Journal of Computer Science

Original Research Paper

Investigation and Statistical Analysis of Cloud Droplet Dynamics Using Quantum Computing

[1]Mukta Nivelkar, [1]Sunil Bhirud, [2]Rahul Ranjan and [3]Bipin Kumar

[1]*Department of Computer Engineering and IT, Veermata Jijabai Technological Institute, Mumbai, India*
[2]*Department of Environmental Science, Stockholm University, Veermata Jijabai Technological Institute, Mumbai, India*
[3]*Indian Institute of Tropical Meteorology, Ministry of Earth Science, Government of India, Pashan, Pune, India*

Article history
Received: 02-10-2023
Revised: 23-10-2023
Accepted: 18-12-2023

Corresponding Author:
Mukta Nivelkar
Department of Computer Engineering and IT, Veermata Jijabai Technological Institute, Mumbai, India
Email: mmnivelkar_p19@ce.vjti.ac.in

Abstract: Cloud droplet dynamics is an important part of cloud physics. This element of cloud physics analyses the features of each droplet, including its size distribution, probability density and mean saturation. The cloud's structure is significantly important for the Earth's atmosphere and this structure is affected by changes in the droplet's micro-physical properties. In order to investigate and understand the dynamics of cloud droplets in both the high and low vortex areas, data obtained from Direct Numeric Simulations (DNS) are utilized. Data generated from simulations of cumulus clouds, which are defined as low-level clouds located between 800 and 1200 m above the surface of the earth. DNS data reveals complex droplet dynamics on a scale that is three-dimensional. When employing conventional machine learning methods, the processing of data relating to dynamic droplets requires a substantial amount of CPU resources. In this study, we discussed the advantages of using quantum mechanisms in cloud physics in order to investigate the complicated nature of cloud droplets. The use of quantum computing in the study of droplet dynamics using the quantum k-mean approach was further investigated in the discussion. Quantum machine learning is used to study the micro-physical characteristics of cloud droplets in order to investigate the effect that droplet dynamics have on the overall structure of clouds. The current topic of discussion delves more into the specifics of how data relating to DNS can be processed by an analog quantum computer in order to deal with enormous amounts of data in this specific area of research.

Keywords: Quantum Computing and Machine Learning, Direct Numeric Simulation, Superposition and Entanglement, Cloud Droplets, Vorticity

Introduction

Global climate models are facing several hurdles in data processing and the accuracy of model outcomes. Cloud microphysics plays a very crucial role in assessing global climate models. In this study, we have integrated cloud physics with quantum computing. Cloud physics is the study of the physical processes of clouds such as cloud formation, density, growth and formation of precipitation. Cloud is one of the major components of the climate model and plays a very important role in Earth's atmosphere. World Meteorological Organization categorizes clouds into three major types cumulus, stratus and cirrus. The other clouds are represented as either a combination or modification of these forms. Clouds have various levels based on the height earth's surface, these levels are called high, middle and lower levels of clouds. Climate models come with a large scope and researchers' attention is required in many sectors of the domain. Our interest lies in the study of cloud droplet dynamics using quantum mechanisms. A droplet refers to a tiny particle of cloud. The microphysical properties of these droplets have the ability to alter the structure and arrangement of cloud formations. Our study conducts how quantum mechanism benefits to analyze the micro-physical properties of cloud (Kumar *et al*., 2021). To research this domain Direct Numeric Simulation (DNS) (Kumar *et al*. 2014) data and its formats are discussed in the upcoming section. DNS represents cloud droplet data of 3-dimension in size which consists of velocity and temperature features. Further discussion explores how this domain benefited from the integration of quantum mechanisms. Several sectors benefit from using machine learning through accurate analysis of data required to generate reports, predictions, decision-making and optimization. Machine learning and quantum computing have coalesced to create quantum machine learning. Quantum machine learning was chosen over classical machine learning to investigate the benefits of quantum mechanisms such as superposition and entanglement. In contrast to classical bit data, a quantum bit as a qubit is used to represent data in quantum states. The traditional approach of machine learning methods on climate data processing results in poor performance in terms of accuracy and time. Huge and complex data processing and analysis is a major challenge in this domain. It takes more computational power and resources. Therefore, we have to find how future quantum computers solve these obstacles. We have identified issues such as the application of climate models on small and error-prone devices as one of the major points during simulation and modeling. Secondly, the amount of data is very large so how to load the data, embed data into quantum states and data prepossessing.

Quantum-supervised machine learning and classification problems are discussed using various quantum tools and subroutines. how these methods accelerate the performance of the algorithm by using labeled learning techniques such as Support Vector Machine (SVM) and k-nearest neighbor using classical and quantum tools are discussed by Ablayev *et al.* (2019a-b). This study highlights more advanced tools and subroutines to implement using the Qiskit tool on the IBM cloud (IBM, 2020). Machine learning has been one of the major and successful branches of artificial intelligence over the years. ML has very well-set methods for classical mechanisms. The theoretical and mathematical perspective of the machine learning algorithm is well-tested and executed on the traditional approach of computing. Identifying the complex problem that is classically hard to implement on the classical mechanism is selected to integrate with the quantum mechanism discussed in the paper (Li *et al*., 2020; Maheshwari *et al*., 2022). Application in biomedical domain and Eeg signal feature extraction discussed on use of quantum mechanics and its advantages over selected domains. Classical vs. quantum machine learning pros and cons of proposed methods are specified and various algorithms are discussed (Khan and Robles-Kelly, 2020). Egger *et al*. highlight how quantum computing works in finance problems (Egger *et al*., 2020); Pushpak and Jain, mention application in insurance claim fraud (Pushpak and Jain, 2022). The current quantum device is small and noisy, the limitation of current hardware in noisy intermediate-scale quantum computers using circuit compilation is reviewed and it helps to understand the scenario of hardware while proposing models for various other domains (Kusyk *et al*., 2021). Quantum support vector machine is binary classification and how it is integrated with quantum mechanism is discussed (Mafu and Senekane, 2021). Author Schuld has proposed a classical-quantum approach to quantum machine learning. Supervised learning using quantum computing is discussed and explored with various methods to accelerate the process. The classical-quantum approach requires classical data to be converted into quantum states. Various methods for encoding such as basis, amplitude and hybrid for classical data into quantum states are articulated by (Schuld *et al*., 2015). The design of the Variational Quantum Classifier (VQC) and the effect of encoding techniques on VQC is elaborated (Schuld *et al*., 2021). Use of variational quantum circuits as linear model, quantum kernel method and how model classifies data explicitly in Hilbert space. Feature map design and quantum kernel implementation over a sample 2-dimensional data set are discussed in the article (Schuld and Killoran, 2019).

