energy efficiency will no longer be the primary priority.We need to express quantum computing in ways that don't needlessly downplay the importance of programming in order to illustrate how it works with machine learning. The neighborhood has looked into two possibilities to lower the price of number factoring : 1)The subexponential number field sieve algorithm was created by mathematicians, computer scientists, and programmers over the course of about 100 person-years. 2)Shor's polynomial-time quantum algorithm was created, but it needs a yet-to-bebuilt quantum computer. The development of quantum algorithms coincided with advancements in their conventional counterparts, creating competition between the 100 personyears of study and the unique characteristics of quantum information. On the other hand Quantum computers are unfairly disadvantageous because computational complexity theory seeks the optimum algorithm without taking algorithm development or programming effort into account. Logic gate placement on an integrated circuit. Logic gates are placed on a chip's surface by chip design tools' optimizers with just enough room between them to accommodate the wire that specifies the chip's function. On the other hand Logic gates are placed on a chip's surface by chip design tools' optimizers with just enough room between them to accommodate the wire that specifies the chip's function. A vision for future applications, the author has created a scenario in which current business applications, such as route optimization for transportation, are enhanced and subsequently used in routine personal circumstances. Quantum tunneling is a technique used in one type of quantum machine learning. With the breadth of the peak decreasing the probability of this happening exponentially. Societal implications, despite the fact that we currently use computers as agents for many of our commercial activities and information management, the author treated future computers as though they were isolated devices.

According to this paper [9], Authors discuss botnets - Devices that a hacker can remotely manipulate are referred to as the Botnet. The word & quot ; botnet vquot ; refers to the interaction of robots with networks, in which the & quot ; botmaster r & quot; and "bot slave & quot are two key players. By sending commands from the botmasters to the bot clients to act as slaves to the bot master, the botnet # 39;s job is to launch attacks. Peer-to-peer network attacks using botnets have gotten more difficult because it is difficult to locate the command and control center. In a peer-to-peer network, the peer-to-peer attack is carried out by sending botnet attacks from one system to another, whereas the command-and-control attack is carried out by a botmaster attack on a server that engages in various exchanges with systems on the network, with those systems acting as slave nodes in the network. Through the use of malware programs, a botnet is a type of malware that impairs both the functionality and access to a secure computer system.The authors set up botnet programs carry out phishing, spam, and DDoS attacks. In a DDoS attack, the attacker, known as the botmaster, uses powerful computers and servers to run command and control malware programs that give instructions to the machines at the next level, known as handlers. When applying the BARCA framework, it has been seen that metrics correlations and a feedback-based approach are preferred to using historical dataBARCA has three main components which are Behaviour extractor (BE), Behaviour Identifier (BI) and a Feedback Provider (FP). . The research's suggested technique for detecting and stopping IoT botnet attacks achieved 98.89% accuracy, 99.01% precision, 98.74% recall score, and a f1-score of 98.87%

The authors of this review [10] conducted a survey of various quantum methods used to address learning problems. While some promising results have been obtained, the current literature does not yet provide sufficient evidence to conclude that quantum techniques can offer an exponential advantage in realistic learning settings. Even in cases where quantum algorithms for linear algebra have rigorous guarantees, practical issues such as data access and restrictions on the types of problems that can be solved may impact their performance. To fully evaluate the potential of these techniques, it will be necessary to make advances in quantum hardware development. Most of the literature on quantum machine learning (QML) has been produced within the quantum community, and it is believed that further progress in the field will require greater collaboration between the quantum and machine learning communities. To appeal to both groups of researchers, the authors structured this review to emphasize the computational aspects of machine learning. While this perspective allows for an efficient discussion of quantum algorithms designed to accelerate runtime compared to classical counterparts, it is important to note that statistical problems like determining the generalization performance of an algorithm are also relevant. Some papers of interest have been excluded from this review, and readers are advised to refer to other sources for a more comprehensive review. The authors discussed the computational cost as a major challenge for the future of machine learning, and how quantum algorithms can reduce the complexity of certain regularization methods. They divided quantum approaches into four main categories: linear algebra, neural networks, sampling, and optimization. Quantum algorithms based on linear algebra show the greatest potential for computational advantages, but it is uncertain whether their memory access requirements will allow them to speed up the analysis of classical data. Quantum methods for training neural networks, sampling, and optimization have mostly provided quadratic advantages and may be implementable on first-generation quantum computers, but their theoretical foundations are not yet well established. In summary, the works reviewed in this article, including the theoretical evidence presented, suggest the possibility of a quantum speed-up for some machine learning problems, but the extent and impact of these speed-ups on practical problems is still unknown. The authors identified several promising areas for future research, including exploring the relationship between noise, generalization performance, and hardness in a quantum context, improving our understanding of how quantum resources impact sample and time complexity, determining whether large quantum random access memory (QRAM) devices can be built, and understanding if non-polynomial machine learning problems can be efficiently solved using quantum computers.

