IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 1 June 2023, accepted 18 June 2023, date of publication 27 June 2023, date of current version 17 July 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3289924

APPLIED RESEARCH

Quantum Computing to Study Cloud Turbulence Properties

MUKTA NIVELKAR[1], SUNIL BHIRUD[1], (Member, IEEE), MANMEET SINGH[2], RAHUL RANJAN[3], AND BIPIN KUMAR[2]

[1]Veermata Jijabai Technological Institute, Mumbai 400019, India
[2]Indian Institute of Tropical Metereology, Pune 411008, India
[3]Department of Environmental Science, Stockholm University, 10691 Stockholm, Sweden

Corresponding author: Mukta Nivelkar (mmnivelkar_p19@ce.vjti.ac.in)

All India Concil of Technical Education under Grant 8.60/RFID/RPS/Policy-1/2017-18.

ABSTRACT The analysis and investigation of the data obtained from Direct Numerical (DNS) simulation of droplet dynamics in cloud turbulence is a complex and time-consuming task when performaed on traditional computers. The DNS data generally have, a high spatial resolution $\approx 1mm$ and require considerable space to store. It is tedious to find specific features of this data, such as locating high and low vortex areas in cloud turbulence using machine learning algorithms. In this research, we employ quantum computing to examine and analyze cloud droplet dynamics data and present a quantum supervised machine learning algorithm, namely, a support vector machine (SVM) to segregate low and high vortex regions and investigate the droplet characteristics in those regions. The result show that use of quantum computers can accelerate the entire process, and quantum mechanics tools, such as quantum kernels and quantum circuits can better manage the complex nature of data than traditional methods.

INDEX TERMS Quantum computing, quantum machine learning, DNS, cloud droplet, vorticity.

I. INTRODUCTION

In traditional computing, information is stored in a binary form, called bits, which are strings of 0s and 1s. These bits were processed during the computation. Quantum computing is an emerging technology that uses the quantum mechanics and can revolutionize computing strategies in most of the domains compared to conventional computing [1], [2]. Some of the terminology often used in relation to quantum mechanics include qubit, superposition, and entanglement. For a 1-qubit system, the qubit is simultaneously in states 0 and 1, as there are two states in superposition. For a 2 qubits system, the superposition states are defined as 00, 01, 10 and 11. Therefore, the n-qbit system has 2^n states. Multiple states coexist concurrently during the superposition process. Fundamentally, we can say that qubit power is greater than traditional bit power. The Entanglement allows several qubits to pair together at the same time and exponentially accelerate the computing process. Multiple states with weights can be sandwiched between qubit [3], [6], [7]. Although superposition makes it possible to calculate several states simultaneously, it also limits the number of responses that can be obtained for each state. The quantum states described using density matrices and the Dirac notations. An expression for the qubit state is provided in Eq. (1),

$$|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle \tag{1}$$

The relation between α and β according to the Max Bourn Rule using Schrödinger 's wave function is expressed as:

$$|\Psi\rangle = \cos\frac{\theta}{2}|0\rangle + e^{i\phi}\sin\frac{\theta}{2}|1\rangle \tag{2}$$

where $\phi \in [0, 2\pi]$ describes the relative phase and $\theta \in [0, \pi]$ determines the probability measure. $P(|0\rangle) =$

The associate editor coordinating the review of this manuscript and approving it for publication was Wei Huang.

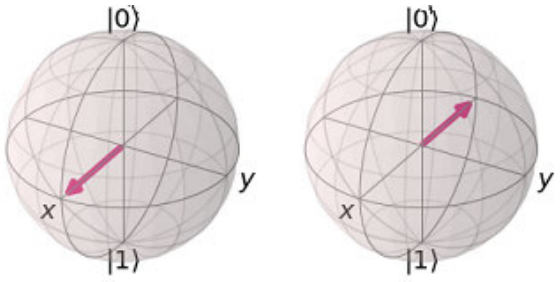

FIGURE 1. Bloch sphere visualization for superposition state.

TABLE 1. Representation of quantum gates for different operations.

QGates		
Hadamard (H)	1 qubit	Superposition
X	1 qubit	X-rotation (Bit Flip)
Y	1 qubit	Y-rotation
Z	1qubit	Z-rotation
AND	2qubit	Performs AND operation
SWAP	2 qubit	Performs Bit swapping
CNOT	2 qubit	Controlled NOT gate
Taffoli	3 qubit	Operates on 3 qubits

$\cos^2\frac{\theta}{2}, P(|1\rangle) = \sin^2\frac{\theta}{2}$. All normalized pure states can be illustrated on the surface of a sphere with a radius $|\vec{r}| = 1$ which is called a Bloch sphere. A Bloch sphere was used to represent a single-qubit. A superposition can be created for state using Eq.(3) and Eq. (4). An example is shown in Figure 1.

$$|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \tag{3}$$

$$|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \tag{4}$$

A. QUANTUM GATES AND CIRCUITS

Simple quantum circuit that uses a few qubits are known as a quantum gates or quantum logic gate [6], [39]. The most prevalent quantum gates operate in one- or two-qubit spaces. In general, 2×2 or 4×4 matrices with orthonormal rows can be used to model quantum gates as matrices. Table 1 shows the commonly used quantum gates and gate implementations in terms of the number of qubits.

Quantum wires connect the quantum gates to form up a quantum circuits. A unitary transformation U, performed by a circuit determines the actual structure of a quantum circuit, the number and types of gates, and the connecting scheme. However, the input and output registers of qubits are also used in the description of quantum circuits, and it should be noted that physically, a quantum circuit's input and output are not separated from one another as their classical counterparts are. This convention enables us to describe the effect of unitary transformation performed by the circuit more coherently.

B. QUANTUM COMPUTING AND MACHINE LEARNING

The differences between classical and quantum machine learning have been discussed in several publications on quantum machine learning approaches [11], [21]. Quantum states are necessary for storing quantum information in quantum random access memory (QRAM). One of the newest disciplines for combining machine learning and quantum computing is quantum machine learning. Classical datasets were processed using a classical-quantum method. The dataset is available in the traditional forms of observation, such as pictures, text, and spreadsheets. These datasets should be transformed into quantum states to be used as input for quantum machines. It is necessary to encode classical data into quantum data states. A quantum computer receives encoded data as input, after which additional processing is required, such as circuit creation and output measurement [4], [32], [33].

1) MACHINE LEARNING

Machine learning algorithms can be classified into two types: supervised and unsupervised. The supervised learning algorithm use labelled examples. By contrast, Unsupervised learning, discovers structures in a sample. To analyze and interpret the data, machine learning searches for patterns in the data. Feynman observed that modelling quantum systems on classical computers becomes impossible as the system size increases, although quantum particles are not constrained in the same manner. The goal of the machine learning model is to learn the pattern from the data to draw inferences. Machine learning requires data such as pictures, videos, spreadsheets and numerical and categorical [34]. The data with N samples can be represented as given in Eq. (5)

$$data = [\vec{X}_1 + \vec{X}_2 + - - - + \vec{X}_n] \tag{5}$$

Sample $\vec{X}_1$ has m features as $\vec{X}_1 = \begin{bmatrix} f_1 \\ f_2 \\ . \\ . \\ f_m \end{bmatrix}$ where $f_1, f_2, f_3 \ldots f_m$ are features of the data. These samples were taken as input and processed in the model as follows,

$$\vec{X} \rightarrow f(\vec{X}; \theta) \rightarrow \hat{y} \tag{6}$$

$\hat{y}$ is output label or predicted value for each sample. The model score is calculated as follows:

$$outcome \leftarrow cost[f(\vec{x}; \theta), True] \tag{7}$$

Mean square error is considered as a metric to evaluate the model performance. Samples are used for supervised learning as, $(\vec{X}_1, y_1), (\vec{X}_2, y_2) \ldots (\vec{X}_n, y_n)$

In the case of linear model input, n samples having n output labels referring Eq. (8). for sample training data

$$\hat{y} = (\vec{X}_1 + y_1) + (\vec{X}_2, y_2) \ldots + (\vec{X}_n, y_n) \tag{8}$$