Quantum Computer Mechanism

Our goal is to use the Noisy Intermediate Scale Quantum computer (NISQ) to develop a solution and a technique. NISQ are the 100+ qubit quantum computers of the near future. An unsupervised machine learning algorithm called quantum k-mean is proposed on IBM Qiskit. Open source software development kit Qiskit. Commercial quantum computers are not yet accessible. Thus, IBM quantum devices are used in experimental implementation on cloud-based quantum computers. Results are improved to function properly on near-term quantum computers. The implementation is done using the simulator for IBMQ QASM. A variety of quantum gates and instructions are incorporated into this 32-qubit general-purpose, context-aware and noise-modeling simulator to represent quantum circuits. Detailed instructions for connecting to cloud quantum computer simulators are provided below (IBM, 2020).

```
import qiskit. providers. ibmq. jupyter IBMQ.
load_account ()
provider = IBMQ. get_provider (group =’ open’,
 project =’ main’)
system = provider.ge-backend(’ibm-qasm-
simulator’)
system
```

The disparity between the mathematical model and the practical model has narrowed in recent years. Quantum computers employ quantum phenomena like superposition and entanglement over the qubit. A “qubit" also referred to as a quantum bit, is a quantum representation of data. Fault-tolerant quantum computers are not feasible at the available intermediate scale with noise. Due to its short coherence time, the qubit hypothesis would only hold for a short time. As a result, the output circuit develops a flaw. Shallow-depth quantum circuits can provide us with better results on NISQ devices.

Quantum Circuit Composition

Qubit data is represented using vectors of bits. Dirac vector notation to represent state zero of a qubit is $|0\rangle$ and state one as $|1\rangle$. Quantum computations are measured by performing several operations on qubits using various quantum gates (Wittek, 2014). Quantum circuits are used to model the quantum computations on sample data. Quantum circuit design using IBM quantum circuit composer (IBM, 2020). Figure 1 shows a quantum circuit for Bell state design using 2-qubits. Hadamard gate H is used to superposition on qubit q [0] wire and the CNOT gate is used to entangle the qubit q [0] and qubit q [1]. Bell state of 2-qubits represented as:

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

Quantum Interference

Quantum particle has two states described as wave-particle duality. The quantum interference pattern is generated by the superposition of waves. The amplitude of a wave is measured as constructive and destructive interference. Waves that are in phase produce constructive interference and waves that are with out of phase of 180° produce destructive interference in which amplitude cancels each other. Hadamard gate is used as an interference transformation. Superposition of 2-qubit quantum state is:

$$|\phi\rangle = a_0|00\rangle + a_1|01\rangle + a_2|10\rangle + a_3|11\rangle \tag{2}$$

Amplitude vector matrix represented as:

$$|\phi\rangle = \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} \tag{3}$$

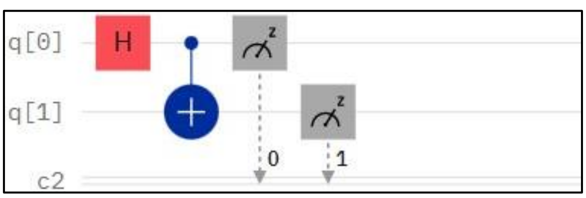

Fig. 1: Quantum circuit model

Interference of four amplitude of quantum states is achieved using the Hadamard gate. The Hadamard gate is a unitary matrix represented as follows:

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{4}$$

The tensor product of the Hadamard gate for a two-qubit system is represented as:

$$H \otimes I = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix} \tag{5}$$

Applying $H \otimes I$ to the state φ results in the following equation:

$$|\Phi\rangle = (H \otimes I)|\phi\rangle \tag{6}$$

$$|\Phi\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} a_0 + a_2 \\ a_1 + a_3 \\ a_0 - a_2 \\ a_1 - a_3 \end{pmatrix} \begin{pmatrix} |0\,0\rangle \\ |0\,1\rangle \\ |1\,0\rangle \\ |11\rangle \end{pmatrix} \tag{7}$$

Qubits interfere with each other and result in constructive and destructive interference of amplitude. The aforementioned mathematical representations are all utilized in the construction of the quantum computing circuit. The quantum mechanism is utilized in order to do an analysis of the micro-physical characteristics of cloud droplets. The transformation of classical data into quantum states offers numerous benefits, including the ability to investigate previously concealed features in the data. Finding intricate patterns in large amounts of data is a task that is conventionally difficult. The utilization of quantum mechanisms makes it possible to carry out a variety of rotations on data points. The Quantum mechanism is responsible for handling the process of designing high-dimensional feature sets for traditionally complex data by employing rotations and block rotations.

Introduction to the Proposed System

System Flowchart

The method of quantum computing is applied in the system that has been presented in order to process 128 mm^3 worth of data that pertains to DNS domains. Quantum computers are able to enhance both the speed at which computations are performed as well as the time complexity of the system. The quantum circuit's implementation and measurement are handled by separate modules within the system. These modules are responsible for the system's data, pre-processing and

DOI: 10.3844/jcssp.2024.344.356

conversion functions, respectively. It is possible to acquire the data from DNS in the format that is commonly used. The information obtained from a conventional DNS system is converted into quantum states. Figure 2 provides a visual representation of the several phases that comprise the processing of DNS data. A quantum feature map is utilized for the purpose of encoding data in order to transfer it from a classical state to a quantum state. The quantum kernel matrix is constructed in such a way that it is determined by the training data pairings. The parameter shift module is the fundamental component that underpins the optimization of the quantum circuit. The overall goal of utilizing quantum machine learning is to get the quantum benefit in terms of data processing and this is done by employing the methodology in its entirety. As a consequence of this, the quantum circuit is executed a number of times in order to overcome the effects of quantum noise.