In this research [11], the authors used quantum circuits to represent data in quantum Hilbert spaces, which function as feature spaces for the data. They developed two strategies, inspired by kernel theory, to identify patterns in the data. The first approach involves estimating intractable quantum kernels on a quantum device and using them with a classical kernel method. The second approach involves using quantum models based on variational circuits to learn models that process the feature vectors. These simple models benefit from incorporating non-linearity into the process of encoding inputs into a quantum state or quantum feature map. The authors believe that combining quantum computing with kernel theory will lead to the creation of quantum machine learning algorithms for near-term quantum devices that may offer potential quantum speed-ups.

This article [12] reviews some of the current research on quantum machine learning, specifically focusing on quantum versions of supervised and unsupervised machine learning algorithms based on the quantum circuit model. While some quantum versions of machine learning algorithms have been shown to provide a speedup over their classical counterparts, there are still issues in this area, such as the problem of data input and output. In some cases, reading classical data may dominate the cost of quantum algorithms, preventing a speedup at the macro level, and it may be infeasible to exactly read out the data for certain learning tasks. As a result, it is believed that applying quantum computing directly to quantum data is more effective and efficient than using it on classical data. Therefore, quantum machine learning is more suited to solving problems within quantum systems. However, with the continued development of research and theory, it is possible that in the near future these quantum machine learning algorithms will be able to effectively help people solve data and decision-making problems

IV. Challenges of Quantum Machine Learning

When it comes to quantum machine learning, both academics and practitioners encounter a number of difficulties (QML). Several of these difficulties include:

- **Hardware limitations:** One of the biggest problems with QML is how little quantum hardware is actually available. Despite having the ability to execute some jobs significantly more quickly than classical computers, quantum computers are still in their infancy and are not yet commonly used. This restricts the scope of QML experiments and impedes advancement in this area.
- **Noise and mistakes:** Quantum computers are also susceptible to noise and mistakes, which can have a substantial impact on the accuracy of QML algorithms. This is particularly true for quantum computers with additional qubits because the quantity of mistakes rises as qubits are added.
- **Lack of quantum algorithms:** Although some quantum algorithms have been created for QML, the discipline is still in its infancy and there aren't many well-established quantum algorithms that can be used for a variety of jobs.
- **Uncertainty:** It is currently unclear how to employ quantum computers for machine learning tasks effectively. It is also unclear which machine learning activities are best suited for quantum computers.

- **Lack of big-scale data:** The ability of traditional machine learning algorithms to process massive volumes of data is one of their strengths. However, the difficulty quantum computers now face in managing vast amounts of data restricts their applicability to several machine learning applications.

Despite these difficulties, QML has the power to transform machine learning and address issues that are currently beyond the scope of traditional algorithms. To address these issues and develop the QML discipline, researchers and practitioners are working hard.

V. Boosting Classification and Regression Models

The field of quantum machine learning (QML) blends machine learning methods with quantum computers. It intends to enhance the efficiency of machine learning algorithms by utilizing the special characteristics of quantum systems. Using quantum computers to carry out some calculations more quickly than conventional computers is one method that QML can speed up classification and regression models. For instance, a quantum computer is substantially quicker than a classical computer at carrying out specific linear algebra operations like singular value decomposition and matrix inversion. These procedures are frequently applied in machine learning methods like principal component analysis and support vector machines. Using quantum algorithms, which can solve optimization problems more quickly than conventional algorithms, is another method that QML can enhance the performance of machine learning algorithms.

Finding the ideal parameters to fit a model to a dataset is only one example of how optimization is used in many machine learning techniques. These optimization issues might be solved more quickly by quantum algorithms like quantum annealing and quantum gradient descent than by conventional algorithms. It's crucial to remember that QML's potential advantages are primarily theoretical at this point, and it's unclear how much quantum computing will be able to outperform classical computing in machine learning applications.