The cost of training sample is calculated as,

$$cost_{tr} = \frac{1}{N}\sum_{i=1}^{n}(\vec{y_i} - y_i)^2 \tag{9}$$

C. NEED OF QUANTUM MECHANISM IN CLOUD DROPLET DYNAMICS

Numerous industries, including those dealing with land slides, earthquake prediction, and catastrophe preparedness, would greatly benefit from improved weather forecasting. The data used in this study were obtained from the simulation of droplet dynamics in cloud turbulence, which contains complex eddy structures. Owing the complexity of data, the traditional method requires considerable computational resources. Quantum computers can provide a better option for determining how to handle with these data and investigate their complex properties. Quantum computers can model for static and real-time climate data analyse in cloud turbulence [24], [25]. Quantum computing can manage the complex nature of data and can be used for additional analysis to produce findings that are superior to those of traditional calculations. The analysis of cloud data quantifies the changes over time in the droplet characteristics of the cloud [20], [22], [27]. One of the objective is to apply quantum computing to analyze the turbulent properties from droplet dynamics data obtained from the DNS. In this study, we used quantum computing to apply machine learning models to the analysis of cloud droplet data. Superposition and entanglement were used to identify the characteristics of the turbulent eddies. Fault-tolerant quantum machines were employed for this purpose. Encoded classical data are required to generate a parameterized model for a quantum machine learning algorithm, which is then applied to a variational circuit. The use of variational quantum circuits for modelling quantum machine learning and processing quantum information is discussed in the next section [12], [14], [28].

D. CLOUD PROCESSES

Clouds are extremely important for ecosystem. When water vapor condenses, a cloud is created. Meteorologist Luke Howard divided clouds into three groups: cumulus, stratus, and cirrus. Cloud processes range from millimeter to many kilometers. Figure 2 shows the scale of cloud from large to small domains. Cloud data processing can be performed easily using computer simulations of cloud models. The microphysical characteristics governed by numerous dynamic cloud processes involved in cloud formation determine the role of the cloud. Entrainment and detrainment are terms used to describe the passage of ambient air into and out of the clouds, respectively. These activities have the ability to alter droplet features as well as the evolution of the cloud's overall structure Vaillancourt et al. [37]. The dynamics of metres to the submeter scale can be explicitly resolved in cloud models that combine a cloud-scale

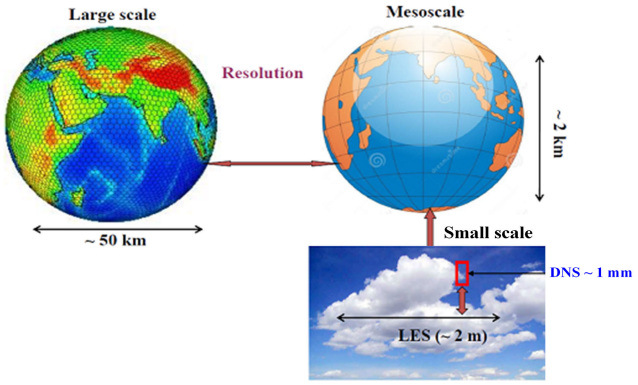

FIGURE 2. Cloud scales rages from Kiolmeters to few meters.

atmospheric flow models with cloud processes. Two types of cloud models that are frequently used: Direct Numerical Simulation(DNS) and Large Edgy Simulation(LES). Small-scale turbulence simulations were performed using DNS. At the edge, where ambient air and cloud droplets interact, additional fascinating characteristics are observed. Droplets that are near dry entrained air are either severely or not impacted. In the natural world, ambient air continuously entrains clouds. Therefore, a numerical model with continuous entrainment and millimeter-scale resolution is ideal for examining the effects of extremely small turbulence on droplet formation [29], [36]. The DNS configuration employed by Kumar et al., which includes entrainment and mixing, can be used to investigate a variety of dynamic cloud phenomena. In this study, we compare several droplet properties between highly vortical and less vortical areas in three-dimensional regions of turbulent cloudy and clear air interaction. We utilized a DNS configuration identical to Kumar et al 2014. The setup is more realistic and incorporates the critical entrainment and mixing processes. droplets are represented as particles, with their locations and velocities traced in the Lagrangian frame. Kumar et. al. studied the analysis of cloud droplet properties based on a classical method, as well as an investigation of high and low vorticities zones in DNS data using unsupervised learning [19]. Handling complexity and its timely behavior is the main challenge when using conventional mechanisms to study meteorological data [19], [26]. The goal of the proposed research was to use a quantum process to locate high and low vorticity zones in a simulated environment. In this work, we compare several droplet-related metrics with a data obtained from three-dimensional DNS of turbulent cloudy and clear air interaction containing high and low vortical regions.

II. SIMULATION DATA

A. DATA AVAILABILITY

Direct numeric simulation (DNS) output data were obtained at the IITM Pune. The data are made available for research purposes upon request [38].

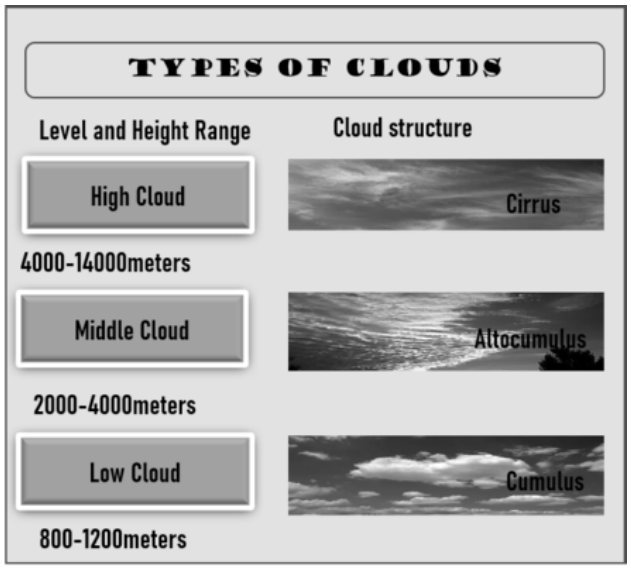

FIGURE 3. Cloud genera based on height and appearance (Image credits: https://scijinks.gov/).

B. DNS DESCRIPTION

DNS data were used to analyze the cloud droplets. Before delving into the data, we describe the many forms of cloud representation and the cloud's structure. The World Meteorological Organization later adopted this classification and currently recognizes ten cloud genera based on their height and appearance. Figure 3 shows various levels of clouds. High clouds in the top panel, middle clouds in center and low clouds in the bottom panel. We used simulation data of the low cloud of cumulus type. These clouds represent fair weather and look like white cotton balls. Clouds play a variety of roles in the atmosphere, ranging from influencing the radiation budget to influencing the hydrological cycle. The role of diverse cloud types is determined by the microphysical features that are governed by the various cloud dynamics processes involved in cloud formation. Non-precipitating cloud types play a significant role in radiation budgeting [37]. Clouds are responsible for the majority of the precipitation that falls on Earth. Many mechanisms can affect precipitation dynamics and cloud longevity, including entrainment, instability, mixing, updraft strength, and moisture supply. The clouds are constantly interact with the surrounding sub-saturated air. Clouds primarily change structure owing entrainment, detrainment and air mixing [19], [38]. Kumar et al. employed. DNS numerical model used in this study. The fluid model is the in Eulerian frame and solves the Navier-Stokes equation for incompressible viscous fluids. Water droplets were considered as Lagrangian points. The particle positions, velocity, and radius were tracked inside a Eulerian grid. The equations were solved in periodic domains of $128mm^3$ with 1mm grid length. For the purposes of this study, a $128mm^3$ domain size was selected. There werel $128*128*128$ points in the entire volume. The entire simulation domain was divided into two parts: a clouded slab with specific-sized droplets and unsaturated air in the rest. The fluid is incompressible which is mathematically represented by Eq.(10) and Eq.(11). 'u' and 'v' are horrizontal velocity components while 'w' represents verical velocity.

$$\nabla.u = 0 \quad i.e. \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0 \tag{10}$$

Momentum equation is refer Eq. (11),

$$\partial_t + u.\nabla = -\frac{1}{\rho_0} + \vartheta\nabla^2 u + Be_z + f_{LS} \tag{11}$$

Here, Eq. (12) is takes care of local temperature change as well as advection.

$$\partial_t T + u.\nabla T = k\nabla^2 T + \frac{L}{C_p} C_d \tag{12}$$

where, ϑ is kinematic viscosity, k is thermal diffusivity of air, L is latent heat for condensation of water vapour, C_p is specific heat for constant pressure. B is Buoyancy.