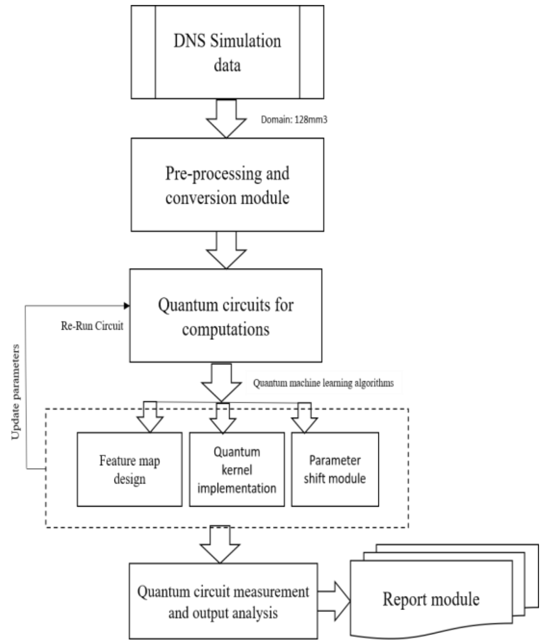

Fig. 2: DNS data processing using quantum computing

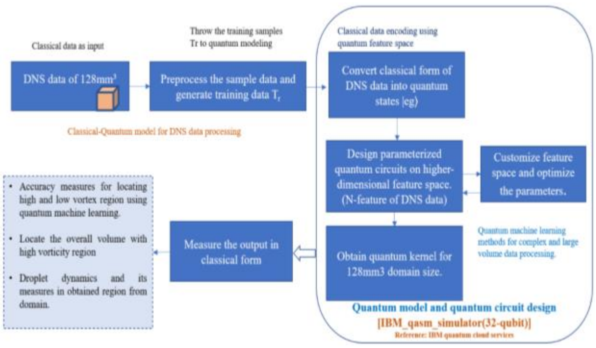

Fig. 3: Classical-quantum model for DNS data processing

System Architecture

The system design that is proposed for a small cloud domain with a budget that falls somewhere between 128 and 2048 mm^3 is depicted in Fig. 3. This architecture may be found in the figure. The method is accurate in determining the precise positions of both the low and high vortex zones. The reporting module, along with the data module, the pre-processing module and the quantum circuit module, make up the architecture. Also included is the quantum circuit module. In the course of this inquiry, we will be utilizing a DNS domain that has a price tag of 128 ms of dollars. Following the completion of pre-processing on the data obtained from the Domain Name System (DNS), the information is next translated into a format that is compatible with quantum machine learning. Data must be represented in a form that is consistent with quantum states for quantum computing to be possible. Before continuing with the processing of the data, it is necessary to transform the classical data into quantum states. This data has been handed to you. In order to convert classical information into quantum states, a variety of encoding methods, such as basis, amplitude, angle and hybrid, are utilized (Schuld *et al*., 2021). The purpose of building quantum circuits is to allow for the processing of data in a manner that is consistent with quantum theory. When it comes to dealing with the noise that is produced by the circuits, quantum computers encounter a substantial obstacle. This might be seen as a significant barrier. In order to lessen the amount of sampling noise, we have incorporated a particular strategy into the quantum circuits that we have developed. The system architecture is made up of its two fundamental components, which are the classical model and the quantum circuit model respectively. The Classical-Quantum paradigm (also known as CQ) is going to be used for the aim of this investigation. In the first part of the process, the data will be analyzed and pre-processed by using procedures that are more often employed. This is the starting point. Creating quantum circuits that can be used for encoding, calculations and the actual measurement of the results is the second step in this process. Quantum machine learning possesses a few special qualities that standard learning methodologies, when applied to traditional computers, are unable to mimic. These qualities pertain to the fact that quantum computers are able to solve problems that ordinary computers cannot. A computer model that is capable of simulating a system that conducts computing operations using a quantum device while simultaneously applying quantum machine learning to data is the goal of this research. This model will be developed as part of this study. The suggested system is experimentally developed with the QASM simulator (IBM, 2020) and droplets are divided into high vortex locations and low vortex regions utilizing quantum machine learning from a given domain.

The vorticity of each individual droplet is currently being determined. In addition, we looked at the concept of vorticity and illustrated how it relates to a variety of other aspects.

When it comes to the process of feature encoding, the operation of a variational quantum circuit is contingent on parameters such as the angle rotation theta. In many circles, variational quantum algorithms are held up as a potentially fruitful way to achieve quantum advantage on devices in the not-too-distant future. The outsourcing of the optimization procedure to a conventional optimizer, which is not required in the majority of cases because these issues do not ask for the execution of deep quantum circuits, is one way to partially reduce the likelihood of making systematic errors. However, the Variational Quantum Algorithm (VQA) also faces a number of challenges. The most significant of these challenges are inquiries into whether or not they are capable of being trained effectively and whether or not they generate answers that are, in fact, superior to those generated by traditional algorithms. In spite of these challenges, Virtual Quality Assessments (VQAs) have been proposed as a solution for a wide variety of problematic scenarios, including the ones that are described below. We have made use of variational quantum circuits in order to construct parameterized quantum models and variational quantum classifiers in order to carry out quantum machine learning utilizing a given sample of data. This has allowed us to accomplish the task of using the data in a way that is more efficient. The Variational Quantum Circuit (VQC) that was constructed by making use of these three characteristics is depicted in Fig. 4. On each individual feature, the rotation operation denoted by R_x is carried out. The superposition and entanglement effects produced by the quantum circuit are a part of this phenomenon. A quantum CNOT gate is utilized in order to couple the qubits together after a Hadamard gate *H* has been applied to each individual qubit wire.

System Objectives

- Find high and low vortex coordinates in the given 3D DNS domain
- Trace the high-vorticity regions
- Analyze the droplet properties in the high vortex regions

Materials and Methods

Our institution serves as the location for the study that is being carried out. The utilization of hardware to carry out experimental work is done on IBM quantum computers, which are also available online as quantum simulators. One of the newer sub-fields of research is known as quantum computing. IBM has made available a number of tutorials on the subject. For the purpose of this study, data was provided by IITM Pune, India.

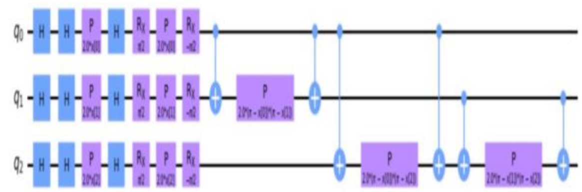

Fig. 4: Quantum feature map using 3-features and several rotations

Methodology A

About Cloud Data

The numerical model that is being used for research is called Direct Numeric Simulation (DNS). DNS models are put to use in research on cloud computing. In order to reap the benefits of a quantum process when applied to fluid data, quantum machine learning is being combined with cloud physics. Owing to the fact that we have chosen data that exhibits a dynamic behavior in terms of the droplet features it possesses. A droplet is a very small particle that makes up a cloud and any changes to the structure of droplets will cause an overall shift in the cloud's composition. We have decided that clouds are one of the elements of the atmosphere that are responsible for the changing climate. The study of how droplet behavior is affected by quantum mechanics contributes to a more precise understanding of how cloud behavior is affected. For the purpose of this study, we are utilizing the DNS data of cloud droplets. The Domain Name System (DNS) data is organized as data points on a three-dimensional Lagrangian scale. The investigation makes use of a cloud domain measuring $128mm^3$, which contains 128×128×128 data points. This study includes the development of a system for the observational analysis of cloud droplets. The system looks for statistical measures of cloud physics parameters that are connected with droplet data in regions of high and low vorticity. The measure of how fluid spins in a certain domain is called its vorticity. The analysis of droplets can be fairly challenging because droplets only remain for a very brief period of time. When it comes to processing fluid data, the traditional method has quite a few obstacles to overcome (Kumar *et al*., 2021).