Our major objective is to improve classical ML models with quantum machine processing. The classical data will be processed appropriately by the quantum model. We will require a quantum model to process the classical input in a quantum manner. The two main groups into which quantum models can be split are deterministic and variable quantum models. When we measure our quantum system using a deterministic quantum model, we can design an algorithm or model where we can predict the result with certainty. The Deutsch-Jozsa algorithm is one illustration of this. A ket-vector (complex column vector) that is undergoing an operation (U) is the initial parameter in this situation. An example of this operation would be a matrix multiplication or another matrix function. The outcome of the system's measurement is 'y' at the end. The drawback of these models is that they frequently exhibit significant noise. Which implies that they are prone to interference and occasionally produce inaccurate outcomes.

The most recent generation of quantum models are variational quantum models. Different from its differential counterparts, variational quantum models provide stochastic results as opposed to deterministic results. Thus, these models are repeatedly run to obtain data regarding the result (expectation value). Quantum support vector machines and quantum neural networks are examples of variational quantum models, which are more comparable to many conventional machine learning models. A significant number of quantum machine learning models also fit into this category.

Although there are some significant distinctions, the deterministic model's operating principle is relatively similar. These variational models rely on some variables that we can adjust and modify. As a result, the operations, U, are parameterized, and 'y' represents the model's stochastic output. These variational quantum models are superior because they are less vulnerable to noise and more unlikely to produce inaccurate results.

Since accessing a quantum machine requires queuing a job in the IBM quantum machine servers, the objective is to use a quantum simulator rather than a quantum machine. Modern quantum computers are also highly noisy and tiny. As a result, each test that is done will have a random quantity of noise. Additionally, because other people also use the quantum computers, one must line up, and the wait periods can vary. There is a maximum period permitted before someone else can acquire access to the quantum computer once you do. We have chosen to employ a simulator that will provide us with a noise-free outcome and mimic an ideal or almost ideal scenario for the future, when quantum computers have advanced significantly.

VI. Quantum Algorithms

Quantum algorithms are algorithms that make use of quantum mechanical phenomena, such as superposition and entanglement, to perform operations on data. These algorithms can potentially solve certain problems much faster than classical algorithms, although they are often more difficult to implement and can be sensitive to noise and other errors.

A. Deutsch's algorithm

It is one of the first quantum algorithms to be developed and is often used as a simple example to illustrate the principles of quantum computing. The problem that Deutsch's algorithm is designed to solve is the following: Given a function f(x) that takes a single input and returns either 0 or 1, determine whether the function is constant (always returns the same value) or balanced (returns 0 and 1 equally often). This problem can be solved using a classical computer in a single evaluation of the function, but Deutsch's algorithm is able to solve it using just one evaluation of the function in the quantum case. An outline of how Deutsch's algorithm works:

- Initialize a quantum register with two qubits in the state —00>.

- Apply a quantum gate to the first qubit that puts it in the state —+>. This is a superposition state that is equal to a linear combination of —0> and —1>.
- Apply a quantum gate to the second qubit that puts it in the state —->. This is also a superposition state that is equal to a linear combination of —0> and —1>.
- Apply a quantum gate to both qubits that applies the function f(x) to both qubits. This gate is called the ”oracle,” and it is the key to Deutsch’s algorithm.
- Apply a quantum gate to both qubits that performs a controlled-NOT operation. This flips the state of the second qubit if the first qubit is —1>, but leaves it unchanged if the first qubit is —0>. Measure both qubits. If the function is constant, the measurement will always yield —00>. If the function is balanced, the measurement will yield either —01> or —10> with equal probability.

Deutsch’s algorithm illustrates the power of quantum computers to solve certain problems more efficiently than classical computers.

B. Deutsch–Jozsa algorithm

This is a generalization of Deutsch’s algorithm and is used to determine whether a function is constant or balanced when the function takes multiple inputs. Like Deutsch’s algorithm, the Deutsch–Jozsa algorithm is designed to solve the following problem: Given a function f(x) that takes multiple inputs and returns either 0 or 1, determine whether the function is constant (always returns the same value) or balanced (returns 0 and 1 equally often). This problem can be solved using a classical computer, but the Deutsch–Jozsa algorithm is able to solve it using fewer evaluations of the function in the quantum case. An outline of how the Deutsch–Jozsa algorithm works:

- Initialize a quantum register with n+1 qubits, where n is the number of inputs to the function f(x).
- The first n qubits are initialized to —0¿ and the last qubit is initialized to —1>.
- Apply a quantum gate to the first n qubits that puts them in a superposition state. This is a linear combination of all possible input states for the function f(x).
- Apply a quantum gate to the n+1 qubits that applies the function f(x) to the first n qubits and stores the result in the last qubit. This gate is called the ”oracle,” and it is the key to the Deutsch–Jozsa algorithm.
- Apply a quantum gate to the n+1 qubits that performs a controlled-NOT operation. This flips the state of the last qubit if any of the first n qubits are —1¿, but leaves it unchanged if all of the first n qubits are —0>.
- Measure the last qubit. If the function is constant, the measurement will always yield —0>. If the function is balanced, the measurement will yield —1>.

C. Simon’s algorithm

It is also known as Simon’s periodicity algorithm is used to find hidden periodicities in a function, which is a problem that is believed to be difficult for classical computers. The problem that Simon’s algorithm is designed to solve is the following: Given a function f(x) that takes a single input and returns a single output, find the period of the function if it is periodic (i.e., if there exists an integer p such that f(x) = f(x+p) for all x). The function f(x) is promised to be periodic, but the period p is not known and may be very large. An outline of how Simon’s algorithm works:

- Initialize a quantum register with n qubits, where n is the number of bits needed to represent the period p.
- Apply a quantum gate to the n qubits that puts them in a superposition state. This is a linear combination of all possible input states for the function f(x).
- Apply a quantum gate to the n qubits that applies the function f(x) to the first qubit and stores the result in the second qubit. Repeat this process for each pair of qubits, so that the function f(x) is applied to the first qubit and the result is stored in the second qubit, the function f(x) is applied to the second qubit and the result is stored in the third qubit, and so on. This gate is called the ”oracle,” and it is the key to Simon’s algorithm.
- Apply a quantum gate to the n qubits that performs a controlled-NOT operation. This flips the state of the second qubit if the first qubit is —1>, but leaves it unchanged if the first qubit is —0>.
- Repeat this process for each pair of qubits. Measure the n qubits. The measurement will yield a string of bits that represents the period p.

D. Grover’s algorithm

It is also known as Grover’s search algorithm, is used to search through a database more efficiently than classical algorithms. The problem that Grover’s algorithm is designed to solve is the following: Given a database of N items, find the item that satisfies a certain property (called the ”marked” item) by making as few queries to the database as possible. This problem is known as the ”unstructured search problem,” and it is believed to be difficult for classical computers to solve. An outline of how Grover’s algorithm works:

- Initialize a quantum register with n qubits, where n is the number of bits needed to represent the size of the database N.
- Apply a quantum gate to the n qubits that puts them in a superposition state. This is a linear combination of all possible input states for the function f(x), where x is the index of the item in the database.
- Apply a quantum gate to the n qubits that applies the function f(x) to the first qubit and stores the result in the second qubit. Repeat this process for each pair of qubits, so that the function f(x) is applied to the first qubit and the result is stored in the second qubit, the function f(x) is applied to the second qubit and the result is stored in the third qubit, and so on. This gate is called the ”oracle,” and it is the key to Grover’s algorithm. The function f(x) is designed to return 1 if the item at index x is the marked item and 0 otherwise.
- Apply a quantum gate to the n qubits that performs a controlled-NOT operation. This flips the state of the

second qubit if the first qubit is —1>, but leaves it unchanged if the first qubit is —0>. Repeat this process for each pair of qubits.
- Measure the n qubits. The measurement will yield a string of bits that represents the index of the marked item in the database.

Grover's algorithm is an example of a quantum algorithm that provides a quadratic speedup over classical algorithms.

VII. Workflow and Simulations

A. Quantum Simulation

Quantum simulation is a powerful tool for studying the behavior of quantum systems, as well as classical systems under certain conditions. It allows researchers to study the properties of quantum systems and their behavior under different conditions, which can be difficult to do using traditional methods.

One of the main reasons quantum simulation is necessary is because quantum systems are often difficult to study directly. Quantum systems are highly sensitive to their environment, and this sensitivity can make it difficult to control and measure their behavior. Quantum simulation allows researchers to study quantum systems in a controlled environment, using a quantum computer or quantum-based model.

