Design of Quantum Machine Learning Course for a Computer Science Program

Sathish Kumar
Cleveland State University
Cleveland, OH, USA 44115
s.kumar13@csuohio.edu

Temitope Adeniyi
Cleveland State University
Cleveland, OH, USA 44115
t.adeniyi@vikes.csuohio.edu

Ahmad Alomari
Cleveland State University
Cleveland OH, 44145, U.S.A.
a.alomari@vikes.csuohio.edu

Santanu Ganguly
Photonic Inc.,
London, UK
santanu.gangoly@gmail.com

***Abstract*—In this work, we present the design and plan of Quantum machine learning (QML) course in a computer science (CS) University program at senior undergraduate level / first year graduate level. Based on our survey, there is a lack of detailed design and assessment plan for the delivery of QML course. In this paper we have presented the QML course design with week by week details of QML concepts and hands on activities that are covered in the course. We also present how this QML course can be assessed from CS program learning outcomes perspective.**

Keywords—Quantum machine learning; Quantum computing; Course Assessment; Course Design; QML Workflow.

I. Introduction And Background

Quantum computing is a radically different approach to classical computing and promises a more efficient problem-solving approach compared to the existing approaches [7]. The innovative progressions of the last decade have made quantum PCs a more practical possibility, and many accept the possibility that the quantum computing could become reality in the following 10 years, if not less. If so, the accompanying impacts in the Information processing and decision making will be enormous. Most by and large, organizations should carry out quantum innovations into their analytics and decision-making frameworks. Also, there is a critical need to equip the workforce with the quantum skills to implement the quantum revolution that is envisioned. The big technology organizations such as Google, NVIDIA, Microsoft, IBM, Amazon and HP are heavily investing in quantum technologies and quantum computing research. As a result, there are needs of graduates who have a solid foundation in quantum data science. With the anticipated growth in quantum computing needs, the skill needs will expand rapidly.

Quantum computing involves much more than quantum processing [1, 2, 33]. Quantum data science has the potential to become mainstream data analytic technology and it is important to equip the next generation of data scientists and the Artificial Intelligence professionals with the Quantum machine learning techniques. Quantum machine learning can be defined as the process of integrating machine learning and quantum computing for high-performance data processing, modeling, and learning. The role of QML is to enhance the performance and evaluation processes of machine learning models inspired by quantum mechanics [3]. On the other hand, machine learning techniques also can be implemented to describe quantum error-correcting codes, measure quantum systems properties, and develop new quantum algorithms [4]. There are several recent works that highlight the importance and opportunities of Quantum machine learning [5, 6, 10, 21, 22]. The importance of QML is highlighted by the availability of the books related to QML including the one written by the co-author of this work, Ganguly [31].

The contribution of this work are as follows: We outline a framework for the successful implementation of the quantum machine learning course that can be offered to senior level undergraduate and first year graduate students. By identifying, addressing, assessing and promoting workforce development solutions for the quantum machine learning skill gap, the proposed course will serve as an educational foundation for preparing the next generation of the data scientists and the Artificial Intelligence professionals. In addition, the proposed QML course will equip quantum computing and quantum physicists with QML skills.

The rest of the paper is organized as follows. Section II explains the related work in the literature as well as the survey of existing QML courses. Section III provides the motivation of creating the QML course. Section IV outlines the design of the proposed QML course along with the description of the QML concepts and QML practical hands on exercises that will be covered. Section V explains the assessment plan for the proposed QML course. Section VI outlines the future work recommendations and finally section VII concludes the paper with concluding remarks.

II. Related Work and Existing QML Courses

As per our literature review, there are quite a few papers related to Quantum Computing education course development [8, 9, 11, 27-29]. Mykhailova presented a paper on the delivery of programming assignments [27] for teaching quantum computing. Temporao et al., presented a case study of how quantum computing can be taught without prerequisites [28]. Considering that the field of Quantum computing is evolving, Wootton et al., described in their paper [29], how quantum computing can be taught using interactive textbook. M. Piattini analyzes the existing international curriculums as well as some global training programs to illustrate the quantum computing education needs. This related work also defines a quantum computing specialization that can be included within the existing international curriculums [8]. C.L. Friedrichs et al., has looked into the analysis of Quantum Computing Courses World Wide [9]. In their study they analyze the textbooks that

979-8-3503-4323-6/23/$31.00 ©2023 IEEE
DOI 10.1109/QCE57702.2023.20326

are being used for Quantum computing. They also surveyed the type of textbook, the department that teaches the Quantum Computing course , circuit simulators that are being used in these courses and the companies that have shown interest in the Quantum computing students. O. Salehi et al., in their work presents a computer science (specifically quantum programming tool as an educational tool) oriented approach to introduce Quantum Computing to the students [11]. They use the concept of generalized mathematical and statistical concepts (algebra and probability theory) rather than a field emanating from physics to teach the quantum computing to the students. The tools that the students will use include Jupyter notebooks for coding and analysis, along with the Qiskit framework by IBM and the Pennylane framework by Xanadu [11].

As per our survey, to our knowledge there is only one related work on teaching Quantum Machine Learning course [30]. That work is more related to research training of Quantum machine learning and most of the effort in the related work is spent on describing about Quantum Machine Learning, Data Embedding rather than the detailed design and delivery of the course using the theoretical concepts and the programming assignments. Also, that work did not deal with the assessment plan for the QML course.

As per our quick survey, while there are classical machine learning courses offered as an elective course in quantum graduate program, we could not see offering of a QML course at the University settings. While looking at the Massively Open Online Course (MOOC) platforms, Udemy offers a Quantum machine learning as part of Quantum computing and Quantum machine learning course and EdX offers Quantum Machine Learning course in partnership with the University of Toronto. However, these courses are dated and does not have detailed coverage of QML topics.

III. Motivation

While there is a heavy need for graduates with classical computing skills in CS, IS and IT, there is a critical shortage of quantum computing and quantum machine learning professionals who can research, develop and maintain quantum-based machine learning models to address the scientific and business needs [12].

The Global Quantum Computing market is expected to grow from USD 472 million in 2021 to USD 1,765 million by 2026, at a CAGR of 30.2% [13]. There are also excellent commercial aspects of Quantum Computing [24].

The governments across the world recognizes the importance of the Quantum computing education /research and has invested in millions of dollars in the Quantum computing recently [23, 25]. For example, The US National Quantum Initiative [17], Quantum Canada [14], Europe's Quantum Flagship initiative [15], Quantum information science and technology in Japan [18], Quantum technologies in Russia [19], Charting the Australian quantum landscape [16], and Quantum information research in China [20].

A quick survey at the Quantum Machine Learning related online job boards reveal that the following QML related jobs are available currently [26] that requires expertise in QML: Sr. Research Scientist Quantum AI and ML; Quantum Computational Scientist; Quantum Research Data Scientist; Quantum Computing Computer Scientist; Applied Researcher Consultant; Principal Solutions Architect, Quantum Computing; Sr. Software Engineer, Quantum AI; and Quantum Solutions Scientist.

The proposed course address majority of the skills that are required by the above-mentioned job titles and succeed in the Industry, Academia and Government agencies. The course will provide the necessary theoretical and applied practical skills that will help the prospective graduate from this course, succeed in the job of applying quantum computing principles for the data science/AI/ML related problems. The proposed course will prepare graduates to build end to end quantum-based learning models by enabling them to apply quantum mechanics and data science concepts to solve real world problems.

IV. QML Course Design

The course can be offered as a 10 to 15-week program. In this work, we chose to design it for a 14-week program. Following Table 1 shows the topics that will be covered in each week. We plan to offer this course for a senior level undergraduate / first year graduate student. The pipeline shown in the following Figure 1 will be used in the course for the student training purposes especially for the practical exercises and assignments.