Figure 4 shows the attributes of the simulation data. Quantitie such as the flow velocities, temperature, water vapor mixing ratio, domain-averaged Turbulent Kinetic Energy(TKE) and TKE dissipation rate were saved as the simulation output.

C. VORTICITY AND ITS MEASURES

The term "vorticity" refers to a three-dimensional vector measure of fluid spins and rotations. The velocity at each point has three components in the Eulerian frame. The vorticity at each grid point was determined using velocity components. The following three factors that make up the vorticity were calculated: Equations (13), (14) and (15) are the associated velocity components.

$$w(i\ component) = \frac{\partial w}{\partial x} - \frac{\partial v}{\partial z} \tag{13}$$

$$w(j\ component) = \frac{\partial u}{\partial z} - \frac{\partial w}{\partial x} \tag{14}$$

$$w(k\ component) = \frac{\partial v}{\partial z} - \frac{\partial u}{\partial y} \tag{15}$$

Magnitude of vorticity at each point is defined as,

$$w_{net} = (w_i^2 + w_j^2 + w_k^2)^{1/2} \tag{16}$$

Figure 5 shows a 3D view of the vorticity. The purpose of this study is to examine cloud droplet properties in high- and low-vorticity cloud turbulence zones. The vorticity magnitude of each grid point is first estimated using Eulerian data, which contain components of velocity in the x, y, and z directions.

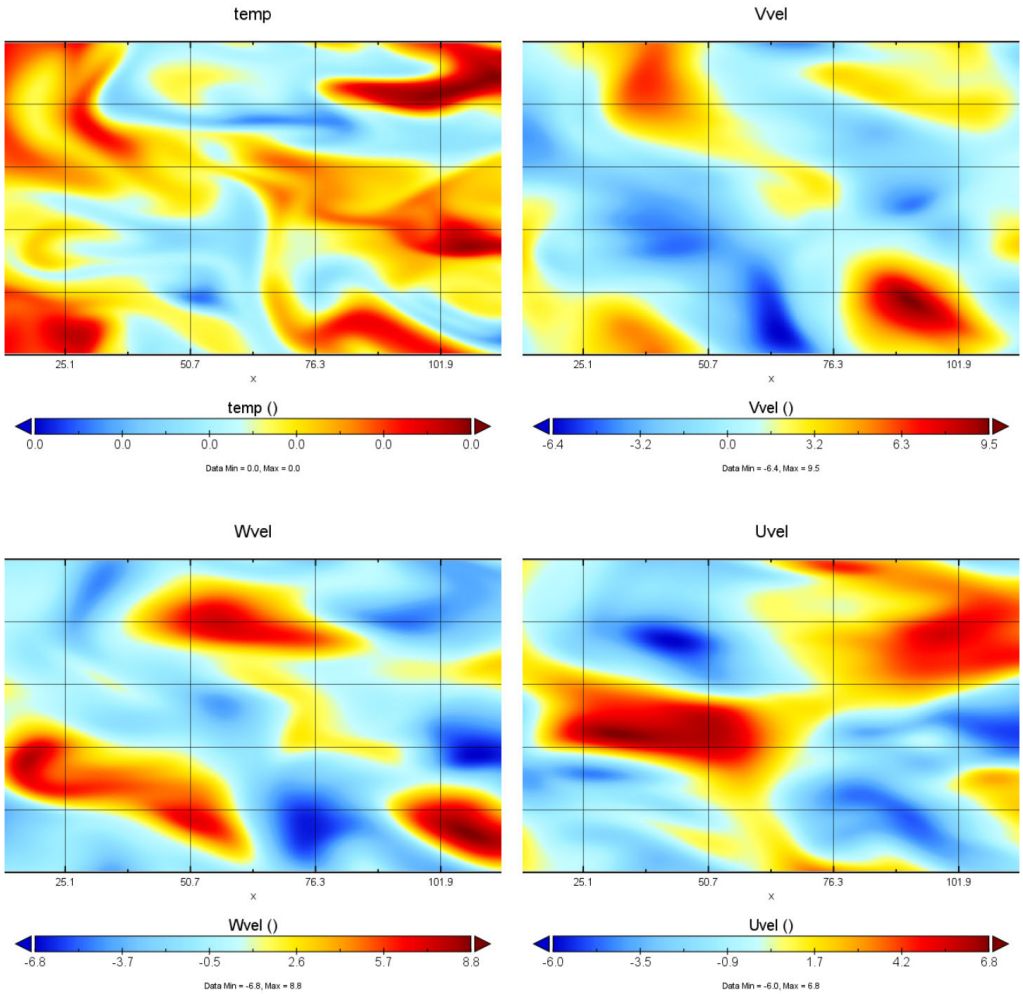

FIGURE 4. Visualization of Uvel, Vwel and Wvel components from simulation data.

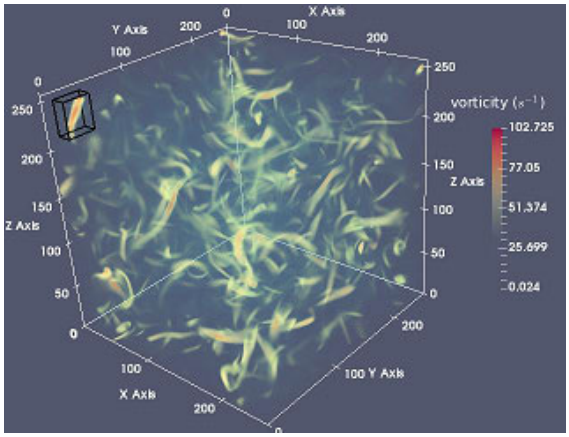

FIGURE 5. 3D view of vorticity.

III. EXECUTION ENVIRONMENT

The computation required for this study was performed using IBM's free and open-source QISKIT [29], [30] quantum software development kit and Python programming language. Quantum circuit can be execute on both local quantum simulator and actual quantum computers. Using an API access key, IBM Quantum Experience offers access to actual quantum computers. The answers described in this work's quantum circuits were initially tested on a local quantum simulator before being run on a quantum computer (quantum_statevector). We also performed experimental work on Qauntum QASM simulator. In order to obtain a reliable estimation of the probabilities of the final quantum state, each run of the circuit was performed 8192 times.

A. NOISY INTERMEDIATE SCALE QUANTUM COMPUTERS

The next generation of quantum computers will include 50- 100 noisy intermediate-scale quantum (NISQ) qubits. These computerscan solve problems considerably faster than traditional computers. Nonetheless, the efficiency of NISQ computers is constrained by several factors. The size of the quantum circuit that can be completed is constrained by noise in quantum gates, and short qubit coherence durations.

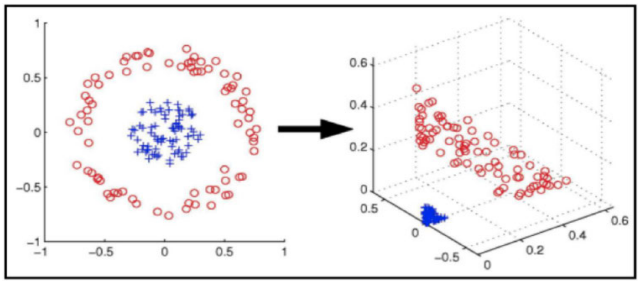

FIGURE 6. Support vector classification on 2D and 3D plane.