In addition, quantum simulation can be used to study the behavior of quantum systems at a very small scale, such as the behavior of atoms or molecules. Traditional methods, such as experiments or classical simulations, may not be able to accurately capture the behavior of quantum systems at this scale, making quantum simulation an important tool for understanding these systems.

Finally, quantum simulation can be used to study the behavior of classical systems under certain conditions, such as at very low temperatures or in extreme environments. This can be useful for understanding the behavior of materials and for developing new technologies.

Overall, quantum simulation is a powerful tool for studying the behavior of quantum and classical systems and is an important tool for researchers and developers working in a variety of fields

B. Qiskit Quantum Simulations

Qiskit is an open-source quantum computing software development framework for working with quantum computers. It provides tools for creating and manipulating quantum circuits, running them on simulated quantum computers, and connecting to real quantum processors via the cloud. One of the key features of Qiskit is its quantum simulation capability. Quantum simulation is the use of a quantum computer or quantum-based model to study the behavior of another physical system. This can be used to study the behavior of quantum systems, such as atoms or molecules, or to study the behavior of classical systems under certain conditions.

Qiskit provides a number of tools for performing quantum simulation, including the QuantumCircuit class for constructing quantum circuits and the Aer quantum simulator for running those circuits on a classical computer. It also includes tools for analyzing the results of quantum simulations, such as the QuantumCircuit.measure() method for measuring the state of qubits in a circuit and the QuantumCircuit.draw() method for visualizing quantum circuits. Overall, Qiskit is a powerful tool for working with quantum computers and quantum simulations, and is widely used by researchers and developers in the quantum computing community.

C. Running QML models using Qiskit

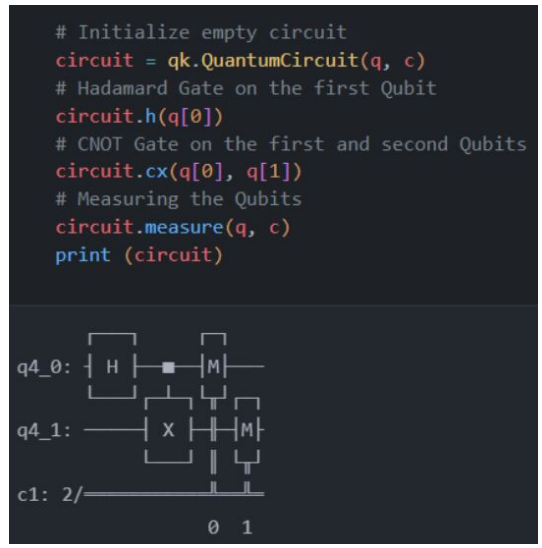

Fig. 1. Constructing quantum circuit

We have used a quantum feature map and qiskit, we compute a kernel matrix that we then feed into the scikit-learn classification and clustering methods. After Importing necessary libraries, creating Qbits and constructing quantum circuit (Fig. 1) we will go forward to classification.

D. Classification

We'll use the scikit-learn support vector machine classification (svc) technique and the adhoc dataset from Supervised learning with quantum augmented feature spaces for our classification example (Fig. 2). We build up the "FidelityQuantumKernel" class to compute a kernel matrix using the ZZFeatureMap once we have our training and testing datasets ready. We employ the ComputeUncompute fidelity and the reference implementation of the Sampler primitive to determine state overlaps. These are the default settings, and the same objects will be generated for you if you don't give a Sampler or a "Fidelityinstance" parameter. We have two options for creating a custom kernel with the scikit-learn SVC algorithm: either we provide the kernel as a callable function or we precompute the kernel matrix. Using the FidelityQuantumKernel class in qiskit, we can carry out either of these. The training and testing kernel matrices (Fig. 3) are computed and plotted beforehand before being supplied to the scikit-learn svc algorithm.

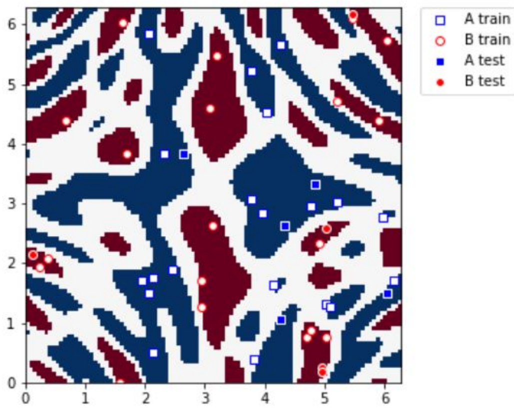

Fig. 2. Adhoc dataset for classification. The train and test segments of A and B are visible.