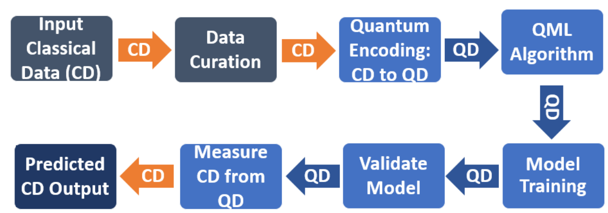


Fig. 1 – The QML Pipeline (CD–Classical Data; QD-Quantum Data)

Table 1 – The QML Course Design

Week	Theoretical Concepts and Practical Hands On Exercises Covered
1	Math Basics and Overview of Classical ML
2	Introduction to Quantum Computing – Gates; Operators and Circuits
3	Introduction to Quantum Computing – States; Superposition; Bloch Sphere; Entanglement and Measurement
4	Quantum Algorithms (Shor's Algorithm; Grover Algorithm and Deutsch-Jozsa Algorithm)
5	Introduction to Quantum Machine Learning
6	Quantum Data Encoding and Representation
7	Quadratic Unconstrained Binary Optimization and Quantum Approximate Optimization Algorithm
8	Variational Quantum Circuits
9	Quantum Support Vector Machines and Kernel Methods
10	Quantum K-Means and K-Medians Clustering
11	Quantum Neural Networks
12	Quantum Generative Models
13	Quantum Reinforcement Learning
14	Hybrid Quantum Classical Machine Learning

Students will be expected to have a basic programming, statistics and probability background. Week 1 to 4 will cover the basics of quantum computing, mathematics, programming tools/infrastructure and classical machine learning that is needed for the students to be successful in the course. Several works exist in the literature on how these modules can be designed [8, 9, 11, 27-29]. In the rest of the sub-sections, we describe the design of QML related topics from theoretical and practical exercises perspective.

a Week 5:Introduction to Quantum Machine Learning:

In week 5, we introduce the fundamental concepts of QML, where we provide a structured overview with practical exercises that sets the stage for students to delve deeper into specific QML approaches for the later weeks in the course.

Theoretical concepts: We first provide a clear explanation of QML's primary objective, principles, and potential application. This centers around the development of quantum applications that harness the immense computational capabilities of quantum computers in conjunction with the scalability and learning capacity of machine learning algorithms [36], [37]. We will follow by presenting a comprehensive taxonomy of approaches for QML, categorizing the different methodologies and algorithms. This serves as a foundational framework to help students understand the underlying principles and techniques employed in various QML approaches. By systematically classifying QML algorithms into distinct categories, such as quantum-inspired algorithms, quantum circuit-based algorithms, and quantum kernel methods[38], students gains a holistic understanding of the different strategies and their unique advantages and limitations. The topic concludes with exploring the process and workflow involved in developing QML algorithms.

Practical Exercise: The following practical exercises are assigned in Week 5. *Exercise 1:* Involves installing all necessary libraries in Qiskit. *Exercise 2:* This exercise guides student to explore how quantum circuits can be designed to perform machine learning operations and the steps involved, with Qiskit to represent quantum model for a QML algorithm as shown in Figure 3. *Exercise 3:* Using the workflow as shown in Figure 2 and a synthetic dataset of numerical datapoints the students will experiment with the designed QML model to classify the dataset.

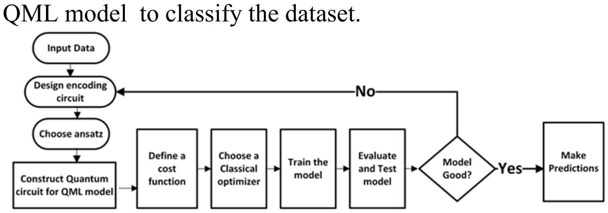


Fig 2: Workflow for designing a QML Algorithm

```
Designing a QML Algorithm

#Step1: we prepare a dataset, X as input data and y as the corresponding
label
X = input_data, y = label

#Step 2: Construct Quantum Circuit for QML model
def QML_Circuit(qubit,gates):
  ................................. #insert code here
#step 3: Chose a feature map
feat_map = PauliFeatureMap(feature_dimension=X.shape[1])
#step 4: Choose an ansatz
ansatz = RealAmplitudes(num_qubits=featuremap.num_qubits,
entanglement='linear', reps=1)
#step 5: Define a cost function
cost_function = ........................#insert code here
#step 6: choose a classical optimizer
optimizer = COBYLA(maxiter = 100)
#step 7: Create a quantum circuit
backend = Aer.get_backend('statevector_simulator')
quantum_instance = QuantumInstance(backend)
QML_model = QML_circuit(qubit, gates)
model = QML_model(optimizer, feat_map, ansatz, cost_function,
quant_instance)
#step 8: Train the model
train_data, train_labels = split_dataset_to_data_and_labels(X, y)
model.train(train_data, train_labels)
#step 10: Iterate and refine, adjust feature_map, ansatz or optimization
strategy as necessary
................................#insert code here
print("Accuracy:", accuracy)
```

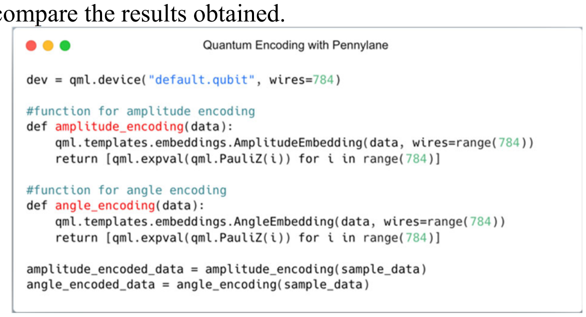
Fig 3: Sample code for the exercise where students will design a simple QML model. "Insert code here" is where each student input their own custom codes

b Week 6: Quantum Data Encoding and Representation:

Week 6 theoretical concepts and practical exercises explore the intricacies of representing and manipulating data in the context of QML.

Theoretical Concepts: Students will learn about the fundamental principles and techniques involved in representing classical data in the quantum form [39]. As shown in Figure 5, various approaches will be discussed, such as quantum feature maps [40], where the classical data is mapped onto a quantum feature space and also methods for encoding classical data into quantum states such as amplitude encoding, angle encoding, and quantum circuit-based encoding [41]. Then we also describe how these encoded data is used to train QML models. The techniques for quantum measurement and extracting classical information from quantum states will also be taught, in this they explore measurement operators and measurement bases, understanding how to retrieve classical data from quantum systems [42]. By comprehending the advantages and limitations of each technique, they will be better equipped to choose appropriate data representation and encoding methods for different QML tasks [43].

