scientific reports

Check for updates

OPEN

An enhanced approach for predicting air pollution using quantum support vector machine

Omer Farooq[1], Maida Shahid[1], Shazia Arshad[1], Ayesha Altaf[1]✉, Faiza Iqbal[1]✉, Yini Airet Miro Vera[2,3,4], Miguel Angel Lopez Flores[2,5,6] & Imran Ashraf[7]✉

The essence of quantum machine learning is to optimize problem-solving by executing machine learning algorithms on quantum computers and exploiting potent laws such as superposition and entanglement. Support vector machine (SVM) is widely recognized as one of the most effective classification machine learning techniques currently available. Since, in conventional systems, the SVM kernel technique tends to sluggish down and even fail as datasets become increasingly complex or jumbled. To compare the execution time and accuracy of conventional SVM classification to that of quantum SVM classification, the appropriate quantum features for mapping need to be selected. As the dataset grows complex, the importance of selecting an appropriate feature map that outperforms or performs as well as the classification grows. This paper utilizes conventional SVM to select an optimal feature map and benchmark dataset for predicting air quality. Experimental evidence demonstrates that the precision of quantum SVM surpasses that of classical SVM for air quality assessment. Using quantum labs from IBM's quantum computer cloud, conventional and quantum computing have been compared. When applied to the same dataset, the conventional SVM achieved an accuracy of 91% and 87% respectively, whereas the quantum SVM demonstrated an accuracy of 97% and 94% respectively for air quality prediction. The study introduces the use of quantum Support Vector Machines (SVM) for predicting air quality. It emphasizes the novel method of choosing the best quantum feature maps. Through the utilization of quantum-enhanced feature mapping, our objective is to exceed the constraints of classical SVM and achieve unparalleled levels of precision and effectiveness. We conduct precise experiments utilizing IBM's state-of-the-art quantum computer cloud to compare the performance of conventional and quantum SVM algorithms on a shared dataset.

Keywords Air quality prediction, Quantum encoding, Quantum support vector machine, Sustainable environment

The air we breathe in, contains many different gases, including oxygen (21%), nitrogen (78%), argon (1%), carbon dioxide (0.33%), hydrogen (0.33%), helium, and neon[13], but its composition can change depending on factors like latitude, elevation, and pollution. Pollen, ash, dust, and aerosols are major contributors to air pollution, as do harmful gases and particles[6,8,9,25]. Air quality index (AQI) is a metric that calculates air quality rating at a given time and location. Particulate matter (PM2.5/PM10), sulfur dioxide (SO2), nitrogen dioxide (NO2), carbon monoxide (CO), ammonium (NH4), ozone (O3), and lead (Pb) are some other matrices, used in the existing literate to show air quality[31]. A reading of 50 or below on the AQI is regarded as "excellent" and poses little harm to human health. A level between 51 and 100 on the AQI is considered "moderate" and can have a negative impact on sensitive populations. For levels ranging between 101 and 150, AQI is considered "unhealthy for sensitive groups", and for levels between 151 and 200, the AQI is considered "unhealthy". The threshold of 201 to 300 is "extremely unhealthy" and an AQI of above 300 is regarded as "dangerous"[31].

When the AQI is high, people should stay inside with the windows and doors closed and use air purifiers to protect themselves from the harmful effects of air pollution. Climate variables such as temperature, wind

[1]Department of Computer Science, University of Engineering & Technology, Lahore 54890, Pakistan. [2]Universidad Europea del Atlantico, Isabel Torres 21, Santander 39011, Spain. [3]Universidad Internacional Iberoamericana Arecibo, Puerto Rico 00613, USA. [4]Universidad Internacional do Cuanza, Cuito, Bie, Angola. [5]Universidad Internacional Iberoamericana, Campeche 24560, Mexico. [6]Instituto Politecnico Nacional, UPIICSA, Mexico City, Mexico. [7]Department of Information and Communication Engineering, Yeungnam University, Gyeongsan 38541, Republic of Korea. ✉email: ayesha.altaf@uet.edu.pk; faiza.iqbal@uet.edu.pk; imranashraf@ynu.ac.kr

speed, humidity, precipitation, and sun radiations must be studied in conjunction with air quality monitoring in order to determine the specific chemical reactions occurring in the air[15]. The World Health Organization (WHO) reports that 99.99 percent of people around the world breathe polluted air[10]. Since it can penetrate far into the lungs, PM 2.5 poses a particular health risk since it contains harmful particles like viruses. Prolonged exposure to PM 2.5 has been associated with an increased risk of respiratory and cardiovascular diseases like asthma, bronchitis, heart disease, stroke, and, more recently, COVID-19 (viral particles). In conclusion, due to its potential adverse health impacts, PM 2.5 is a major factor in the AQI. Manual methods such as gravimetric, optical microscopy, beta attenuation, and elemental analysis can be used to quantify PM 2.5, but it's clear that due to new technological revolution, new monitoring of air quality has begun[5,17,27,30,32].