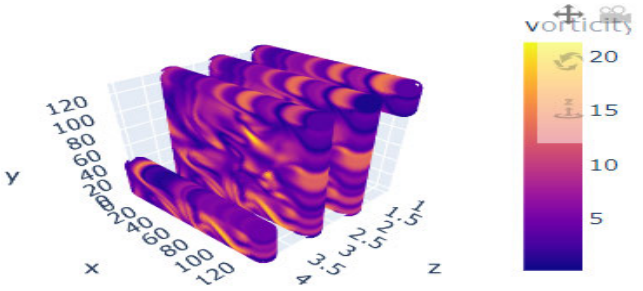

FIGURE 7. 3D scatter plot for droplet datapoints ranging vorticity from 0-20 magnitude.

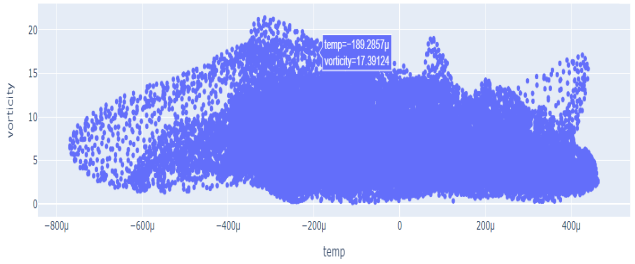

FIGURE 8. 2D Scatter plot for vorticity against temperature feature.

Smaller depth quantum circuits are preferred in light of these restrictions to be safely run on NISQ computers [15], [31].

IV. CML AND QML IN CLASSIFICATION OF DATA

Support vector machines (SVM) are popular supervised techniques for classifying data. Data are represented on a higher dimension using the kernel method so that they may be separated linearly when a function is applied [8], [23], [40].

Given M training points with the following form,

$$(\vec{x_j}, y_j) : x_j \in R^N, R = \pm 1(True\ or\ False) \tag{17}$$

The decision boundary on two hyperplane are created as,

$$\vec{w}.\vec{x} - \vec{b} = \pm 1 \tag{18}$$

Training data classified as,

$$\vec{w}.\vec{x_i} - \vec{b} \geq +1 \ \ if\ y_i = +1 \tag{19}$$

$$\vec{w}.\vec{x_i} - \vec{b} \leqslant -1 \ \ if\ y_i = -1 \tag{20}$$

$\vec{w}$ denotes the decision boundary. The margin is given by two parallel hyperplanes, separated by a distance $\|\frac{2}{w}\|$.

Kernel Matrix is Formulated as,

$$K_{ij} = k(\vec{x_i}, \vec{x_j}) = x_i.x_j \tag{21}$$

The classical performance of SVM is the product of $N_{features}$ and $N^2{}_{samples}$. The time complexity for SVM method was $O(N^3)$. Referto Figure 6 for SVM classifier hyperplane in 2D and 3D planes.

A. HYPERPLANE SEPARATES DATAPOINTS ON HIGHER DIMENSION

Locating high and low vortex points separated by a hyperplane using the support vector machine of the supervised quantum learning method. The algorithm is used to find the hyperplane that separates the high and low vorticity points.

B. VISUALIZATION OF DATAPOINTS

Figure 7 shows a scatter plot of the datapoints in a certain range in 3-dimension. These are 40000 datapoints plotted to view distribution of vorticity. Here, the vorticity ranges from 0-20 magnitude. The value of w ranging from $0 - 200s^{-1}$. The hyperplane separates datapoints on a higher dimension

C. QUANTUM TIME ANALYSIS

Vector and tensor product manipulation in high-dimensional areas is an effective method for quantum computers. The quantum states of $\log_2 N$ qubits may be transferred to the classical data of N-dim complex vectors. Data storage in the qRAM requires $O(\log_2 N)$ steps. The time required for post-processing the quantum data was $O(poly(\log N)$. Large vector inner products and distances was evaluated faster in the quantum domain than in the classical domain.

D. HIGHER DIMENSIONAL FEATURE SPACE

High-dimensional vectors can be handled effectively using quantum computers [33]. d-times tensor product feature map $|\Phi(\vec{x_i})\rangle = |\vec{x_i}\rangle \otimes \ldots . \otimes |\vec{x_i}\rangle$ and the Kernel function X_i and X_j defined as

$$|\Phi(\vec{x_i})\rangle\langle\Phi(\vec{x_j})| = (\langle\vec{x_i}|\vec{x_j}\rangle)^d \tag{22}$$

A non-linear surface in the original space results from the linear hyperplane optimization in the d-times tensor product space.

V. METHODOLOGY

A. DATA COLLECTION AND PRE-PROCESSING

Cloud droplet data was taken from the model output. To improve data clarity, we built quantum feature maps, quantum kernels, and quantum guided data categorizations. As is commonly known, the behavior of droplet data changes with time, making it challenging to manage. Labelled/supervised machine learning is used to process data using a near-term quantum gadget [7]. The text below displays different data attributes.

```
MultiIndex: (0, 0, 0) to (127, 127,
127)
Data columns (total 5~columns):
#    Column   Dtype
---  ------   -----
0    Uvel     float64
1    Vvel     float64
2    Wvel     float64
3    qv       float64
4    temp     float64
dtypes: float64(5)
memory usage: 92.0~MB
```

These numbers represent the fluid velocities measured in the x, y, and z planes. The velocity attributes Uvel, Vvel, and Wvel are depicted in Figure 4. We must normalize the dataset and divide it into training D_{train} and D_{test} samples in accordance with classical classification. We must lower the dimensionality such that the number of qubits we wish to utilize is equal to the number of features in the dataset and scale the range to between -1 and 1, to use the dataset for quantum classification. A classical-quantum model was considered for the machine learning method used in this study. Using a quantum feature map, we encoded classical data in the quantum state space. The essential decision for the selection of the feature map may be influenced by the dataset that has to be classified. In the next section, we take a closer look at the feature maps that Qiskit offers before choosing and tailoring one to encode our data.

B. ENCODING DATA TO QUANTUM STATES

Classical data were used for processing. Converting classical data into a quantum state space using a quantum feature map is called data encoding. This section discuss various feature map design methods that can to be used for data encoding [13]. Feature maps with and without englements are implemented. More quantum benefits on computations were measured using entanglement in the feature map design. The Entanglement is linear and circular. Classical vector x has a feature set $\phi(x)$ in quantum state. $\mathcal{U}_{\Phi(\mathbf{x})}$ is a unitary operation performed during intial state of the qubit. n is the number of qubits used for feature map design. Quantum circuit without entanglement is classically easy to implement; Therefore, if there is no entangling effect on the quantum circuit, there are no quantum benefits. (Referring to Eq. (23))

$$\mathcal{U}_{\Phi(\mathbf{x})} = \left(\exp\left(i \sum_j \phi_{\{j\}}(\mathbf{x}) Z_j \right) H^{\otimes n} \right)^N . \quad (23)$$

To comprehend and analyze droplet dynamics using quantum computing, we have studied cloud droplet data using a quantum feature map under various conditions [10], [14], [16]. A key component of the proposed data analysis is the quantum benefit of handling classically complex data sets. $\phi(\mathbf{x})$ represents quantum feature map from classical feature vector x to the quantum state $|\Phi(\mathbf{x})\rangle\langle\Phi(\mathbf{x})|$. The unitary operation

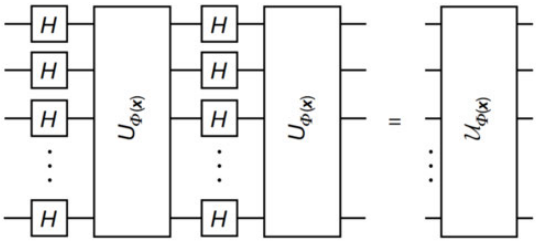

FIGURE 9. Quantum circuit using H gate interleaving with entanglement.