Practical Exercise: Students will reinforce their knowledge with the following practical exercises. *Exercise 1:* We provide the students with the Mnist dataset [44], then guide them in encoding using functions provided by the Pennylane programming framework [45]. They also evaluate the performance of each encoding method in a QML task. *Exercise 2:* Students implement a quantum feature map using Qiskit [46]. Once the quantum feature map is implemented, they encode the classical data samples from the Iris dataset [47] into quantum states using a feature map circuit and then run the circuit on the IBM quantum simulator. Finally, they analyze the encoded quantum states by measuring and comparing relevant metrics using the code as shown in Figure 4. *Exercise 3:* Students design and implement measurement circuits to extract classical information from a prepared quantum state. They compare the extracted information with the original classical data to evaluate the fidelity or accuracy of the information extraction process. Further, the students will experiment with different measurement strategies and compare the results obtained.



```
Quantum Encoding with Pennylane

dev = qml.device("default.qubit", wires=784)

#function for amplitude encoding
def amplitude_encoding(data):
    qml.templates.embeddings.AmplitudeEmbedding(data, wires=range(784))
    return [qml.expval(qml.PauliZ(i)) for i in range(784)]

#function for angle encoding
def angle_encoding(data):
    qml.templates.embeddings.AngleEmbedding(data, wires=range(784))
    return [qml.expval(qml.PauliZ(i)) for i in range(784)]

amplitude_encoded_data = amplitude_encoding(sample_data)
angle_encoded_data = angle_encoding(sample_data)
```

Fig 4: Code illustrating Quantum Encoding

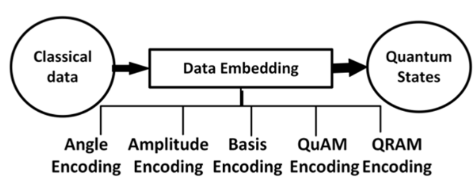


Fig 5: Quantum State preparation process

c. Week 7: Quantum Approximate Optimization Algorithm and Quadratic Unconstrained Binary Optimization

Quantum Approximate Optimization Algorithm: In this week, we provide students with a solid understanding on QAOA.

Theoretical Concepts: We start by introducing and defining the fundamental idea behind QAOA. We present it as a general technique used to find an approximate solution to combinatorial optimization problems [72]. Students learn about the structure of QAOA, we also briefly introduce the quantum variational principle and the role of parameterized quantum circuits in optimization tasks. Other theoretical aspects that will be covered will include understanding the QAOA ansatz, the optimization landscape, and the role of the quantum-classical interface in solving optimization problems [73]. We will also provide the performance, challenges, and limitation of the QAOA and its industry applications in optimization tasks. These concepts will help student to gain experience in applying QAOA to solve real-world combinatorial optimization problems.

Practical Exercise: Following are the practical exercises that will be covered, which will enable student to have hands-on experience. *Exercise 1:* In this exercise, students will experiment with the quantum circuit that implements the QAOA using the Qiskit programming framework, as shown in Figure 5 and design a parameterized quantum circuit, that represents the QAOA ansatz [74]. *Exercise 2:* In exercise 2, we will guide the students through the steps involved in using QAOA to reveal approximate solutions to an optimization problem. The exercise will focus on solving a specific combinatorial optimization problem, such as the MaxCut problem [75], [76], whose sample circuit is as shown in Figure 6. *Exercise 3:* In this exercise, the students will explore the effects of varying the number of QAOA layers and the choice of variational parameters on QAOA performance. They will experiment with different numbers of layers and optimize the variational parameters using classical optimization techniques or built-in optimization routine.

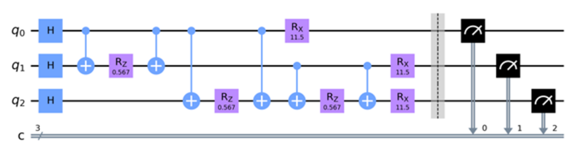


Fig 6: Sample circuit for implementing QAOA applied to the MAXCUT problem.

```python
from pennylane import qaoa

cost_h, mixer_h = qaoa.min_vertex_cover(graph, constrained=False)
def qaoa_layer(gamma, alpha):
    qaoa.cost_layer(gamma, cost_h)
    qaoa.mixer_layer(alpha, mixer_h)
def circuit(params, **kwargs):
    for w in wires:
        qml.Hadamard(wires=w)
    qml.layer(qaoa_layer, depth, params[0], params[1])

dev = qml.device("qulacs.simulator", wires=wires)
@qml.qnode(dev)
def cost_function(params):
    circuit(params
    return qml.expval(cost_h)
optimizer = qml.GradientDescentOptimizer()

steps = 70
params = np.array([[0.5, 0.5], [0.5, 0.5]], requires_grad=True)
for i in range(steps):
    params = optimizer.step(cost_function, params)

@qml.qnode(dev)
def probability_circuit(gamma, alpha):
    circuit([gamma, alpha])
    return qml.probs(wires=wires)
probs = probability_circuit(params[0], params[1])
```

Fig 7: Sample algorithm for QAOA implemented on Pennylane.

Quadratic Unconstrained Binary Optimization: In addition to QAOA, in this week we will introduce QUBO for QML concepts to the students as well. The goal is to provide theoretical concepts and practical exercises that enhance students understanding of how QUBO will help optimize QML.

Theoretical Concepts: First, we will explain the fundamental concept of optimization problems. Then, we will introduce students to QUBO, which involves optimizing a quadratic objective function subject to binary variables [77, 78]. Finally, we will discuss the advantages and disadvantages of using QML optimization algorithms, such as quantum annealing and Quantum Approximate Optimization Algorithm (QAOA) to solve the Ising problem as a QUBO problem [79, 80, 81].

Practical exercises: *Exercise 1:* In exercise 1, we will introduce students to the process of encoding the Ising problem as QUBO. We will also Guide them in converting the Ising problem objective function into an equivalent QUBO formulation using binary variables and quadratic terms. In this exercise, we will help the students understand the correspondence between the Ising problem and the QUBO representation through the implementation. *Exercise 2:* In this exercise 2, we will ask students to use quantum annealing and QAOA to find an optimal solution for the Ising problem [79, 80, 81]. We will help them understand the principles and limitations of different solvers and interpret the results obtained through this exercise.

d. Week 8: Variational Quantum Circuits:

In this week, we provide students with a comprehensive introduction to VQC, establishing a solid foundation for understanding of this powerful quantum computing paradigm.

Theoretical Concept: Students will learn how VQCs leverage the flexibility and adaptability of parameterized quantum circuits to tackle optimization and QML tasks [48]. They will learn about the architecture and structure of VQCs, which consist of parametrized quantum circuits where the parameters are optimized to minimize a cost function [49]. The concept of variational optimization and gradient-based methods will be introduced, enabling students to understand how these techniques are applied to optimize the parameters of VQCs. Additionally, students will explore the role of ansatz design, which involves selecting appropriate quantum gates and their arrangement within the circuit to perform specific QML operations. Through the theoretical discussions, students will gain insights into the underlying principles of VQCs and their potential applications in solving machine learning problems. We will also delve into the applications of VQCs in various domains, highlighting their ability to address complex computational problems.

Practical Exercise: The assigned practical exercises for Week 7 concepts are as follows. *Exercise 1:* The students will learn how to construct and implement a VQC for QML tasks. Figure 8 provides a step-by-step guide on how the students will build a parameterized quantum circuit using the Qiskit as shown in the code in Figure 9. The student also learns how to choose suitable gate sequences and define variational parameters to create a flexible and trainable quantum circuit

[50]. *Exercise 2*: The students will be given a simple quantum machine learning task, such as quantum data classification that the students will be asked to use their designed VQC in exercise 1 to tackle the given task. *Exercise 3:* Students will be provided with sets of pre-designed VQC and a specific optimization objective, and they will be tasked with implementing gradient-based optimization techniques to optimize the circuits and achieve the best possible performance for the given objective.

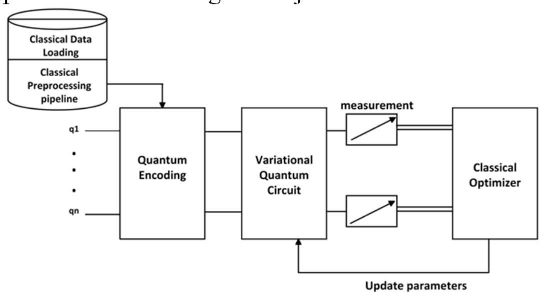


Fig 8: A Classifying Protocol using VQC.

```
VQC Algorithm

feat_map = ZZFeatureMap(fetaure_dimension=feature_dim, reps=2)
optimizer = SPSA(maxiter=40, c0=4.0, skip_calibration=True)
var_form = TwoLocal(feature_dim, ['ry', 'rz'], 'cz', reps =3)
vqc = VQC(optimizer, feat_map, var_form, train_input, test_input,
datapoints[0])

backend = BasicAer.get_backend('qasm_simulator')
quant_inst = QuantumInstance(backend, shots=1024, seed_simulator=seed,
seed_transpiler=seed)

result =vqc.run(quant_inst)
```

Fig 9: Code snippet showing VQC used as a QML model.

e. Week 9: Quantum Support Vector Machine (QSVM):

In week 9, we introduce the topic of QSVM to students through theoretical concepts and practical exercises that cover the components of QSVMs.