Calculation of Vorticity

Vorticity is a measurement that can be used to describe rotation and spin in a fluid. There are three components of velocity, denoted *u*, *v* and *w*, at every place. In the standard Eulerian frame used for DNS data format, *w* denotes the vertical velocity component, while *u* and *v* stand for the horizontal velocity components. When calculating the vorticity at specific grid points, these velocity components are taken into account. The magnitude of the vorticity at each position is:

$$W_{net} = \left(W_i^2 + W_j^2 + W_k^2\right)^{\frac{1}{2}} \tag{8}$$

where:

$$W_i = \frac{\partial w}{\partial x} - \frac{\partial v}{\partial z} \tag{9}$$

$$W_j = \frac{\partial u}{\partial x} - \frac{\partial w}{\partial x} \tag{10}$$

$$W_k = \frac{\partial v}{\partial z} - \frac{\partial u}{\partial y} \tag{11}$$

The Eulerian frame of DNS data contains variables like flow velocity, mixing ratio and temperature. Every point in the 128 mm^3 domain consists of a grid size of 1mm and the total points are 128×128×128. We have calculated vorticity at each droplet point and the statistics of the given DNS sample data are shown in Table 1. Data points are categorized into High Vortex (HV) and Low Vortex region (LV). According to the distribution of vorticity, the threshold value is set for HV and LV regions.

The distribution of vorticity in the given DNS domain is shown in Figs. 5-6 shows.

Proposed System

To model quantum supervised machine learning HV and LV regions are traced from 128 mm^3 DNS domain, the resultant HV region is less than 2%. We have used a quantum-supervised learning approach to find the HV regions in the given sample of DNS. We have calculated and projected the outcome on the 2048 mm^3 DNS domain. Table 2 shows statistics for high vortex regions in large-scale DNS domains. The DNS domain is divided into various partitions such as A, B, C, D and so on. By applying quantum supervised learning running time complexity achieved is $O(N^2)$. 'N 'is a number of high-dimensional samples. qRAM requires to store $O(\log_2 N)$ post-processing the quantum data was $O(poly(\log_2 N)$.

Grouping DNS Domains

The grouping of numerous locations and their projection onto a larger DNS scale is displayed in Fig. 7. In order to project output on a bigger scale using the cloud, stitching together multiple 128 mm^3 DNS domains is used. The huge cloud represents an integration of a total of 16 DNS domains (G1-G16). Each domain has around 2 million data points. Every single data point is portrayed through its respective velocity, mixing ratio and temperature attributes. Figure 8 shows a vorticity distribution graph.

Table 1: High vortex points in sample DNS region

Vorticity	>20	>30	>40
128 mm^3	12882	714	61
Percentage	0.61	0.034	0.029

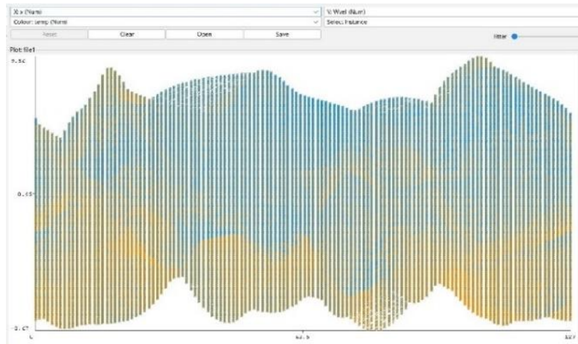

Fig. 5: Plot of distribution of Wvel velocity component against x, y and z coordinates of droplet data

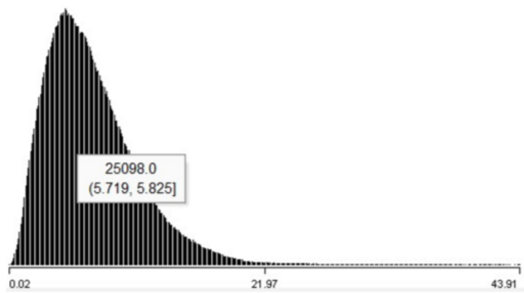

Fig. 6: DNS vorticity ranging between values 0-44

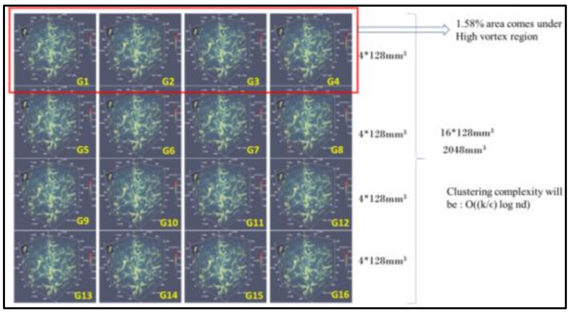

Fig. 7: Large DNS projection

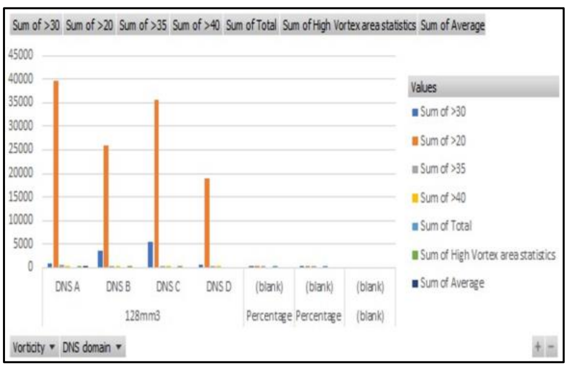

Fig. 8: Vorticity ranges in the DNS domain

DOI: 10.3844/jcssp.2024.344.356

Table 2: High vortex points on large DNS region

Vorticity	DNS domain	> 20	> 30	> 35	> 40	Total	HV% statistics	Average %
128 mm^3	DNS A	39653.000	897.0000	784.0000	47.000		1.89	1.42
Percentage		1.890	0.0420	0.0370	0.0022	1.973		
128 mm^3	DNS B	25896.000	3562.0000	369.0000	96.0000		1.42	
Percentage		1.230	0.1690	0.0175	0.0045	1.426		
128 mm^3	DNS C	35734.000	5434.0000	320.0000	87.0000		1.98	
Percentage		1.703	0.2590	0.0152	0.0041	1.982		
128 mm^3	DNS D	18882.000	714.0000	197.0000	61.0000		0.94	
Percentage		0.900	0.0340	0.0093	0.0029	0.946		

Results

Locating HV and LV Regions

Figure 9 shows how the vortices appear when viewed in three dimensions. It is possible to recognize a vortex by its many different irregular shapes and sizes. It can be difficult to locate vorticity since the shape and size of these features are not clearly defined and the amplitude of these features can change in a fraction of a second. We have recommended dividing the entire domain up into a number of three-dimensional cells (cell size can be 2×2×2) and then calculating the average vorticity of each of those cells. When we compare the vorticity of each 3D cell by utilizing the average vorticity value, we are able to filter out the cells that have a vorticity value that is greater than the threshold value. To carry out this process will take a significant amount of computational resources. In the supplied example DNS domain, we came to the conclusion that 2% of the points fall into the high vorticity zone. Finding areas with strong vorticity can be accomplished using unsupervised learning.