$\mathcal{U}_{\Phi(\mathbf{x})}$ is applied in the starting state $|0\rangle^n$, where n is the number of qubits used for encoding. One can build a feature map on near-term quantum devices, which is challenging to simulate using classical models [39]. These circuits are called short depth circuits on near-term quantum devices, as mentioned in Havlicek et al. [35]. The encoding the classical data in the circuit diagram for r=N is shown in Figure 9.

$$U_{\Phi(\mathbf{x})} = \prod_N U_{\Phi(\mathbf{x})} H^{\otimes n}, \; U_{\Phi(\mathbf{x})} = \exp\left(i \sum_{S \subseteq [n]} \phi_S(\mathbf{x}) \prod_{k \in S} P_i \right) \quad (24)$$

The quantum circuit schematic in Figure 9 uses a layer of Hadamard gates to obtain the superposition of a data feature before interleaving with an entangling box. Entangling effects $U_{\Phi(\mathbf{x})} : P_i \in \{I, X, Y, Z\}$ using Pauli matrices. In other word we can say that, an angle encoding is performed by entangling the given datapoints.

For the given data X rotation is applied to the feature, and n is the number of qubits interleaving with H gate to make following circuit (refer to Eq. (25)).

$$\mathcal{U}_{\Phi(\mathbf{x})} = \left(\exp\left(i \sum_j \phi_{\{j\}}(\mathbf{x}) X_j \right) H^{\otimes n} \right)^d . \quad (25)$$

C. VARIOUS CONDITIONS TO DESIGN QUANTUM PARAMETERIZED CIRCUIT FOR FEATURE SET

1) FEATURE MAP WITH ENGLEMENT

As parameterized quantum circuits, we constructed feature maps [9], [17] as given in the Eq.(26) In this design quantum circuit takes N repeatations with 'FULL' entaglement effect of linear type. The feature dimension is equal to the number of qubits n. The quantum feature maps using linear and circular entanglement are shown in Figure 10 and 11.

$$\mathcal{U}_{\Phi(\mathbf{x})} = \left(\exp\left(i \sum_{jk} \phi_{\{j,k\}}(\mathbf{x}) Z_j \otimes Z_k \right) \exp\left(i \sum_j \phi_{\{j\}}(\mathbf{x}) Z_j \right) H^{\otimes n} \right)^N \quad (26)$$

2) FEATURE MAP USING PAULI GATES

The Pauli gates in the feature map can also be customized;, for instance, P0=X,P1=Y,P2=ZZ:

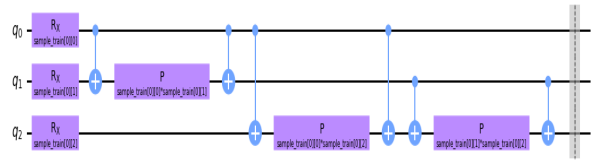

FIGURE 10. Feature map with linear entanglement.

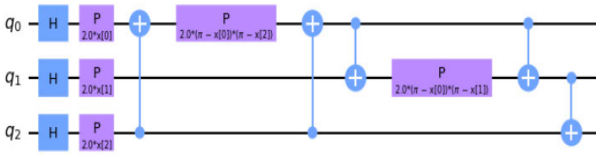

FIGURE 11. Feature map with circular entanglement.

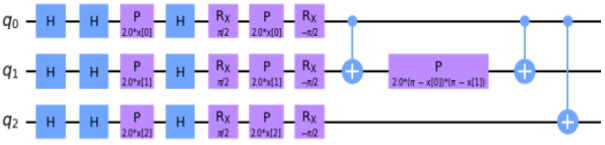

FIGURE 12. Feature map with pauli gates.

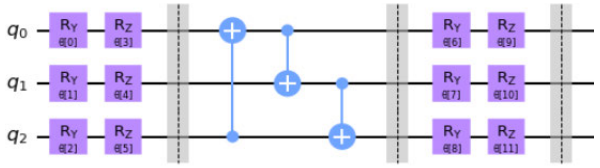

FIGURE 13. Feature map with block rotations.

Figure 12 shows the proposed parameterzed circuit using pauli gate customization.

3) FEATURE MAP USING BLOCK ROTATIONS

In this formation, circuit feature rotations are taken in pairs and also repeated N times. The following circuit block rotation is shown for R_z and R_y rotations. (Refer Figure 13)

4) QUANTUM PARAMETERISED CIRCUIT WITH COMPLETE CUSTOMIZAION

The absence of entanglement in the parameterized circuit indicates that no quantum advantage is gained from the operation. Complete customization of the circuit enables a very accurate design further to handle complex data and benefit from quantum advantages over conventional implementation.

D. QUANTUM PARAMETERIZED CIRCUIT FOR TRANSITION AMPLITUDE BETWEEN TWO INPUT SAMPLES

Figure 14 shows the circuit for any two training samples on a high-dimensional scale to generate a single entry in the kernel matrix. The circuit shows how the layer of Hadamard gate is used to obtain superposition on each feature and, later, how entanlement is used to obtain more quantum benefit on the given DNS data. Classically, these measures have computational hurdles in studying and understanding data. In this section we present how computational challenges are fulfilled using quantum mechanisms in near-term quantum devices [9].

TABLE 2. Quantum classifier performance using different quantum kernel functions.

Classifier's performance			
Linear	Poly	Sigmoid	rbf
0.95	0.95	0.95	0.93

E. FORMATION OF KERNEL AND ESTIMATION

Quantum kernel as a matrix with finite data: $K_{ij} = \left|\langle\Phi^\dagger(\mathbf{x}_j)|\Phi(\mathbf{x}_i)\rangle\right|^2$ [11]. On a quantum computer, the transition amplitude for each component of the kernel matrix can be computed as:

$$\left|\langle\Phi^\dagger(\mathbf{x}_j)|\Phi(\mathbf{x}_i)\rangle\right|^2 = \left|\langle 0^{\otimes n}|\mathbf{U}_\Phi^\dagger(\mathbf{x}_j)\mathbf{U}_\Phi(\mathbf{x_i})|0^{\otimes n}\rangle\right|^2 \quad (27)$$

assuming that the feature map is a unitary transformation, or a parameterized quantum circuit $\mathbf{U}_\Phi(\mathbf{x})$ on n qubits. A quantum kernel matrix can be created, as shown in Figure 15, where a single entry in the kernel matrix is measured using an 8-qubit circuit. The measurement part of the circuit, and the entry is recorded in the kernel matrix.

F. QUANTUM KERNEL FOR TRANING AND TESTING SAMPLES

Figure 16 shows the quantum kernel matrix for the training and testing samples of the DNS data. The training kernel matrix was then filled in for each pair of training data samples, and the testing kernel matrix was filled in for each pair of training and testing data samples. Because each matrix is symmetric, only half of the elements are explicitly calculated to speed up the calculation [18].

Callable Kernel classification test score:0.95

Precomputed Kernel classification test score:0.897

G. SVC AND DATA TRAINING

In this study, we used a supervised quantum machine learning quantum support vector machine algorithm to analyze the droplet data from the DNS output. A quantum kernel was used by the quantum model for each pair of training and testing samples to provide the transition probability. The quantum kernel is a precomputed model that can be referred to as a function of traditional classification techniques. Table 2 displays the test score values for the provided data of several types of classical kernel using the quantum kernel function [5].