Theoretical Concepts: First, we discuss that QSVMs are an emerging application in the field of quantum machine learning and how they build upon classical Support Vector Machines (SVMs) but leverage the principles of quantum mechanics for advanced computation [3, 32]. Then we describe how QSVMs utilize quantum superposition and entanglement, which enable quantum qubits to encode and manipulate data in quantum states [3]. Utilizing QML workflow in Figure 1, we illustrate the QSVMs workflow including the steps of data encoding, quantum circuit training, and classification. Then we explain how classical data is encoded into quantum states using encoding techniques, highlighting the quantum algorithms involved. Then we discuss how QSVMs leverage quantum interference and optimization techniques to find the optimal hyperplane in the feature space. Finally, we illustrate the training phase, where the parameters of the quantum circuit are iteratively adjusted to minimize classification errors and maximize class separation to provide efficient classification [32, 34].

Practical exercises: We plan to assign the following practical exercises to the students covering Week 8 concepts. *Exercise 1*: As shown in Figure. 10, we will provide students with a simplified QSVM circuit and guide them in coding it using a quantum programming framework, such as Qiskit using the sample code as shown in Figure 11. As an illustrative assignment, students will encode a Distributed Denial of Service (DDoS) attacks dataset and perform classification based on the trained circuit to classify the data as benign and DDoS attacks [35]. *Exercise 2:* In the second exercise, students will experiment with different quantum feature maps and observe their impact on QSVM performance. *Exercise 3:* In the third exercise, students will train a QSVM circuit with varying parameters. Then they will evaluate the classification accuracy and separation of classes to understand the impact of different QSVM configurations. Finally, students will compare the performance of the QSVM with a classical SVM.

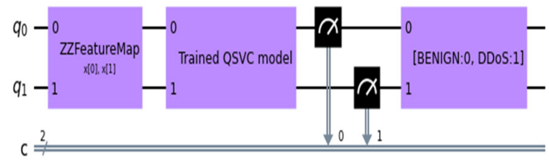


Fig 10. Example of a QSVM circuit for detecting DDoS attacks.

```
feature_map =  ZZFeatureMap(feature_dimension=2, reps=2, entanglement='linear')

qsvm = QSVM(feature_map, training_input, test_input, datapoints[0])

backend = BasicAer.get_backend('statevector_simulator')
quantum_instance = QuantumInstance(backend, shots=5000, seed_simulator=seed, seed_transpiler=seed)

#Results on my classsical computer
result2 = qsvm.run(quantum_instance)
kernel_matrix = result2['kernel_matrix_training']

print("predicted class: {}".format(result2['predicted_classes']))
print("accuracy:        {}".format(result2['testing_accuracy']))
```

Fig 11. Code illustrating QSVM components.

Figure. 10, shows the QSVM circuit that students will be asked to implement. Figure. 10 describes part of the QSVM circuit code that the students will be developing using Qiskit. The code shows the components of the QSVM, how to run it using a quantum circuit simulator, and how to generate the results.

f. Week 10: Quantum K-Means and K-Medians Clustering:

Theoretical concepts and practical exercises cover how quantum computing can be leveraged to solve clustering problems more efficiently and potentially discover hidden patterns in complex datasets.

Theoretical Concept: We discuss the theoretical foundation of Quantum K-Means and K-Medians clustering algorithms[51], [52]. We then teach how these quantum-inspired algorithms adapt classical clustering methods to the quantum realm, taking advantage of the unique properties of quantum systems. The topic will also cover key concepts, including quantum distance metrics, centroids, and the optimization of clustering objectives. They will understand how quantum gates and quantum circuit architectures can be designed to optimize the clustering results. We illustrate how by manipulating quantum states and applying quantum operations one can exploit the computational advantages of quantum systems to achieve more accurate and efficient clustering [53]. We also show how quantum circuit parameters are systematically fine-tuned to minimize the distance between data points and cluster centroids, enabling efficient clustering [54].

Practical Exercise: Following practical activities will be offered during this week. *Exercise 1:* We provide the students with numerical datapoints and guide then on how to implement quantum K-means and K-medians algorithms to cluster the data points into predefined K clusters using circuits generated using Qiskit as shown in Figure 12.

Exercise 2: We provide a customer segmentation dataset and then guide the students to apply Quantum K-Means and K-Medians Clustering algorithms to group customers based on their similarities [55]. The dataset is first preprocessed, and then the quantum clustering algorithm is implemented using Qiskit to identify distinct customer segments based on their shared characteristics as shown in the sample code in Figure 13. They evaluate the performance of the algorithm by comparing it with classical clustering techniques. *Exercise 3:* Students explore the robustness of quantum K-means and K-medians clustering algorithms in handling noisy data. We provide them with a generated synthetic dataset and added noise. The goal is to investigate how well the quantum clustering algorithms can handle such noisy data compared to classical clustering methods.

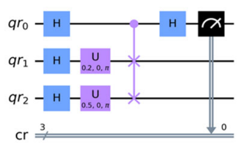


Fig 12: A quantum circuit for Quantum K-means model

Quantum K-means Circuit

```
theta_list = np.linspace(0, np.pi, num = 3)
qr = QuantumRegister(5, name="qr")
cr = ClassicalRegister(5, name="cr")
qc = QuantumCircuit(qr, cr, name="k_means")

# compute the distance between each pair of points
for i in range(9):
    for j in range(1, 10 - i):
        # Set the parameter theta for different points
        theta_1 = theta_list[i]
        theta_2 = theta_list[i + j]
        # Create the quantum circuit using qiskit
        qc.h(qr[2])
        qc.h(qr[1])
        qc.h(qr[4])
        qc.u(theta_1, pi, pi, qr[1])
        qc.u(theta_2, pi, pi, qr[4])
        qc.cswap(qr[2], qr[1], qr[4])
        qc.h(qr[2])
        qc.measure(qr[2], cr[2])
        qc.reset(qr)
        backend = Aer.get_backend('qasm_simulator')
        job = execute(qc, backend=backend, shots=1024)
        result = job.result()
```

Fig 13: Code snippet implementing Quantum k-means.

g. Week 11: Quantum Neural Networks:

In this week, we introduce QNNs by providing theoretical concepts and practical exercises covering all the essential components of QNNs. The goal is to provide students with the knowledge and skills to understand and implement QNNs effectively.

Theoretical Concepts: First, we discuss the motivation behind QNNs, which aim to leverage quantum phenomena to enhance classical Neural Networks (NNs) [65]. Then, we describe the architecture of QNNs that consist of quantum layers and explain how they will utilize quantum gates to process quantum states. Finally, we illustrate the concepts of entanglement, unitary matrix multiplications, and quantum activation functions, which are used in quantum layers to perform computations on quantum states.