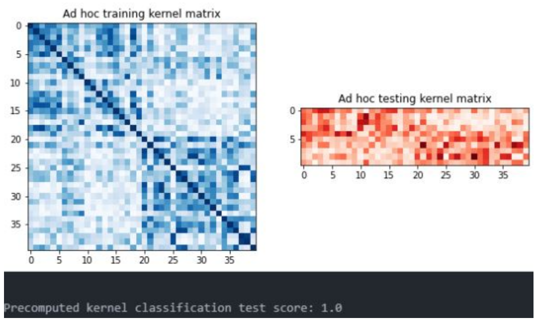

Fig. 3. Training and Testing Kernel Matrices of Ad hoc dataset

E. Clustering

We will once more utilize the ad hoc dataset from Supervised learning with quantum augmented feature spaces along with the scikit-learn spectral clustering algorithm for our clustering example (Fig. 4). Since clustering is an unsupervised machine learning job, we renew the dataset with a wider difference between the two classes and do not require a test sample. The FidelityQuantumKernel class is once more configured to compute a kernel matrix using the ZZFeatureMap and default values this time. In order to define a unique kernel for the scikit-learn spectral clustering algorithm, we can either precompute the kernel matrix (Fig. 5) or provide the kernel as a callable function. We can only use the latter with Qiskit Machine Learning's FidelityQuantumKernel class. Given that we already know the class labels, the following code scores the labels using normalized mutual information before passing them to the scikit-learn spectral clustering algorithm.

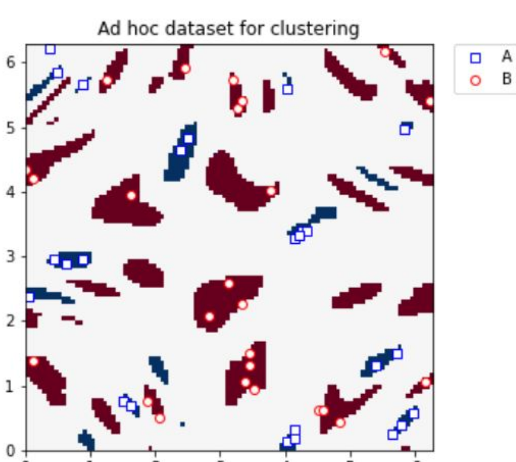

Fig. 4. Clustering of Ad hoc dataset

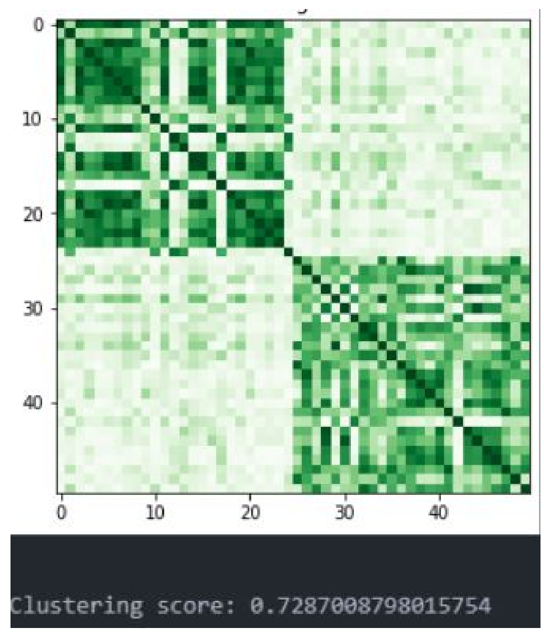

Fig. 5. Clustering of Ad hoc clustering kernel matrix

VIII. Conclusion

In this age of big data and machine learning, billions of bytes of data are analyzed each day to draw conclusions. It is our responsibility to digest this vast collection of facts so that we may make informed choices. Big industry-grade components are needed to analyze such a large volume of data, but these are exceedingly expensive and out of reach for most people. To improve the optimization of machine learning models, we are thus investigating the field of quantum computers. In this research, we explain how quantum machine layer may be used to improve and increase efficiency of classification and regression machine learning models. It's important to remember that quantum machine learning is a very new and rapidly evolving field, and that there are many challenges and limitations that must be overcome in order to fully realize its promise. For example, it is difficult to employ quantum computers for machine-learning tasks since they are still in their infancy and prone to mistakes. To develop new quantum machine learning algorithms and appreciate their potential and limitations, a lot of research is still being done.