Data Description and Associated Variables

The experimental data used in this study were obtained from DNS (using the model developed by Kumar *et al*. and were made available by IITM Pune (Kumar *et al*., 2014). This sample DNS is a three-dimensional cloud on a side. The variables included in the data, such as velocity, mixing ratio and temperature, are shown in the following language. Each droplet point has a three-dimensional (x, y and z coordinate) size. Accessing an online quantum computer requires the usage of a classical computer. The standard configuration for a traditional computer is an Intel Core i5-10400F processor with 16 gigabytes of memory. On the IBM quantum simulator known as state vector, the programming tool that is used is called quantum qiskit and the programming language that is utilized is Python. The features of droplets are represented by DNS data in terms of their velocities, temperatures and mixing ratios. Each domain has approximately 2 million points in the Lagrangian frame, with one point representing one millimeter of Eulerian variable space. The velocity of a droplet is determined by its three different component dimensions (sizes): x(128), y(128), z(128) variables(dimensions): Float64 Uvel(z, y, x), float64 Vvel(z, y, x), float64 Wvel(z, y, x), float64 qv(z, y, x), float64 temp(z, y, x).

Methodology B

The k-mean algorithm's application to a specific domain is the topic of discussion in this section. In the prior part, we determined the whole region that is subject to high vorticity and came to the conclusion that about 1.58% of the territory is subject to high vorticity. It was challenging to investigate droplets in such places due to the distribution of points over the domain. In order to find a solution to this issue, the HV region has been further subdivided into many cubes of varying sizes. Within each of these tiny cubical zones, the average vorticity is computed and droplets in the region are examined. Quantum k-mean is utilized in order to locate a number of small regions within the provided domain and the running time complexity is computed. When looking for smaller clusters in the DNS region, the quantum k-mean is utilized to aim at high vortex coordinates. The subsequent procedures will demonstrate how it will be implemented using DNS data. The flow of the model that was proposed in approach 2 is illustrated in Fig. 10.

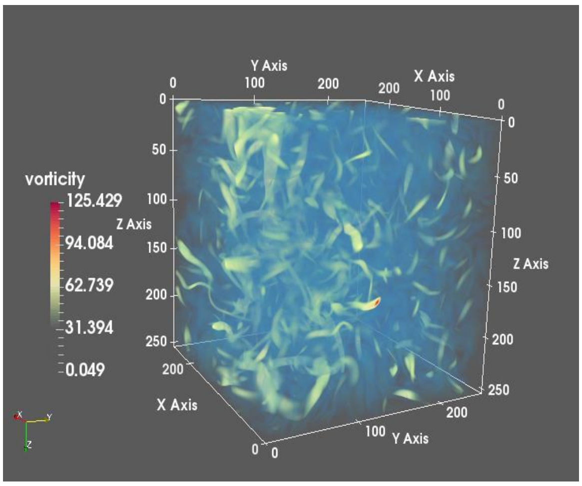

Fig. 9: Vorticity in 3D dimensional space

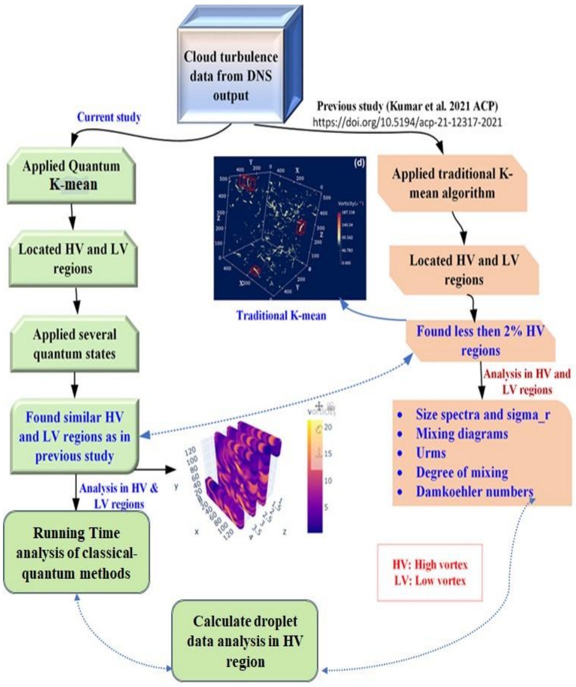

Fig. 10: System working model

Unsupervised Learning Algorithm

We have used classical mechanism unsupervised ML learning using a k-mean algorithm and also on quantum mechanism quantum k-mean is used. K-mean is the most simple unsupervised learning method. K-mean is used to make a k-group in the given data set based on some common features. Using k-mean we will group the given data into high and low vorticity regions. Selection of a number of groups is very important for better results. Classical k-mean is used in the following manner (Shah and Singh, 2012). Simple k-mean steps:

- Group the data set in k-groups
- Mark the k-number of clusters with the k-number of centroids
- Assign each data point to the closest centroid
- A number of iterations are performed and k-groups are formulated

K-Mean Algorithm

The k-mean algorithm takes the input data in terms of vector v_i for $i \in n$. Data points are clustered in K number of groups according to similarities in the features set. The similarity measure used in the K mean is the Euclidean distance between the data point and the centroid of the cluster. Initially, K random centroids are selected. Each data point is compared with the selected centroid and the distance is calculated. A data point is assigned in the cluster with a minimum distance from the selected centroid. Then each centroid is updated based upon the average of all the data points associated with that cluster. We have given the data set D of vectors v_i at a given time t. There are K clusters by the groups g_j^t for $j \in K$ with corresponding centroid g_j^t. At each iteration t, the data vector v_i is assigned to the cluster g_j^t. Euclidean distance between vector v_i and centroid g_j^t is dist (v_i, g_j^t). The algorithm assigns each v_i a label $l(V_i^t)$ with respect to the closest centroid that is:

$$l\left(v_i^t\right) = argmin_j \in [K]\left(d\left(v_i, g_j^t\right)\right) \tag{12}$$

In the first iteration $t = 0$, Formulated clusters are used to update the centroid:

$$g_j^{t+1} = \frac{1}{\left|c_j^t\right|}\sum_{i \in c_j^t} vi \tag{13}$$

The average of all the points of each cluster from iteration at $t = 0$ would be the revised centroid. Small threshold τ is used for the convergence such as:

$$g_j^{t+1} = \frac{1}{k}\sum d\left(g_j^t, g_j^{t+1}\right) \leq \tau \tag{14}$$

Algorithm 1: DNS data processing using k-means
Require: DNS domain data 128 mm^3, velocity, mixing ratio, temp
Ensure: Dataset D has vector v_i of feature dimension d
Step 0: BEGIN
Iteration $t \in T$
t = 0
Step 1: Centroid Selection
Clusters $k \in g_j^t$ where $j \in [k]$
Step 2: Distance Calculation
for all g_j^t **do**
while $v_i \leq N$ **do**
if v_i is True **then**
Calculate $d\left(v_i, g_j^t\right)\left\{g_j \in G_j^t\right\}$
end if
end while
end for
Step 3: Cluster Assignment to Datavector v_i (Based on HV and LV regions)
Assign v_i to min $d\left(v_i, g_j^t\right)$ where $j \in g_0^t, g_1^t ... g_k^t$
Step 4: Update Cluster Centroid to $g_0^{t+1}, g_1^{t+1} ... g_k^{t+1}$
Step 5: Repeat Step 2,3,4 until reach convergence
$\tau \leq$ threshold
Step 6: Stop

Running Time of DNS Using K-Mean

Every iteration has a time complexity of $O(knd)$, there are n vectors of dimension d to be compared with k centroids.