Practical Exercises: *Exercise 1:* As shown in Figure 14, we intend to provide students with a QNN structure and guide them in coding it using Qiskit. The students will experiment with different quantum gates and activation functions to observe their impact on QNN behavior. *Exercise 2:* In exercise 2, students will train a QNN using a quantum variant of backpropagation or other optimization techniques. We will provide the students with the DDoS attacks dataset and guide them in evaluating the QNN classification performance. *Exercise 3:* For exercise 3, we will generate variations of the DDoS dataset by introducing noise, class imbalances, or outliers. For this exercise, we will task students with adapting the QNN architecture or training process to handle these variations and observe the impact on the model's robustness. Exercise 4: For exercise 4, we will develop a quantum activation function and guide the students in implementing it [66]. Students will experiment with the effect of the quantum activation function on the performance of the QNN compared to classical activation functions.

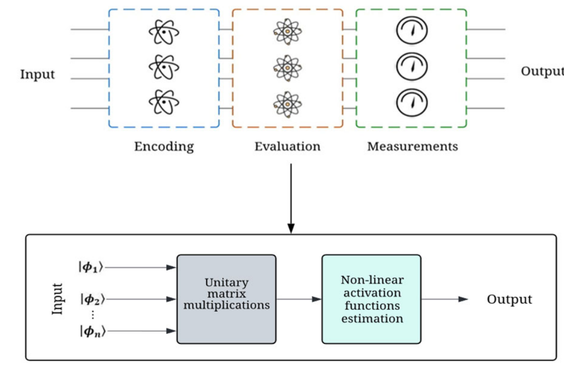


Fig 14. Example of a QNN.

Figure 14, shows the QNN approach that students will be asked to implement Here, the input represents the DDoS attacks dataset that will be encoded into quantum states in the encoding layer. The evaluation layer represents the unitary matrix multiplications and the estimation of the nonlinear activation functions. The results of the evaluation layer will be fed forward to the measurement layer to learn the model to predict future data [65, 66].

h. Week 12: Quantum Generative Models:

The theoretical concepts and practical exercise introduced in week 10 covers QGM and error mitigation techniques for quantum generative algorithms.

Theoretical Concept: We explore the theory behind quantum generative models, briefly introducing them to the mathematical foundations of quantum probability theory and its implications in generative modeling. Major approaches will be discussed, such as quantum generative adversarial networks (qGAN) [56], quantum Boltzmann machines (qBM) [57], and quantum variational autoencoders (qVAE) [58]. These models enable the generation of quantum data by learning from existing datasets or by sampling from a learned probability distribution. We discuss how these quantum-inspired algorithms adapt classical generative modeling techniques to the quantum domain, and how they are developed using quantum circuits. Additionally, students are introduced to quantum noise and its impact on generative algorithms [59], [60]. Here they learn about various sources of noise in quantum systems and explore techniques for noise mitigation [61]–[64]. By understanding the effects of quantum noise and how to mitigate them, students can develop robust and reliable generative models in the presence of noise.

Practical Exercise: We present following practical exercises for the students to gain a hands-on experience on the theoretical concept taught. *Exercise 1:* In exercise 1, we will guide student on the task of creating a qGAN model using the sample code as shown in Figure 16 for generating synthetic image of human faces. They design a quantum circuit in Qiskit to serve as the generator as shown in Figure 15 and another quantum circuit as the discriminator. *Exercise 2:* We guide students to implement a qVAE for data compression and reconstruction tasks. They utilize the theoretical concepts of quantum encoding and decoding covered in the week 6 to design a quantum circuit that acts as the encoder and decoder for the qVAE. They train the qVAE using molecular structures dataset for compression and reconstructing new sample. *Exercise 3:* Students explore the implementation of a qBM for unsupervised learning tasks. They develop a quantum circuit that represents the energy function of the qBM and use Qiskit to train the model using contrastive divergence learning algorithm. Students experiment with different numbers of visible and hidden units in the qBM to observe the effect on the model's learning capacity and performance. They apply the trained qBM for pattern recognition financial dataset consisting of correlated currency pairs.

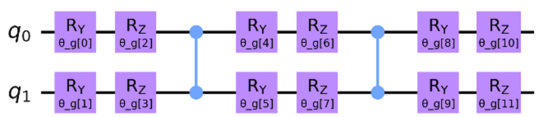


Fig 15: A sample ansatz for a generator in a qVAE

Variational Quantum Generator

```
generator = TwoLocal(2,
                     # Parameterized single qubit rotations
                     ['ry', 'rz'],
                     'cz',  # Entangling gate
                     'full', # Entanglement structure: all to all
                     reps=2, # Number of layers
                     parameter_prefix='θ_g',
                     name='Generator')
generator = generator.decompose() # decompose into standard gates
generator.draw()
```

Fig 16: Code snippet for the generator circuit in a qVAE

i. Week 13: Quantum Reinforcement Learning:

In this week, we will introduce QRL concepts to the students by providing theoretical concepts and practical exercises covering the essential components of QRL.

Theoretical Concepts: First, we will discuss the motivation behind QRL, which aims to use the power of quantum computation to address classical reinforcement learning problems more efficiently [69]. Then, we will explain the elements of QRL and how its states and actions differ from the states and actions in classical reinforcement learning. Then, we will emphasize that QRL explores quantum strategies and utilize quantum algorithms to enhance classical reinforcement. Finally, we will illustrate the workflow of QRL, highlighting the key components and steps involved [69, 70]. Then we explain the quantum representation of states and actions, where qubits encode the information and quantum gates manipulate the quantum states. Then we discuss the role of quantum algorithms in QRL, such as quantum amplitude amplification and quantum phase estimation for optimizing policy or value functions. We will then explain how quantum agents interact with the environment, make decisions based on the quantum state and rewards, and update their policies or value functions accordingly [71]. Finally, we illustrate the iterative process of QRL, where the quantum agent uses trial-and-error to interact with the environment, updating its strategies to maximize rewards.

Practical exercises: *Exercise 1:* As shown in Figure. 17, we will provide students with a QRL approach that consists of a Grover Autonomous Quantum Agent (GAQA) and a Quantum Tic-Tac-Toe (QTTT) environment [82, 83, 84]. We will assist them in coding this approach and teach them how the quantum strategies affect the GAQA learning and decision-making. *Exercise 2:* In exercise 2, we will ask students to provide a solution for the QTTT environment [82, 83, 84]. This involves teaching students how to run the GAQA, how to make it manipulate the QTTT through a set of quantum gates, and how to construct the quantum circuit that represents the solution of the QTTT. *Exercise 3:* In exercise 3, we will ask students to integrate quantum phase estimation into the GAQA learning process and have them observe the effects on policy optimization or value function estimation.

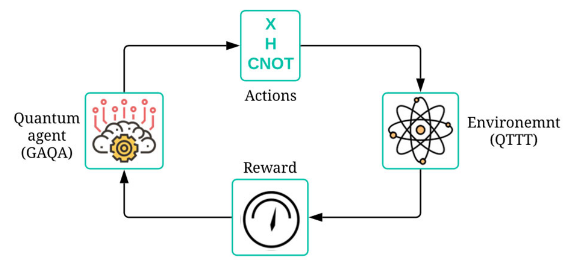


Fig 17. The QRL approach.

Figure 17 illustrates the QRL approach that we will provide to the students. The GAQA interacts with the QTTT environment using quantum gates that represents the actions and utilizes Grover-based probability amplitude updating to select the optimal action to be taken [82, 83]. We used QTTT as an environment because it provides an efficient example of teaching quantum mechanisms. As shown in Figure 18, the QTTT represents a quantum board circuit that consists of qubits that can be manipulated using X, H, and $CNOT$ gates. Therefore, students will gain an in-depth understanding of superposition and entanglement [84].