H. RUNNING TIME FOR CLASSICAL AND QUANTUM ALGORITHM

Table 3 shows classical and quantum time comparison for DNS data. As DNS dataset has 3D data points for processing. Processing such huge amount of samples is time consuming and challenging too. Modelling using quantum computing on Noisy Intermediate Scale Quantum device is

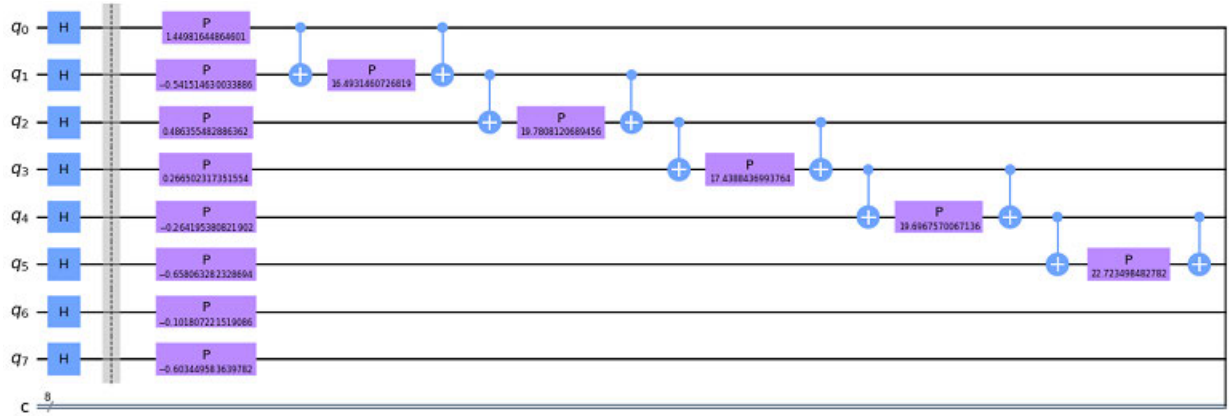

FIGURE 14. Quantum circuit for two parameters input.

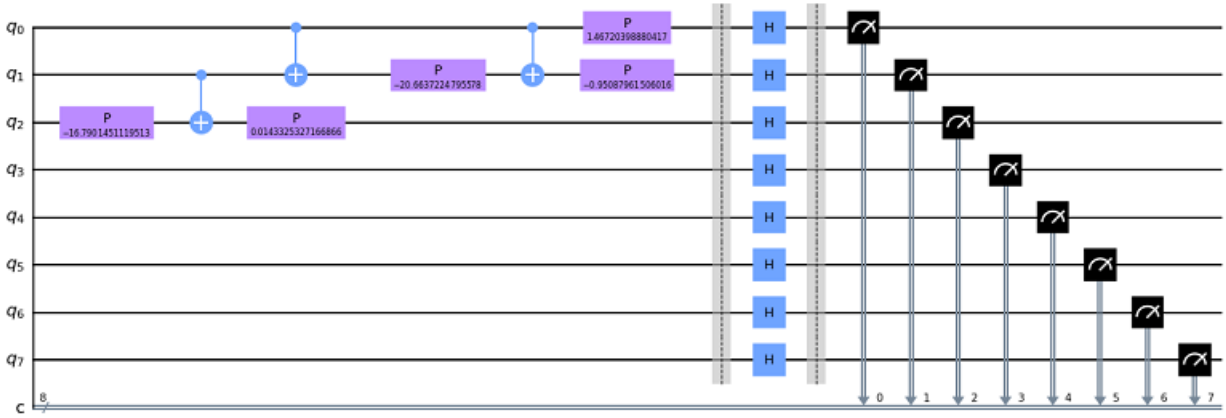

FIGURE 15. Circuit for single parameter measure using two data sample.

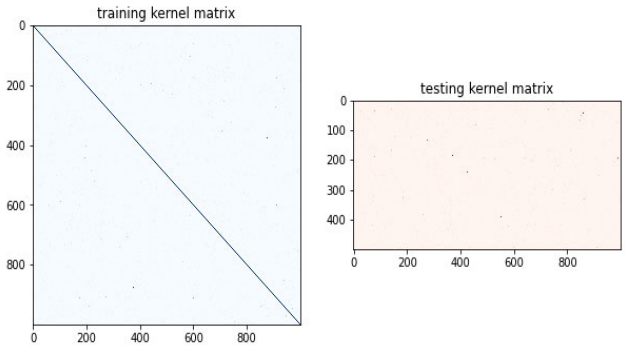

FIGURE 16. Quantum kernel for train and test samples.

another challenge. Sample DNS domain can be partitioned into small domains as test cases.

VI. POSTPROCESSING OF DROPLET DATA

The primary goal was to compare the droplet properties in low and high vortical areas of the simulated domain. We investigated the following droplet propeties in the low and high votrtex regions.

TABLE 3. Running time analysis of DNS data.

Running Time and Classifier's performance			
Classifier	Classical output	QASM simulator	Quantum statevector
Preprocessing Time	0.066	0.035	0.035
Training Time	0.95	0.015	0.015
Prediction Time	0.75	0.032	0.017
Accuracy	0.83	0.95	0.95

1) Drop size distribution (DSD),
2) Evolution of vapor mixing ratio,
3) Variation in U_{rms},
4) Number density (N_d) and
5) Mean radii (R_{mean})

A. CALCULATION OF VORTICITY

In 3−Dimensional vector vorticity, is a measure of the rotation and spin in the fluid. In the Eulerian frame, three components such as x, y and z velocities were used to calculate the vorticity at individual grid points. The magnitude

TABLE 4. High and low vorticity points.

Selection of Vorticity threshold		
Vorticity	Total data-points(2097152)	Percentage
20	12882	0.61
30	714	0.034
35	197	0.0093
40	61	0.0029

of the vorticity at each point is defined as:-

$$w_{net} = (w_i^2 + w_j^2 + w_k^2)^{1/2} \tag{28}$$

The location of the vortices in the form of coordinates must be determines so that the droplet characteristics can be studied. The majority of the simulation domain was populated by low vorticity, with only a small fraction being highly vortical. As in the given domain, the vorticity ranged between 0-43s^{-1}. We have set the vorticity value threshold to be greater than 20s^{-1} to be highly vortical, otherwise low vortical. The following statistics were used to set the vorticity threshold shown in Table 4,

B. ANALYSIS OF EULERIAN VARIABLE IN HIGH AND LOW VORTICITY REGION

Variables such as flow velocity, mixing ratio and temperature are in the Eulerian frame, which means that every point in 128mm∗128mm∗128mm domain has grain size of 1mm(mean 2097152 are in total). Suppose we want to study how mixing ratio varies in high vorticity regions, we can do this as follows:

- Classify the points with high vorticity values based on threshold set for the domain (> 20)
- Identify the coordinates of those points.
- Locate 3D cordinates regions in the available classified data at time 't'.
- Similar steps can be followed to find mean at other time.

C. PROBABILITY DENSITY FUNCTION

The probability density function of the droplet radius is the likelihood of determining a specific drop size in a collection of droplets. The probability of finding a drop with a size between d1 and d2 among a group of 'n' droplets with radii $r1, r2, r3 \ldots rn$ is calculated as follows:

$$P = (Number\ of\ droplets\ having\ radius\ between\ d1\ and\ d2)/n \tag{29}$$

where, n denotes the total number of droplets in the domain. within drop ensembles, we can obtain size ranges between $d2 - d3$, $d3 - d4$ and so on. these individual size ranges are called as 'bins'. If there are N_i droplets in the i^{th} bin, the PDF is calculated as:-

$$P_i = (Number\ of\ droplets\ in\ ith\ bin)/n \tag{30}$$

where P_i is Probability of finding size between i^{th} bin.

VII. CONCLUSION

In this study, we have quantum machine learning to analyze cloud turbulence data obtained from Direct Numerical Simulation (DNS) of cloud droplet dynamics. Because clouds play a crucial role in weather forecasting, a cloud dynamics dataset was selected. Investigation and analysis of these data present a number of issues, including the collection, storage, pre-processing, and analysis of different droplet properties in low and high vortex regions. It is essential to have an understanding of various locations within the clouds because cloud droplets exhibit distinct behaviors in each of these regions, which, in the end, contributes to an improved comprehension of the cloud lifetime.