References

[1] Cerezo, M., Verdon, G., Huang, H.-Y., Cincio, L., & Coles, P. J. (2022, September 15). Challenges and opportunities in quantum ma-

chine learning. Nature News. Retrieved December 31, 2022, from https://www.nature.com/articles/s43588-022-00311-3
[2] Huang, H.-Y., Broughton, M., Mohseni, M., Babbush, R., Boixo, S., Neven, H., & McClean, J. R. (2021, May 11). Power of data in Quantum Machine Learning. Nature News. Retrieved December 31, 2022, from https://www.nature.com/articles/s41467-021-22539-9
[3] Carlo, C., Carlo Ciliberto Carlo Ciliberto Department of Computer Science, Ciliberto, C., Carlo Ciliberto Department of Computer Science, Herbster, M., Mark Herbster Department of Computer Science, Ialongo, A. D., Alessandro Davide Ialongo Department of Engineering, Pontil, M., Massimiliano Pontil Department of Computer Science, Rocchetto, A., Andrea Rocchetto http://orcid.org/0000-0002-5216-0095 Department of Computer Science, Severini, S., Simone Severini Department of Computer Science, Wossnig, L., & Leonard Wossnig Department of Computer Science. (2018, January 17). Quantum Machine Learning: A classical perspective. Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences. Retrieved December 31, 2022, from https://royalsocietypublishing.org/doi/full/10.1098/rspa.2017.0551
[4] Broughton, M., Verdon, G., McCourt, T., Martinez, A. J., Yoo, J. H., Isakov, S. V., Massey, P., Halavati, R., Niu, M. Y., Zlokapa, A., Peters, E., Lockwood, O., Skolik, A., Jerbi, S., Dunjko, V., Leib, M., Streif, M., Von Dollen, D., Chen, H., … Mohseni, M. (2021, August 26). TensorFlow quantum: A software framework for Quantum Machine Learning. arXiv.org. Retrieved December 31, 2022, from https://arxiv.org/abs/2003.02989
[5] Broughton, M., Verdon, G., McCourt, T., Martinez, A. J., Yoo, J. H., Isakov, S. V., Massey, P., Halavati, R., Niu, M. Y., Zlokapa, A., Peters, E., Lockwood, O., Skolik, A., Jerbi, S., Dunjko, V., Leib, M., Streif, M., Von Dollen, D., Chen, H., … Mohseni, M. (2021, August 26). TensorFlow quantum: A software framework for Quantum Machine Learning. arXiv.org. Retrieved December 31, 2022, from https://arxiv.org/abs/2003.02989
[6] Khan, T. M., & Robles-Kelly, A. (2020). Machine Learning: Quantum vs Classical. IEEE Access, 8, 219275–219294. https://doi.org/10.1109/access.2020.3041719
[7] Perdomo-Ortiz, A., Benedetti, M., Realpe-Gómez, J., & Biswas, R. (2018). Opportunities and challenges for quantum-assisted machine learning in near-term quantum computers. Quantum Science and Technology, 3(3), 030502. https://doi.org/10.1088/2058-9565/aab859
[8] DeBenedictis, E. P. (2018). A Future with Quantum Machine Learning. Computer, 51(2), 68–71. https://doi.org/10.1109/mc.2018.1451646
[9] Shinan, K., Alsubhi, K., Alzahrani, A., & Ashraf, M. U. (2021). Machine Learning-Based Botnet Detection in Software-Defined Network: A Systematic Review. Symmetry, 13(5), 866. https://doi.org/10.3390/sym13050866
[10] Ciliberto, C., Herbster, M., Ialongo, A. D., Pontil, M., Rocchetto, A., Severini, S., & Wossnig, L. (2018). Quantum machine learning: a classical perspective. Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences, 474(2209), 20170551. https://doi.org/10.1098/rspa.2017.0551
[11] Schuld, M., & Killoran, N. (2019). Quantum Machine Learning in Feature Hilbert Spaces. Physical Review Letters, 122(4). https://doi.org/10.1103/physrevlett.122.040504
[12] Zhang, Y., & Ni, Q. (2020). Recent advances in quantum machine learning. Quantum Engineering, 2(1). https://doi.org/10.1002/que2.34