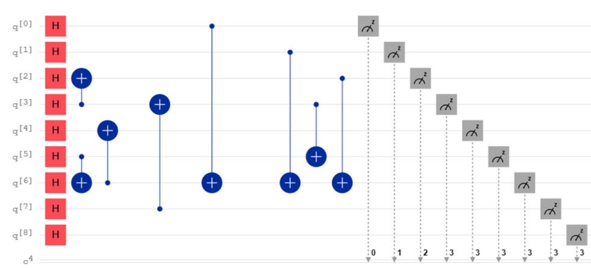


Fig 18. Quantum board circuit of nine qubits.

Figure 18 represents the board circuit of nine qubits that is generated by the GAQA during the learning process. The gates that the actions that the agent takes to interact with the QTTT environment construct the board circuit.

j. Week 14: Hybrid Quantum-Classical Machine Learning:
In this week, we introduce students to HCQML. Our approach will consist of a combination of theoretical knowledge and practical exercises, enabling students to understand the fundamental components of HCQML.

Theoretical Concepts: First, we will explain the motivation behind HCQML, which aims to integrate the strengths of both quantum computing and classical machine learning to handle complex computational problems and enhance machine learning tasks [67]. Then, we will emphasize that HCQML leverages the power of quantum computations in specific parts of the learning process while utilizing classical machine learning algorithms for other aspects. Finally, we will describe the workflow of HCQML, which typically involves a combination of classical and quantum components. We will explain how classical data preprocessing and feature extraction techniques are employed to prepare the data for hybrid processing. Further, we illustrate how quantum algorithms and computations are used in specific tasks, such as quantum optimization for parameter tuning, quantum sampling for generating training data, or quantum feature selection. We then emphasize that classical machine learning models are still utilized for decision-making such as classification, regression, or result interpretation.

Practical exercises: *Exercise 1:* As shown in Figure 18, we intend to provide students with a Hybrid Classical-Quantum Neural Network (HCQNN) and guide them in coding it using Pennylane and other machine learning packages. The students will train the HCQNN using a solar radiation dataset and evaluate its performance [68]. *Exercise 2:* In exercise 2, students will be asked to expand the HCQNN created in Exercise 1. They will add more classical layers and introduce a new quantum layer to the HCQNN. Then, they will evaluate and analyze the performance of the HCQNN.

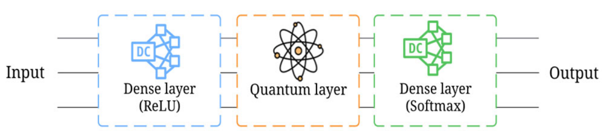


Fig 18. The HCQNN approach.

As shown in Fig. 18, the HCQNN consists of three layers: a dense layer that uses the Rectified Linear Unit (ReLU) function, a quantum layer, and another dense layer that utilizes the Softmax function.

```
dev = qml.device("default.qubit", wires=n_qubits)
@qml.qnode(dev)
def qnode(inputs, weights):
    qml.templates.AngleEmbedding(inputs, wires=range(n_qubits))
    qml.templates.StronglyEntanglingLayers(weights, wires=range(n_qubits))
    return [qml.expval(qml.PauliZ(i)) for i in range(n_qubits)]

weight_shapes = {"weights": (layers,n_qubits,3)}

model = tf.keras.models.Sequential()
model.add(tf.keras.layers.Dense(n_qubits,activation='relu',input_dim=2))
model.add(qml.qnn.KerasLayer(qnode, weight_shapes, output_dim=n_qubits))
model.add(tf.keras.layers.Dense(data_dimension, activation='softmax'))
```

Fig 19. HCQNN layers.

Figure 19 shows the sample code to implement HCQNN layers. The quantum layer represents a quantum circuit that consists of encoding, entanglement, and measurements. The output of this layer is passed to the last dense layer to perform the classification

V. QML Course Assessment

In this section, we explain how the proposed course can be assessed from Accreditation Board for Engineering and Technology (ABET) program learning outcomes perspective. by mapping CS program outcomes with this QML course student course learning outcomes. Table 2 depicts the six course learning outcomes that measure the student's success after the successful completion of this course.

Table 2 – Student Course Leaning Outcomes

Course Learning Outcome	*Course Learning Outcome description*
CLO1	Describe Basics and Use cases of Quantum Computing and Quantum Machine Learning
CLO2	Explain Fundamentals of Quantum Machine Learning, Feature Maps, Kernel Tricks, and fundamental algorithms.
CLO3	Implement Quantum Machine Learning Techniques such as QSVM, QNN, QGAN.
CLO4	Encode classical information into quantum states
CLO5	Implement appropriate Quantum Algorithms such as VQE/VQA, and QUBO
CLO6	Implement QML project and present the project outcomes orally and as a written report

CLO1 and CLO2 will be assessed through the mid-term and final exam. CLO3, CLO4 and CLO5 will be assessed through practical exercise-based assignments. CLO6 will be assessed through in class presentations and term project report that is submitted at the end of the semester. In addition to the above assessment instruments, online quizzes can be administered through learning management platform such as blackboard on a weekly basis.

Following ABET Computer Science program learning outcomes can be mapped to the QML course learning outcomes: PO1. Analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions. PO2. Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline. PO3. Communicate effectively in a variety of professional contexts. PO4. Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles. PO5. Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline. Table 3 shows the mapping of this QML course student learning outcomes with the ABET CS program outcomes. Following Table 3 represents the mapping of ABET program learning outcomes and this QML course student learning outcomes described earlier. Students will also be asked to complete a self-assessment on how they perceive themselves with the course student learning outcomes at the beginning and end of the semester. The outcome from these self-assessments will be used to measure the success of the course.

Table 3 – Mapping of ABET PO and CLO

	PO1	PO2	PO3	PO4	PO5
CLO1	X				
CLO2	X				
CLO3		X			
CLO4		X			
CLO5		X			
CLO6	X		X	X	X

VI. Future Work

Our plan is to implement this course on an annual basis. Also, in the future, we plan to incorporate this course as part of the certificate and degree program related to Quantum computing. We also plan to deploy this course as an online MOOC course.

To assess and mitigate the risks, the course would need to have an annual online course and program evaluation survey to assess the program. Student course evaluations and classroom assessments (peer evaluations) will also be an integral part of the assessment process and course enhancement. Also, we plan to be in touch with alumni through social networking site such as LinkedIn to gather feedback based on their experience and how well the course prepared them for careers in quantum computing and/or quantum machine learning.

VII. Conclusion

To address the existing need, we have proposed the course design and assessment plan of Quantum machine learning course as part of the computer science college degree program. We plan to deploy this in the upcoming Spring semester for the senior undergraduate/first year graduate students as an elective course. This paper proposes a new advancement in the field of Computing education by proposing the QML course as part of the Computer Science college degree program. We believe the course will help the students to become QML professionals with sound theoretical background and practical skillset, making them very attractive to future employers and can further promote cooperation between different scientific disciplines.

Acknowledgement

This work was supported by National Science Foundation Grant No. OMA 2231377.

References

[1] Malloy. Computer Network Defense Through Radial Wave Functions. *ArXiv, abs/1610.01734*, 2016.

[2] Y. Cao, Guerreschi, G.Giacomo, and A. Aspuru-Guzik. Quantum Neuron: an elementary building block for machine learning on quantum computers, 2017.

[3] T.M. Khan, and A. Robles-Kelly. Machine Learning: Quantum vs Classical. IEEE Access, 8, 219275–219294. 2020

[4] M. Schuld, I. Sinayskiy, and F. Petruccione An introduction to quantum machine learning. Contemp. Phys, 56(2), 172–185. 2015.

[5] M. Cerezo et al., "Challenges and opportunities in quantum machine learning." Nature Computational Science 2, no. 9 (2022): 567-576.

[6] S.L. Wu, and S. Yoo. "Challenges and opportunities in quantum machine learning for high-energy physics." Nature Reviews Physics 4, no. 3 (2022): 143-144.