Because cloud droplet is a tiny particle, understanding its behavior in cloud turbulence using quantum computing presents several challenges. The use of quantum computer mechanisms is helpful for obtaining precise and real-time meteorological data. Quantum machine learning was used to classify fluid data into zones of high and low vorticity. To categorize the data, we used a supervised learning algorithm called the SVM. We investigated a the three-dimensional cumulus cloud and its quantum simulation using quantum mechanics. Feature maps and circuirs were created for the data points provided. Circuits with a parameterized design depending on their characteristics are used to create a variational quantum classifier. To further evaluate the outcome of the classical models, a quantum kernel was created and employed as a technique. We used a $(128mm)^3$ DNS domain for this investigation. After categorizing the data into two different regions, we found that the high vortex region was below than 2% of total amount. This finding is consistent with the results obtained by Kumar et al. [19]. The threshold value for determining the high vortex region was take 20s^{-1} which is smaller than that considered by Kumar et al.. This is because the domain in their study was taken $(256\,mm)^3$, which is 8 times larger, in volume, than the domain considered in this study. These results match those reported by Kumar et al. By collecting more parts of DNS and sewing them together to obtain the outcome for a wide size domain, we can demonstrate this on a vast scale of DNS clouds.

VIII. FUTURE WORK

Weather forecasting can be further processed using quantum computers to analyze cloud, rain, and ocean data. In this study, the findings were investigated using a DNS cloud domain size of 128 mm3. The creation of a quantum feature map and quantum kernel for a specific domain is investigated to accomplish the categorization of data points in the high and low vorticity regions. The traditional difficulties in computing these data are highlighted, and the findings demonstrate the quantum advantages. These results may be projected onto larger DNS domains with sizes ranging from 256mm^3 to 2048mm^3. This projection is beneficial for the study of large clouds.

ACKNOWLEDGMENT

The authors would like to thank the Indian Institute of Tropical Metereology (IITM), Pune, funded by the Ministry of Earth Sciences, Government of India, through the high performance computing (HPC) Facility, Aaditya, IITM, Pune, for data support. They would also like to thank the VJTI Mumbai for technical support on the research topic. Work on quantum simulators and their integration with machine learning was aided by IBM's quantum platform cloud support.

REFERENCES

[1] V. Chauhan, S. Negi, D. Jain, P. Singh, A. K. Sagar, and A. K. Sharma, "Quantum computers: A review on how quantum computing can boom AI," in *Proc. 2nd Int. Conf. Advance Comput. Innov. Technol. Eng. (ICACITE)*, Noida, India, Apr. 2022, pp. 559–563, doi: 10.1109/ICACITE53722.2022.9823619.

[2] F. Ablayev, M. Ablayev, J. Z. Huang, K. Khadiev, N. Salikhova, and D. Wu, "On quantum methods for machine learning problems. Part I: Quantum tools," *Big Data Mining Anal.*, vol. 3, no. 1, pp. 41–55, Mar. 2020, doi: 10.26599/BDMA.2019.9020016.

[3] F. Ablayev, M. Ablayev, J. Z. Huang, K. Khadiev, N. Salikhova, and D. Wu, "On quantum methods for machine learning problems. Part II: Quantum classification algorithms," *Big Data Mining Anal.*, vol. 3, no. 1, pp. 56–67, Mar. 2020, doi: 10.26599/BDMA.2019.9020018.

[4] T. M. Khan and A. Robles-Kelly, "Machine learning: Quantum vs classical," *IEEE Access*, vol. 8, pp. 219275–219294, 2020, doi: 10.1109/ACCESS.2020.3041719.

[5] C. Ding, T. Bao, and H. Huang, "Quantum-inspired support vector machine," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 12, pp. 7210–7222, Dec. 2022, doi: 10.1109/TNNLS.2021.3084467.

[6] P. Wittek, *Quantum Machine Learning*. New York, NY, USA: Academic, 2014, pp. 153–163, doi: 10.1016/B978-0-12-800953-6.00020-7.

[7] M. Schuld and F. Petruccione, *Supervised Learning With Quantum Computers*, vol. 17. Berlin, Germany: Springer, 2018.

[8] A. Shah, M. Shah, and P. Kanani, "Leveraging quantum computing for supervised classification," in *Proc. 4th Int. Conf. Intell. Comput. Control Syst. (ICICCS)*, Madurai, India, May 2020, pp. 256–261, doi: 10.1109/ICICCS48265.2020.9120975.

[9] R. Huang, X. Tan, and Q. Xu, "Learning to learn variational quantum algorithm," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Feb. 28, 2022, doi: 10.1109/TNNLS.2022.3151127.

[10] S. Lloyd, M. Schuld, A. Ijaz, J. Izaac, and N. Killoran, "Quantum embeddings for machine learning," 2020, *arXiv:2001.03622*.

[11] M. Schuld, "Supervised quantum machine learning models are kernel methods," 2021, *arXiv:2101.11020*.

[12] M. Schuld and F. Petruccione, *Machine Learning With Quantum Computers*. Cham, Switzerland: Springer, 2021.

[13] A. Abbas, M. Schuld, and F. Petruccione, "On quantum ensembles of quantum classifiers," *Quantum Mach. Intell.*, vol. 2, no. 1, pp. 1–8, Jun. 2020.

[14] M. Schuld and F. Petruccione, "Information encoding," in *Supervised Learning With Quantum Computers*. Cham, Switzerland: Springer, 2018, pp. 139–171.

[15] S. Maria and F. Petruccione, "Fault-tolerant quantum machine learning," in *Machine Learning With Quantum Computers*. Switzerland: Springer, 2021, pp. 247–272.

[16] M. Schuld and F. Petruccione, "Representing data on a quantum computer," in *Machine Learning With Quantum Computers*. Cham, Switzerland: Springer, 2021, pp. 147–176.

[17] M. Schuld and F. Petruccione, "Learning with quantum models," in *Supervised Learning With Quantum Computers*. Cham, Switzerland: Springer, 2018, pp. 247–272.

[18] T. Tomono and S. Natsubori, "The characteristic of quantum kernel in initial learning process," in *Proc. IEEE Int. Conf. Quantum Comput. Eng. (QCE)*, Broomfield, CO, USA, Sep. 2022, pp. 875–878, doi: 10.1109/QCE53715.2022.00150.

[19] B. Kumar, R. Ranjan, M.-K. Yau, S. Bera, and S. A. Rao, "Impact of high- and low-vorticity turbulence on cloudenvironment mixing and cloud microphysics processes," *Atmos. Chem. Phys.*, vol. 21, pp. 12317–12329, Jan. 2021, doi: 10.5194/acp-21-12317-2021.

[20] S. Manmeet, K. Bipin, R. Chattopadhyay, K. Amarjyothi, S. Anup, R. Sukanta, R. Suryachandra, and N. Ravi, "Machine learning for earth system science (ESS): A survey status and future directions for South Asia," 2021, *arXiv:2112.12966*.

[21] D. A. Zaidenberg, A. Sebastianelli, D. Spiller, B. Le Saux, and S. L. Ullo, "Advantages and bottlenecks of quantum machine learning for remote sensing," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, Jul. 2021, pp. 5680–5683.

[22] K. Bipin, A. Namit, C. R. G. Sandeep, S. Bhupendra, S. Arya, B. Patnaik, G. S. Singh, N. Ravi, and S. Manmeet, "Deep learning based forecasting of Indian summer monsoon rainfall," *Algorithms*, vol. 9, no. 11, 2021, Art. no. 132021.

[23] M. Singh, B. Kumar, S. Rao, S. Singh Gill, R. Chattopadhyay, R. S. Nanjundiah, and D. Niyogi, "Deep learning for improved global precipitation in numerical weather prediction systems," 2021, *arXiv:2106.12045*.

[24] N. Sandeep, P. Swapna, R. Krishnan, R. Farneti, F. Kucharski, A. Modi, A. G. Prajeesh, D. C. Ayantika, and S. Manmeet, "On the weakening association between south Asian monsoon and Atlantic multidecadal oscillation," *Climate Dyn.*, vol. 59, nos. 9–10, pp. 2531–2547, Nov. 2022, doi: 10.1007/s00382-022-06224-1.

[25] A. V. Frolov, "Can a quantum computer be applied for numerical weather prediction?" *Russian Meteorol. Hydrol.*, vol. 42, no. 9, pp. 545–553, Sep. 2017, doi: 10.3103/S1068373917090011.