DOI: 10.3844/jcssp.2024.344.356

Quantum K-Mean

The k-mean algorithm using a quantum mechanism provides an exponential speed-up for very high-dimensional data. To load n-dimensional input vectors only log *N* qubits are required. Quantum k-mean is performed using subroutines swap test, distcalc and Grover optimization. The swap Test calculates the overlap between two quantum states ⟨state1|state2⟩ based on the measurement probability of control qubit $|0\rangle$. The |state1⟩ and |state2⟩ consist of n-qubit each prepared using amplitude encoding. The outcome of the swap test is used to calculate the distance in the DistCal subroutine. Grover optimization depends on the Grover algorithm used to calculate the closest cluster centroid. Data points are assigned to the cluster and centroids are recomputed by calculating the mean of all the data points in the respective cluster (Wittek, 2014). Quantum k-mean calculates the distance between the high-dimension data points with exponential speedup by encoding N-dimensional classical information by using $\log_2 N$ qubits.

Distance Calculation

Data and centroid matrix is stored in QRAM. Select the initial centroids $g_1^0, g_2^0, \ldots g_k^0$. Calculate the distance between each data point with cluster centroids. To perform the mapping:

$$\frac{1}{\sqrt{n}}\sum_{i=1}^{n}|i\rangle \otimes_{j\in[k]} |j\rangle|0\rangle \mapsto \frac{1}{\sqrt{n}}\sum_{i=1}^{n}|i\rangle \otimes_{j\in[k]} |j\rangle\left|\overline{d^2\left(vi, g_j^t\right)}\right\rangle \tag{15}$$

Assign Minimum Distance Cluster

Estimate the minimum distance between the data point and cluster centroids such as $g_1^0, g_2^0, \ldots g_k^0$. Minimum distance is:

$$\left[d^2\left(vi, g_j^t\right)\right]_{j\in}\left[g_1^0, g_2^0 \ldots g_k^0\right] \tag{16}$$

Centroid State Formation

Label register provides the centroid state as:

$$\left|S_j^t\right\rangle = \frac{1}{\sqrt{\left(g_j^t\right)}}\sum_{i\in g_j^t}|i\rangle \, with\ prob. \frac{\left|g_j^t\right|}{n} \tag{17}$$

Update the Centroids

For time $t = 1$, perform state tomography for the states $\left|g_j^{t+1}\right\rangle$. Update the QRAM data structure for the next iteration with new centroid vectors $c_1^{t+1}, c_2^{t+1} \ldots . c_k^{t+1}$.

Running Time of DNS Using Quantum K-Mean

The running time of Quantum k-mean is estimated as $O\left(\left(\frac{k}{\epsilon}\right)\log nd\right)$. ε is the desired accuracy, n is the number of vectors and d is their dimension. ε_0, ε_1, ε_3 and ε_4 are the errors calculated at every level of the algorithm.

Objective of Unsupervised Learning

Our goal is to examine the differences between droplets in locations of high and low vorticity. There are always three coordinates used to describe the position of a droplet: *x*, *y* and *z*. The location of the droplet within the specific region is a critical factor in the examination of the droplet. This indicates that we are looking for a droplet inside of an enclosure with a high vorticity. K-mean clustering will look through the boxes with the highest vorticity. The algorithm receives as input the values that are associated with the various vortices. There are a variety of tests that are carried out in order to estimate the number of clusters.

Algorithm 2: DNS data processing using Quantum k-means

Require: DNS domain data 128mm^3, velocity, mixing ratio, temp
Ensure: Dataset D has vector v_i of feature dimension d
Step 0: BEGIN
Iteration t ∈ T, Select initial centroids $g_0^t, g_1^t, \ldots g_k^t$
t = 0
Selection is based on high and low vorticity values.
Step 1: Centroid selection and mapping
Clusters $k \in g_j^t$ where $j \in [k]$

$$\frac{1}{\sqrt{n}}\sum\nolimits_{i=1}^{n}|i\rangle \otimes_{j\in[k]} |j\rangle|0\rangle \mapsto \frac{1}{\sqrt{n}}\sum\nolimits_{i=1}^{n}|i\rangle \otimes_{j\in[k]} |j\rangle\left|\overline{d^2\left(v_i, g_j^t\right)}\right\rangle$$

Step 2: Distance Calculation and Assign the Cluster
for all g_j^t **do**
  while $v_i \le N$ **do**
    if v_i is True **then**
      Calculate $d^2\left(v_i, g_j^t\right)\left\{g_j \in G_j^t\right\}$
    end if
  end while
end for
Select MinDist $\left(v_i, g_j^0\right) j \in [k]$
Create a superposition of all points and their labels

$$\left|S_j^t\right\rangle = \frac{1}{\sqrt{\left(g_j^t\right)}}\sum i \in g_j^n \ with\ prob. \frac{\left|c_j^t\right|}{n}\ |$$

Step 3: Update Cluster Centroid by using tomography
$g_0^{t+1}, g_0^{t+1} .. g_k^{t+1}$
Step 5: Repeat Step 2,3,4 until reach convergence.
Step 6: Stop

Droplet Data Analysis

The major objective is to analyze and contrast the droplet characteristics of the simulated domain's low and high vortex regions. With the help of the quantum computing process, we looked into the following droplet characteristics in both the low and high vortex zones. As a consequence of this, the cloud density, evaporation rate, droplet size, droplet distribution and concentration are all determined. These characteristics are then incorporated into a model that is used for weather forecasting. The initial conditions are the same with the exception of the humidity levels, which are (22% and 85%). (By making reference to the SION format that is connected with pertinent DNS data).

Droplet Size Distribution

Poly-dispersed distribution is Min radius 9.2 µm-maximum 17.6 µm and mono-dispersed distribution is Initial drop radius will be taken is 9.2 µm.

Number of Droplets Tracked in DNS Domain

2.09 million (approx.) and 2.09×16 million on projection.

Average Vorticity

The vorticity threshold is set between 0-43s^{-1} and the Average vorticity is taken as 20s^{-1}.