[7] L. Gyongyosi, and S. Imre. "A survey on quantum computing technology." Computer Science Review 31 (2019): 51-71.

[8] M. Piattini. "Training Needs in Quantum Computing." In QANSWER, pp. 23-30. 2020.

[9] C. L. Friedrichs, N. Honegan, D. Zaferatos, D. Evans, A. Leider, G. Aboujaoude, and C. C. Tappert. "An Analysis of Quantum Computing Courses World Wide." Proceedings of Seidenberg Student Faculty Research Day (2021).

[10] H. Huang et al., "Power of data in quantum machine learning." Nature communications 12, no. 1 (2021): 2631.

[11] O. Salehi, Z. Seskir, and İ. Tepe. "A Computer Science-Oriented Approach to Introduce Quantum Computing to a New Audience." IEEE Transactions on Education (2021).

[12] C. Metz, NY Times Article, "The Next Tech Talent Shortage: Quantum Computing Researchers", Last Accessed using https://www.nytimes.com/2018/10/21/technology/quantum-computing-jobs-immigration-visas.html June 2021

[13] PRNewswire Article. "The Worldwide Quantum Computing Industry is Expected to Reach $1.7 Billion by 2026", Last Accessed using https://www.prnewswire.com/news-releases/the-worldwide-quantum-computing-industry-is-expected-to-reach-1-7-billion-by-2026--301229132.html on June 2021.

[14] Sussman, Ben, Paul Corkum, Alexandre Blais, David Cory, and Andrea Damascelli. "Quantum Canada." Quantum Science and Technology 4, no. 2 (2019): 020503.

[15] Riedel, Max, Matyas Kovacs, Peter Zoller, Jürgen Mlynek, and Tommaso Calarco. "Europe's quantum flagship initiative." Quantum Science and Technology 4, no. 2 (2019): 020501.

[16] Roberson, T. M., and Andrew G. White. "Charting the Australian quantum landscape." Quantum Science and Technology 4, no. 2 (2019): 020505.

[17] Raymer, Michael G., and Christopher Monroe. "The US national quantum initiative." Quantum Science and Technology 4, no. 2 (2019): 020504.

[18] Yoshihisa Yamamoto et al 2019 Quantum Sci. Technol. 4 020502.

[19] A K Fedorov et al 2019 Quantum Sci. Technol. 4 040501.

[20] Qiang Zhang et al 2019 Quantum Sci. Technol. 4 040503.

[21] S. Mensa et al., "Quantum machine learning framework for virtual screening in drug discovery: a prospective quantum advantage." Machine Learning: Science and Technology 4, no. 1 (2023): 015023.

[22] H. Huang, R. Kueng, and J. Preskill. "Information-theoretic bounds on quantum advantage in machine learning." Physical Review Letters 126, no. 19 (2021): 190505.

[23] Thew, Rob, Thomas Jennewein, and Masahide Sasaki. "Focus on quantum science and technology initiatives around the world." Quantum Sci. Technol 5, no. 1 (2019): 010201.

[24] Srivastava, Rupesh, Iris Choi, Tim Cook, and NQIT User Engagement Team. "The commercial prospects for quantum computing." Networked Quantum Information Technologies (2016).

[25] Quereca Inc., Overview on quantum initiatives worldwide, Last Updated Jan 2021accessed online using https://www.qureca.com/overview-on-quantum-initiatives-worldwide/ Last Accessed June 2021.

[26] Quantum Consortium Job Board -accessed online https://quantumconsortium.org/quantum-jobs/ . Last Accessed June 2021. [

[27] M. Mykhailova. "Developing Programming Assignments for Teaching Quantum Computing and Quantum Programming." In *2022 IEEE International Conference on Quantum Computing and Engineering (QCE)*, pp. 688-692. IEEE, 2022.

[28] G. Temporão, TBS Guerreiro, P.S.C Ripper, and AMB Pavani. "Teaching Quantum Computing without prerequisites: a case study." In *2022 IEEE International Conference on Quantum Computing and Engineering (QCE)*, pp. 673-676. IEEE, 2022.

[29] J.R.Wootton, F.Harkins, N. T. Bronn, A. Carrera Vazquez, A. Phan, and A.T. Asfaw. "Teaching quantum computing with an interactive textbook." In *2021 IEEE International Conference on Quantum Computing and Engineering (QCE)*, pp. 385-391. IEEE, 2021.

[30] G. De Luca, and Y. Chen. "Teaching Quantum Machine Learning in Computer Science." In *2023 IEEE 15th International Symposium on Autonomous Decentralized System (ISADS)*, pp. 1-7. IEEE, 2023.

[31] S. Ganguly . Quantum Machine Learning: An Applied Approach. Apress, 2021.

[32] P. Rebentrost, M. Mohseni and S. Lloyd, "Quantum sup-port vector machine for big data classification", Physical review letters, vol. 113, no. 13, pp. 130503, 2014.

[33] F. Jazaeri, A. Beckers, A. Tajalli, and J. M. Sallese, "A Review on quantum computing: From qubits to front-end electronics and cryogenic mosfet physics," in Proceedings of the 26th International Conference "Mixed Design of Integrated Circuits and Systems", MIXDES 2019, 2019, pp. 15–25.

[34] M. Mafu and M. Senekane, "Design and Implementation of Efficient Quantum Support Vector Machine," 2021 International Conference on Electrical, Computer and Energy Technologies (ICECET), Cape Town, South Africa, 2021, pp. 1-4.

[35] I. Sharafaldin, A. H. Lashkari, S. Hakak, and A. A. Ghorbani, "Developing Realistic Distributed Denial of Service (DDoS) Attack Dataset and Taxonomy", IEEE 53rd International Carnahan Conference on Security Technology, Chennai, India, 2019.

[36] A. Zeguendry, Z. Jarir, and M. Quafafou, "Quantum Machine Learning: A Review and Case Studies," Entropy, vol. 25, no. 2, Art. no. 2, Feb. 2023.