[26] C. Marshak, I. Yanovsky, and L. Vese, "Energy minimization for cirrus and cumulus cloud separation in atmospheric images," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, Jul. 2018, pp. 1191–1194, doi: 10.1109/IGARSS.2018.8517940.

[27] Z. Tian and S. Baskiyar, "Fake news detection: An application of quantum K-nearest neighbors," in *Proc. IEEE Symp. Ser. Comput. Intell. (SSCI)*, Dec. 2021, pp. 1–6, doi: 10.1109/SSCI50451.2021.9659944.

[28] K. Benlamine, Y. Bennani, N. Grozavu, and B. Matei, "Quantum collaborative K-means," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2020, pp. 1–7, doi: 10.1109/IJCNN48605.2020.9207334.

[29] *Quantum IBM Lab*. Accessed: Jun. 19, 2020. [Online]. Available: https://quantum-computing.ibm.com/lab

[30] V. Frey, R. Rademacher, E. Durso-Sabina, N. Greenberg, N. Videnov, M. L. Day, R. Islam, and C. Senko, "A quantum computing programming language for transparent experiment descriptions," in *Proc. IEEE Int. Conf. Quantum Comput. Eng. (QCE)*, Broomfield, CO, USA, Oct. 2021, pp. 244–254, doi: 10.1109/QCE52317.2021.00042.

[31] I. G. Karafyllidis, "Quantum computer simulator based on the circuit model of quantum computation," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 52, no. 8, pp. 1590–1596, Aug. 2005, doi: 10.1109/TCSI.2005.851999.

[32] M. Nivelkar and S. G. Bhirud, "Quantum computing and machine learning: In future to dominate classical machine learning methods with enhanced feature space for better accuracy on results," in *Intelligent Computing and Networking*, vol. 301, V. E. Balas, V. B. Semwal, and A. Khandare, Eds. Singapore: Springer, 2022, doi: 10.1007/978-981-16-4863-2.

[33] M. Nivelkar and S. G. Bhirud, "Modeling of supervised machine learning using mechanism of quantum computing," *J. Phys., Conf. Ser.*, vol. 2161, no. 1, Jan. 2022, Art. no. 012023, doi: 10.1088/1742-6596/2161/1/012023.

[34] M. Nivelkar and S. G. Bhirud, "Supervised machine learning strategies for investigation of weird pattern formulation from large volume data using quantum computing," in *Advanced Computing and Intelligent Technologies*, vol. 218, M. Bianchini, V. Piuri, S. Das, R. N. Shaw, Eds. Singapore: Springer, 2022, doi: 10.1007/978-981-16-2164-2.

[35] V. Havlíček, A. D. Córcoles, K. Temme, A. W. Harrow, A. Kandala, J. M. Chow, and J. M. Gambetta, "Supervised learning with quantum-enhanced feature spaces," *Nature*, vol. 567, no. 7747, pp. 209–212, Mar. 2019, doi: 10.1038/s41586-019-0980-2.

[36] *Luke Howard, On the Modification of Clouds 1803, Being the substance OFAN Essay Before Askesian Society in the Year 1802-03*. Accessed: 1803. [Online]. Available: https://digital.nmla.metoffice.gov.uk/IO51ce11e6-5ca5-47a0-8fb4-6a6d251ff3d4

[37] P. A. Vaillancourt, M. K. Yau, and W. W. Grabowski, "Microscopic approach to condensational growth of cloud droplets," in *Proc. 12th Int. Conf. Clouds Precipitation*, 1996, pp. 1–13.

[38] B. Kumar, J. Schumacher, and R. A. Shaw, "Lagrangian mixing dynamics at the cloudy–clear air interface," *J. Atmos. Sci.*, vol. 71, no. 7, pp. 2564–2580, Jul. 2014, doi: 10.1175/JAS-D-13-0294.1.

[39] S. S. Gill, A. Kumar, H. Singh, M. Singh, K. Kaur, M. Usman, and R. Buyya, "Quantum computing: A taxonomy, systematic review and future directions," *Softw., Pract. Exper.*, vol. 52, no. 1, pp. 66–114, Jan. 2022, doi: 10.1002/spe.3039.
[40] R. P. Feynman, "Simulating physics with computers," *Int. J. Theor. Phys.*, vol. 21, nos. 6–7, pp. 467–488, Jun. 1982, doi: 10.1007/BF02650179.

MUKTA NIVELKAR received the B.E. and M.E. degrees in information technology from the University of Mumbai, India. She is currently pursuing the Ph.D. degree in computer engineering with the Veermata Jijabai Technological Institute, Mumbai, India.

Since 2007, she has been an Assistant Professor with the Department of Information Technology, Fr. Conceicao Rodrigues Institute of Technology, Navi Mumbai, India. Her research interests include artificial intelligence and machine learning, quantum computing and quantum machine learning, cloud physics, and droplet dynamics analytics using quantum methods.

SUNIL BHIRUD (Member, IEEE) received the M.Tech. and Ph.D. degrees from the Shri Guru Gobind Singhji Institute of Engineering and Technology (SGGSIET), Nanded, India. He is currently the Director and a Professor with the Department of Computer Engineering, Veermata Jijabai Technological Institute (VJTI), Mumbai, India. His research interests include artificial intelligence, machine learning, machine intelligence, social media analytics, digital image processing, and neural networks. His awards and honors dedicate to his excellent work in academia and research.

MANMEET SINGH is currently a Scientist with the Centre for Climate Change Research, Indian Institute of Tropical Meteorology, Pune. He was a Fulbright-Kalam Fellow with the Jackson School of Geosciences, The University of Texas at Austin, in 2021. His research interests include climate solutions to land, ocean, and atmosphere problems using mathematical models, particularly numerical weather prediction systems, AI/ML techniques, causal approaches, recurrence plots, complex networks, and non-linear time series analysis for solving grand challenges in earth system science. He is an experienced climate modeler who has contributed to the IITM earth system model simulations of the IPCC AR6 report. Together with his Ph.D. co-advisor, he developed and coupled the aerosol module of the IITM earth system model. He is active in teaching and has invited talks at venues, such as the NASA/UAH seminar series and Microsoft India podcast. His Ph.D. focused on the impacts of proposals suggesting volcanic eruptions as an analog of solar geoengineering to halt climate change. Recently, his work has showed substantial improvements in high-impact short-range numerical weather predictions using deep learning.

RAHUL RANJAN received the Graduate degree in electrical engineering from the Madan Mohan Malviya University of Technology, Gorakhpur, and the master's degree in atmospheric science from Savitribai Phule Pune University. He is currently pursuing the Ph.D. degree with the Department of Environmental Science, Stockholm University. His master's thesis focused on the impact of turbulence on cloud microphysics. He employed techniques to handle big data generated by direct numerical simulation along with his knowledge of cloud microphysics to untangle the complexity of cloud-turbulence interaction. However, the entirety of understanding cloud microphysics lies in understanding both the aerosol-cloud interaction and turbulence-cloud interaction. This led him to look for a research topic that offered an opportunity to expand his knowledge of how aerosol properties influence clouds. Currently, he is working on process-level modeling of aerosol-cloud interaction (ACI), which is credited as the most significant source of uncertainty in their future climate projections.

BIPIN KUMAR received the M.Sc. degree in mathematics from IIT Kanpur, India, the M.S. (Research) degree in mathematics from NUS, Singapore, and the Ph.D. degree from the School of Computing, Dublin City University, Ireland. His master's thesis in scientific computing. He is currently a Senior Scientist with the High-Performance Computing Systems (HPCS) Division, Indian Institute of Tropical Meteorology, Pune. His background is mainly in computational science. His motivation in computational science led him to further study in this area. He worked on his thesis on high-performance computing with Dublin City University. Apart from these studies, he worked for five years as a Postdoctoral Researcher in Germany, including one year with the Max Planck Institute for Meteorology, Hamburg. His research interests include parallel numerical algorithms, high-performance computing, direct numerical simulation (DNS) for cloud microphysics, the application of artificial intelligence (AI), and machine learning (ML) algorithms to earth science problems.