Distortion Test

This analysis will identify a number of 3D-Cuboids on several dimensions. On targeted coordinates of high vortex regions, the following size cuboids are searched. The size of the cube used for analysis is 2×2×2, 4×4×4 and 8×8×8.

Figure 11 shows distortion test statistics. Estimation for DNS domain labeled as G1. Total data points with vorticity greater than 20 are 41381 (Table 3).

Silhouette Analysis

This analysis is used to find the distance between separating dimensions where high vorticity regions are identified. Symmetric method (same size cuboid) Asymmetric method (cuboid size can vary).

Silhouettecoef Ficient = (a-b)/max(a, b)

Where 'a' represents the average distance between all points in the cuboid and 'b' represents the average distance between all points within the cuboid.

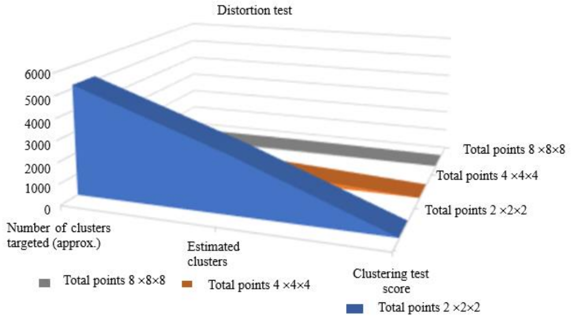

Fig. 11: Distortion test

Table 3: Statistics of clusters estimation

G1 domain	Cuboid size	Number of clusters Targeted (approx.)	Estimated clusters	Clustering test score	Factor affects/ errors
Total points are 41381	2×2×2 4×4×4 8×8×8	5172 646 81	2734 398 69	0.89	Distribution of points, clustering accuracy, computa tional errors Vorticity collision would be more in given cuboid

The silhouette coefficient, often known as the cluster distance, can range anywhere from +1 to -1. A score of one indicates that the cluster is located a significant distance from any nearby clusters. The optimal value for k should be somewhere close to +1.

Analysis of Eulerian Variable in HV and LV

Variables like flow velocity, mixing ratio and temperature are in the Eulerian frame. This means that at every point in the 128×128×128 domain a grid size of 1mm. For analysis of Eulerian variable in HV region mixing ratio and temperature steps for analysis of Eulerian variable:

1. At time 't 'second
2. Take the sorted array of all HV points according to the threshold
3. Take the mean of all mixing ratios and temperature
4. Mean (mixing ratio)
5. Mean (temperature)

Discussion

Several droplet features are compared in this article between high vortical and low vortical areas in three-dimensional regions of turbulent cloudy and clear air interaction. These regions are characterized by the presence of clear air. DNS setup that was identically as the one (Kumar *et al*., 2014) used. The design is more realistic and encompasses the mixing and entrainment processes that are essential to an accurate simulation. A Lagrangian frame is used to trace the location and velocity of droplets, which are represented as particles in the Lagrangian frame. In their study, Kumar *et al*. (2021) investigate both the analysis of cloud droplet attributes using a traditional technique and the examination of high and low vorticity zones in DNS data using unsupervised learning. Both of these investigations are based on learning without supervision. Using traditional techniques to analyze meteorological data presents a number of challenges, the most significant of which is taking care of the complexity and the timely behavior of the data. A quantum method will be utilized in order to identify high and low vorticity zones in a simulated environment, which is the objective of the research that has been presented. In this study, we use a quantum computing technique to compare numerous droplet-related metrics with data acquired from a three-dimensional diffusion network (DNS) of turbulent cloudy and clear air interaction. The DNS has high and low vortical zones. Quantum computing is prone to sophisticated data processing, which is employed for processing extremely dynamic droplet data. This is because quantum computing is being used.

Conclusion

In this particular study, the (128 mm^3) DNS domain was utilized. As a result of segregating the data into two independent regions, we were able to determine that the high vortex zone constitutes fewer than 2% of the total. The findings presented here are consistent with those that (Kumar *et al*., 2014) made public in 2021. The high vortex zone was determined by employing a threshold value of $20s^{-1}$, which is lower than what the researchers (Kumar *et al*., 2021) considered. The first thing that is done is a lookup of the HV and LV droplet data points. The high vortex zones that are produced as a result of this are utilized for unsupervised learning. Quantum k-mean is the modeling technique that is used for the system that searches for clusters with high vorticity. The search for these clusters in the HV region that is accessible is done with cubes of different sizes. It is decided what the average vorticity of each cube will be. After counting the number of clusters that exist inside the provided domain, the total number of clusters is then used to make an estimation, as stated in the result section for droplet analysis. The droplets that make up a cluster are evaluated in this process. In this analysis, the impacts of using both conventional and quantum approaches to solve the problem are studied and analyzed. By analyzing data pertaining to cloud droplets, quantum computers could be utilized to improve the processing of weather predictions. A DNS cloud domain size of 128 mm^3 is utilized in the investigation of the findings of this study. An inquiry into the building of a quantum feature map and a quantum kernel for a certain domain is made possible by the classification of data points into high and low vorticity sectors. The usual difficulties of computing these data are brought to light and the results demonstrate the advantages of using quantum computing. Using this knowledge, it is possible to conceive of larger DNS domains with sizes ranging from 256-2048 mm^3, respectively. When using this projection, users will have an ideal view of the larger clouds. In addition to this quantum mechanism, a classical mechanism's outcomes are compared with those of the suggested problem using the quantum mechanism. The quantum superposition and entanglement process are utilized in the development of the quantum feature set in order to analyze all of the attributes that are present in the data set. In order to prepare a high-dimensional quantum state from the data, classical analysis is performed on it. Quantum advantage was obtained by representing the data on a high-dimensional feature space, while simultaneously producing a quantum feature set composed of classical features. During the processing of data utilizing a quantum mechanism, a number of potential advantages in terms of computations have been examined. In summary, the quantum mechanism can be utilized to accomplish complex computations to study droplet dynamics at an exponentially enhanced pace in order to process the data.