[37] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, "Quantum machine learning," Nature, vol. 549, no. 7671, Art. no. 7671, Sep. 2017.
[38] Z. Abohashima, M. Elhoseny, E. H. Houssein, and W. M. Mohamed, "Classification with Quantum Machine Learning: A Survey".
[39] K. Ghosh, "Encoding classical data into a quantum computer," Jul. 2021.
[40] T. Goto, Q. Tran, and K. Nakajima, "Universal Approximation Property of Quantum Feature Map," ArXiv, Sep. 2020,
[41] Shafi, "Quantum Data and its Embeddings # 1," MLearning.ai, Jan. 02, 2022.
[42] H. Wiseman and G. Milburn, "Quantum Measurement and Control," Nov. 2009.
[43] R. Fauzi, M. Zarlis, H. Mawengkang, and P. Sihombing, "Analysis of Several Quantum Encoding Methods Implemented on A Quantum Circuit Architecture to Improve Classification Accuracy," 2022 6th Int. Conf. Electr. Telecommun. Comput. Eng. ELTICOM, pp. 152–154, Nov. 2022.
[44] L. Deng, "The MNIST Database of Handwritten Digit Images for Machine Learning Research [Best of the Web]," IEEE Signal Process. Mag., vol. 29, no. 6, pp. 141–142, Nov. 2012.
[45] "PennyLane." https://pennylane.ai/
[46] "Qiskit." https://qiskit.org/
[47] "UCI Machine Learning Repository: Iris Data Set." https://archive.ics.uci.edu/ml/datasets/iris
[48] C. B. Prieto, "Variational quantum architectures".
[49] M. Benedetti, E. Lloyd, S. Sack, and M. Fiorentini, "Parameterized quantum circuits as machine learning models," Quantum Sci. Technol., vol. 4, no. 4, p. 043001, Nov. 2019.
[50] "Variational Quantum Classifier (VQC) — Qiskit 0.24.1 documentation."
[51] J. Xiao, Y. Yan, J. Zhang, and Y. Tang, "A quantum-inspired genetic algorithm for k-means clustering," Expert Syst. Appl., vol. 37, no. 7, pp. 4966–4973, Jul. 2010.
[52] E. Aïmeur, G. Brassard, and S. Gambs, "Quantum clustering algorithms," in Proceedings of the 24th international conference on Machine learning, Corvalis Oregon USA: ACM, Jun. 2007, pp. 1–8.
[53] E. Casper and C. Hung, "Quantum Modeled Clustering Algorithms for Image Segmentation," 2013.
[54] S. U. Khan, A. J. Awan, and G. Vall-Llosera, "K-Means Clustering on Noisy Intermediate Scale Quantum Computers." arXiv, Sep. 26, 2019. doi: 10.48550/arXiv.1909.12183.
[55] "Customer Segmentation Dataset." https://www.kaggle.com/datasets/yashgupta011/customer-segmentation-dataset
[56] "[PDF] Quantum generative adversarial networks | Semantic Scholar.".
[57] "[PDF] Quantum Boltzmann Machine | Semantic Scholar.".
[58] A. Khoshaman, W. Vinci, B. Denis, E. Andriyash, H. Sadeghi, and M. H. Amin, "Quantum variational autoencoder," Quantum Sci. Technol., vol. 4, no. 1, p. 014001, Sep. 2018.
[59] E. Fontana, N. Fitzpatrick, D. M. Ramo, R. Duncan, and I. Rungger, "Evaluating the noise resilience of variational quantum algorithms," Phys. Rev. A, vol. 104, no. 2, p. 022403, Aug. 2021.
[60] D. Volya and P. Mishra, "Special Session: Impact of Noise on Quantum Algorithms in Noisy Intermediate-Scale Quantum Systems," 2020 IEEE 38th Int. Conf. Comput. Des. ICCD, pp. 1–4, Oct. 2020.
[61] M. Bączyk, "Noise influence on Quantum Machine Learning models' performance," 2021.
[62] "[PDF] Efficient noise mitigation technique for quantum computing | Semantic Scholar."
[63] "[PDF] Mitigating Realistic Noise in Practical Noisy Intermediate-Scale Quantum Devices | Semantic Scholar."
[64] C. J. Wood, "Special Session: Noise Characterization and Error Mitigation in Near-Term Quantum Computers," 2020 IEEE 38th Int. Conf. Comput. Des. ICCD, pp. 13–16, Oct. 2020.
[65] S. Yan, H. Qi, and W. Cui. "Non-linear quantum neuron: A fundamental building block for quantum neural networks." Physical Review A 102, no. 5 (2020): 052421.
[66] M. Maronese, C. Destri, and E. Prati, "Quantum activation functions for quantum neural networks," *Quantum Inf Process*, vol. 21, no. 5, p. 128, May 2022, doi: 10.1007/s11128-022-03466-0.
[67] M. A. Metawei, H. Said, M. Taher, H. Eldeib and S. M. Nassar, "Survey on Hybrid Classical-Quantum Machine Learning Models," *2020 International Conference on Communications, Computing, Cybersecurity, and Informatics (CCCI)*, Sharjah, United Arab Emirates, 2020, pp. 1-6, doi: 10.1109/CCCI49893.2020.9256649.
[68] E. Baldasso, "Prediction of solar radiation data," 2021. [Online]. Available: https://www.kaggle.com/code/enricobaldasso/prediction-of-solar-radiation-data/input. [Accessed: Apr. 20, 2023].
[69] D. Dong et al., "Quantum Reinforcement Learning," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 38, no. 5, pp. 1207-1220, 2008, doi: 10.1109/TSMCB.2008.925743.
[70] D. Heimann et al., "Quantum Deep Reinforcement Learning for Robot Navigation Tasks*", arXiv:2202.*12180, 2022.
[71] A. Skolik et al., "Quantum agents in the Gym: a variational quantum algorithm for deep Q-learning", *Phys. Rev*, vol 6, doi:10.22331/q-2022-05-24-720, 2022.
[72] R. Shaydulin and Y. Alexeev, "Evaluating Quantum Approximate Optimization Algorithm: A Case Study," in 2019 Tenth International Green and Sustainable Computing Conference (IGSC), Alexandria, VA, USA: IEEE, Oct. 2019, pp. 1–6. doi: 10.1109/IGSC48788.2019.8957201.
[73] L. Zhou, S.-T. Wang, S. Choi, H. Pichler, and M. D. Lukin, "Quantum Approximate Optimization Algorithm: Performance, Mechanism, and Implementation on Near-Term Devices," Phys. Rev. X, vol. 10, no. 2, p. 021067, Jun. 2020, doi: 10.1103/PhysRevX.10.021067.
[74] M. Alam, A. Ash-Saki, and S. Ghosh, "Design-Space Exploration of Quantum Approximate Optimization Algorithm under Noise," 2020 IEEE Cust. Integr. Circui*ts Conf. CICC*, pp. 1–4, Mar. 2020, doi: 10.1109/CICC48029.2020.9075903.
[75] E. Farhi, J. Goldstone, and S. Gutmann, "A Quantum Approximate Optimization Algorithm." arXiv, Nov. 14, 2014.
[76] M. Willsch, D. Willsch, F. Jin, H. De Raedt, and K. Michielsen, "Benchmarking the quantum approximate optimization algorithm," Quantum Inf. Process., vol. 19, no. 7, p. 197, Jun. 2020.
[77] Wang Di and Robert Kleinberg, "Analyzing quadratic unconstrained binary optimization problems via multicommodity flows", *Discrete Applied Mathematics*, vol. 157, no. 18, pp. 3746-3753, 2009.
[78] T. Gabor, M. L. Rosenfeld, C. Linnhoff-Popien and S. Feld, "How to Approximate any Objective Function via Quadratic Unconstrained Binary Optimization," *2022 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER)*, Honolulu, HI, USA, 2022, pp. 1249-1257, doi: 10.1109/SANER53432.2022.00149.
[79] K. Tanahashi, S. Takayanagi, T. Motohashi, and S. Tanaka, "Application of Ising Machines and a Software Development for Ising Machine," *Journal of the Physical Society of Japan*, vol. 88, no. 6, article no. 064802, 2019.
[80] L. Zhou, S. Wang, S. Choi, H. Pichler, and M. D. Lukin, "Quantum Approximate Optimization Algorithm: Performance, Mechanism, and Implementation on Near-Term Devices," *Physical Review X*, vol. 10, no. 2, pp. 144-152, 2020. doi: 10.1103/PhysRevX.10.021067.
[81] J. Bowles et al., "Quadratic Unconstrained Binary Optimization via Quantum-Inspired Annealing," *Phys. Rev. Appl*., vol. 18, no. 3, p. 034016, Sep. 2022.
[82] L. K. Grover, "Quantum mechanics helps in searching for a needle in a haystack," *Phys. Rev. Lett*, vol.79, pp.325-327, 1997.
[83] S. Wiedemann et al., "Quantum Policy Iteration via Amplitude Estimation and Grover Search-Towards Quantum Advantage for Reinforcement Learning," *Phys. Rev. Lett*, arXiv:2206.04741, 2022, doi: 10.48550/ARXIV.2206.04741.
[84] M. Müller, M. Schmidt, and J. Klein. "Quantum Tic-Tac-Toe - learning the concepts of quantum mechanics in a playful way," *Computer Applications in Engineering Education*, vol. 31, no. 2, pp. 1-8, 2023, doi: 10.1016/j.caeo.2023.100125.