Future Work Direction

Quantum computers have the potential to improve weather forecasting by doing in-depth analyses of data relevant to clouds and precipitation. This investigation makes use of a DNS cloud domain size that is somewhere in the range of 128-2048 mm^3 for the purpose of evaluating the findings. Examining is possible for us since we are actively attempting to improve the precision of our earlier findings. due to the inherent restrictions that come with the use of computation in quantum devices. When it comes to correcting errors, there is a vast range of alternative strategies from which to select. The data on DNS droplets is particularly complicated because they are monitored at a large number of time stamps and with a wide variety of parameters, such as temperature and velocity. This makes the data exceedingly difficult to interpret. It is difficult to accurately forecast the behavior of a droplet using a method that is basic because droplet behavior is incredibly dynamic. Clouds are an integral aspect of the climate, thus future quantum computers will undoubtedly be able to build more accurate weather forecast models by incorporating clouds as a component. This is because clouds are a vital part of the climate. There is the possibility that a number of quantum machine learning algorithms might be presented within the DNS domain in order to make data management more effective. This article presents an analysis of droplets, as well as an investigation into the existence of vorticity zones in the HV region. If we first calculate the vorticity collision rate at a number of different time stamps, then we can potentially use vorticity collision for further modeling and composing of the systems. Within the scope of this study, we investigated and analyzed a number of the droplets' micro-physical characteristics. In order to proceed with the micro-physical feature analysis of droplets, more methodologies need to be researched and investigated in order to understand the extremely dynamic behavior of droplets. In order to do more research on droplet data, we have suggested conducting an analysis of the droplet spectra for a certain DNS domain. The size of the droplets, the total number of them, the distance between them and their radius are going to be researched in both high and low-vorticity regions.

Data and Software Availability

Upon obtaining a request, the Indian institute of tropical meteorology in Pune, India is delighted to provide DNS simulation data for droplet analysis. The expert at IITM for questions regarding DNS data requests is Dr. Bipin Kumar, a senior scientist there. The network Common Data Format (netCDF) serves as the foundation for DNS data. Data for 128 mm^3 of the DNS domain is made available in CSV format and can be accessed through the web address. The DNS data has been made public and can be view. The data was made available.

Source Data

Data is published as it becomes compliant with government regulations. DNS data is made available upon request, however privacy concerns are taken into account.

Acknowledgment

Direct Numeric Simulation (DNS) data is simulated at the high-speed computer facility that is located at IITM Pune. In the course of this inquiry, Dr. Bipin Kumar made a contribution to the collection of data. That data requested in regard to DNS will be supplied by IITM Pune in response to specific requests. During the process of constructing quantum simulators, as well as subsequently, during the process of integrating those simulators with machine learning, the cloud-based capabilities of IBM's quantum platform proved to be extremely helpful. We are grateful to VJTI Mumbai for the support that they offered in a technical capacity over the course of the research that was being carried out and we acknowledge the nature of this research.

Funding Information

This research has been funded by the all India Council by technical education under grant F. no. 8.60/RIFD/RPS/Policy-1/2017-23.

Author Contributions

Mukta Nivelkar: Built the model and the computational framework, data analysis, Conceptualization and composition of methodology, written main document composition of methodology, written main document.

Sunil Bhirud: Built the model and the computational framework reviewed and edited.

Rahul Ranjan: Conceptualization and composition of methodology.

Bipin Kumar: Developed the model and the computational framework, conducted numerical simulation, data generation and analysis, administration reviewed and edited.

Ethics

This article does not contain any human or animal subjects, so the requirement for informed consent is not applicable.

Competing Interests

The authors declare that they have no conflict of interest.

References

Ablayev, F., Ablayev, M., Huang, J. Z., Khadiev, K., Salikhova, N., & Wu, D. (2019a). On quantum methods for machine learning problems part I: Quantum tools. *Big Data Mining and Analytics*, *3*(1), 41-55. https://doi.org/10.26599/BDMA.2019.9020016

Ablayev, F., Ablayev, M., Huang, J. Z., Khadiev, K., Salikhova, N., & Wu, D. (2019b). On quantum methods for machine learning problems part II: Quantum classification algorithms. *Big Data Mining and Analytics*, *3*(1), 56-67. https://doi.org/10.26599/BDMA.2019.9020018

Egger, D. J., Gambella, C., Marecek, J., McFaddin, S., Mevissen, M., Raymond, R., ... & Yndurain, E. (2020). Quantum computing for finance: State-of-the-art and future prospects. *IEEE Transactions on Quantum Engineering*, *1*, 1-24. https://doi.org/10.1109/TQE.2020.3030314

IBM. (2020). Ibm quantum lab. (16). https://lab.quantum.ibm.com/

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

Kumar, B., Ranjan, R., Yau, M. K., Bera, S., & Rao, S. A. (2021). Impact of high-and low-vorticity turbulence on cloud-environment mixing and cloud microphysics processes. *Atmospheric Chemistry and Physics*, *21*(16), 12317-12329. https://doi.org/10.5194/acp-21-12317-2021

Kumar, B., Schumacher, J., & Shaw, R. A. (2014). Lagrangian mixing dynamics at the cloudy-clear air interface. *Journal of the Atmospheric Sciences*, *71*(7), 2564-2580. https://doi.org/10.1175/JAS-D-13-0294.1

Kusyk, J., Saeed, S. M., & Uyar, M. U. (2021). Survey on quantum circuit compilation for noisy intermediate-scale quantum computers: Artificial intelligence to heuristics. *IEEE Transactions on Quantum Engineering*, *2*, 1-16. https://doi.org/10.1109/TQE.2021.3068355

Li, Y., Zhou, R. G., Xu, R., Luo, J., & Jiang, S. X. (2020). A quantum mechanics-based framework for EEG signal feature extraction and classification. *IEEE Transactions on Emerging Topics in Computing*, *10*(1), 211-222. https://doi.org/10.1109/TETC.2020.3000734

Mafu, M., & Senekane, M. (2021). Design and implementation of efficient quantum support vector machine. *In 2021 International Conference on Electrical, Computer and Energy Technologies (ICECET),* 1-4. IEEE. https://doi.org/10.1109/ICECET52533.2021.9698509

Maheshwari, D., Garcia-Zapirain, B., & Sierra-Sosa, D. (2022). Quantum machine learning applications in the biomedical domain: A systematic review. *IEEE Access.* https://doi.org/10.1109/ACCESS.2022.3195044

Pushpak, S. N., & Jain, S. (2022). An implementation of quantum machine learning technique to determine insurance claim fraud. *In 2022 10th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO),* 1-5. IEEE. https://doi.org/10.1109/ICRITO56286.2022.9964828

Schuld, M., & Killoran, N. (2019). Quantum machine learning in feature hilbert spaces. *Physical Review Letters*, *122*(4), 040504. https://doi.org/10.1103/PhysRevLett.122.040504

Schuld, M., Sinayskiy, I., & Petruccione, F. (2015). An introduction to quantum machine learning. *Contemporary Physics*, *56*(2), 172-185. https://doi.org/10.1080/00107514.2014.964942

Schuld, M., Sweke, R., & Meyer, J. J. (2021). Effect of data encoding on the expressive power of variational quantum-machine-learning models. *Physical Review A*, *103*(3), 032430. https://doi.org/10.1103/PhysRevA.103.032430

Shah, S., & Singh, M. (2012). Comparison of a time efficient modified k-mean algorithm with k-mean and K-medoid algorithm. *In 2012 International Conference on Communication Systems and Network Technologies*, 435-437. IEEE. https://doi.org/10.1109/CSNT.2012.100

Wittek, P. (2014). Quantum Machine Learning. *Elsevier Science*.s