PM 2.5 can be predicted using machine learning methods due to their ability to manage large and complex datasets. Machine learning models can learn from patterns and relationships in data, and produce accurate predictions, all of which have received a great deal of attention in recent years due to the rapid development and adaptation of artificial intelligence and machine learning techniques. The development of faster, more efficient, and massively powerful computers made it possible for machine learning algorithms to compute mathematical equations more quickly and efficiently.

Several machine learning methods are used to foretell air quality, for example, artificial neural networks (ANNs)[1,21,23] are commonly used to predict air quality due to their ability to learn and model subtle relationships in data. Support vector machine (SVM)[12,18–20] supervised learning algorithms of the SVM type have been applied to air quality prediction. Random forest (RF)[11,35] is an ensemble learning technique that combines many decision trees to make a forecast. K nearest neighbor (KNN)[24,28] is a popular and easy-to-use machine learning approach for forecasting air quality. These works are just a few examples of how machine learning algorithms have been used to make air quality predictions. The success of machine learning algorithms is highly dependent on the kind, quantity, and quality of the provided data, as well as the nature of the problem being solved. Machine learning is useful for forecasting PM2.5 levels and air quality.

In recent years new era of quantum computing has emerged and quantum machine learning has become a promising area of study that can pave the way for breakthroughs in many other branches of science and technology[3]. Combining quantum mechanics and machine learning, quantum machine learning has become a new paradigm[16]. This new field of study has the potential to significantly improve computing speed and efficiency, as well as fundamentally alter how difficult data analysis problems are approached. Recent quantum machine learning research has centered on developing new algorithms and methods for increasing the efficiency of quantum computing systems. These efforts have led to the development of quantum-inspired machine learning models that have proven effective in a variety of contexts, including image and speech recognition, natural language processing (NLP), and drug discovery. In addition, quantum machine learning has the potential to address problems that are currently beyond the capabilities of conventional machine learning methods, thereby presenting new research avenues[2,4,14].

The potential of quantum SVM to address classification issues in established industries like economics, medicine, and ecology has been drawing increasing interest in recent years. Quantum SVM has also been used to improve the accuracy and efficiency of classification models for air quality data, which has been applied to the field of air quality research. Classification of air quality using the AQI[34], for tracking air pollution's impact on human health, with the help of a quantum SVM. Typical convolutional neural network Utilizing quantum computing, the network model is benchmarked and compared with a quantum quanvolutional neural network, showing a striking improvement in detecting air quality[33]. Another hybrid approach for forecasting air quality that takes into account climate-related parameters and uses feature selection and an evolving interval type-2 quantum fuzzy neural network (eIT2QFNN) is[29].

A review of the existing literature indicates that air pollution in the environment plays a significant role in the spread of pandemics such as the Covid-19 virus; therefore, it is necessary to determine the relationship between air pollution factors such as PM10, PM2.5, NO2, O3, CO, and SO2 and Covid-19 spread. In addition, the study reveals that new pandemics, such as COVID-19, have a significant impact on human lives, and we are not prepared for such pandemics. WHO collects data on COVID-19-infected areas of the world, and the air quality data[31] can be used to train the system using quantum enhancement of machine learning for the COVID-19 pandemic and predict the future spread of COVID-19 in these areas. Using preprocessed air pollution data, these can be trained and used to predict the future spread of air pollution.

This gap has been attributed to a lack of research into the application of quantum SVM to data on regional air quality. The purpose of this study is to predict how environmental factors affect air pollution levels using quantum machine learning. The primary objective of this research is to improve quantum SVM over traditional SVM performance on the world air quality (WAQI) dataset[31] containing air quality factors PM 2.5 and temperature in selected cities. The project is a non-profit project that was started in 2007 for the awareness of people covering 130 countries and 2000 cities. This study makes the following contributions

- Classification of AQI levels into different groups including good and satisfactory using machine learning.
- Implementation of quantum SVM on IBM lab to analyze its performance for air quality prediction using the WAQI dataset.
- Performance comparison between traditional SVM and quantum SVM concerning air quality assessment.

The rest of this paper is organized as. The proposed methodology, the dataset used in this study, and the working process of quantum computing are explained in section "Proposed solution". Section "Results" discusses the experimental results while the conclusion is provided in section "Conclusions".

Proposed solution

In this research, we use the quantum machine learning technique of quantum SVM to predict air pollution from environmental factors. To do this, we benchmark the results using classical SVM and accumulate data from the WAQI dataset[31]. A flowchart is presented in Fig. 1 to visually depict the implementation of both classical and quantum techniques that utilize the SVM model.

Data preparation

Dataset collection

The first step in the adopted methodology is acquiring the WAQI[31] dataset. It is followed by training both classical and quantum SVM models using the WAQI dataset and then comparing the results. The dataset has "Date", "Country", "City", "Species", "Count", "Minimum", "Maximum", "Median", and "Variance" columns that make up the 2020-03-06 dataset, as shown in Table 1.

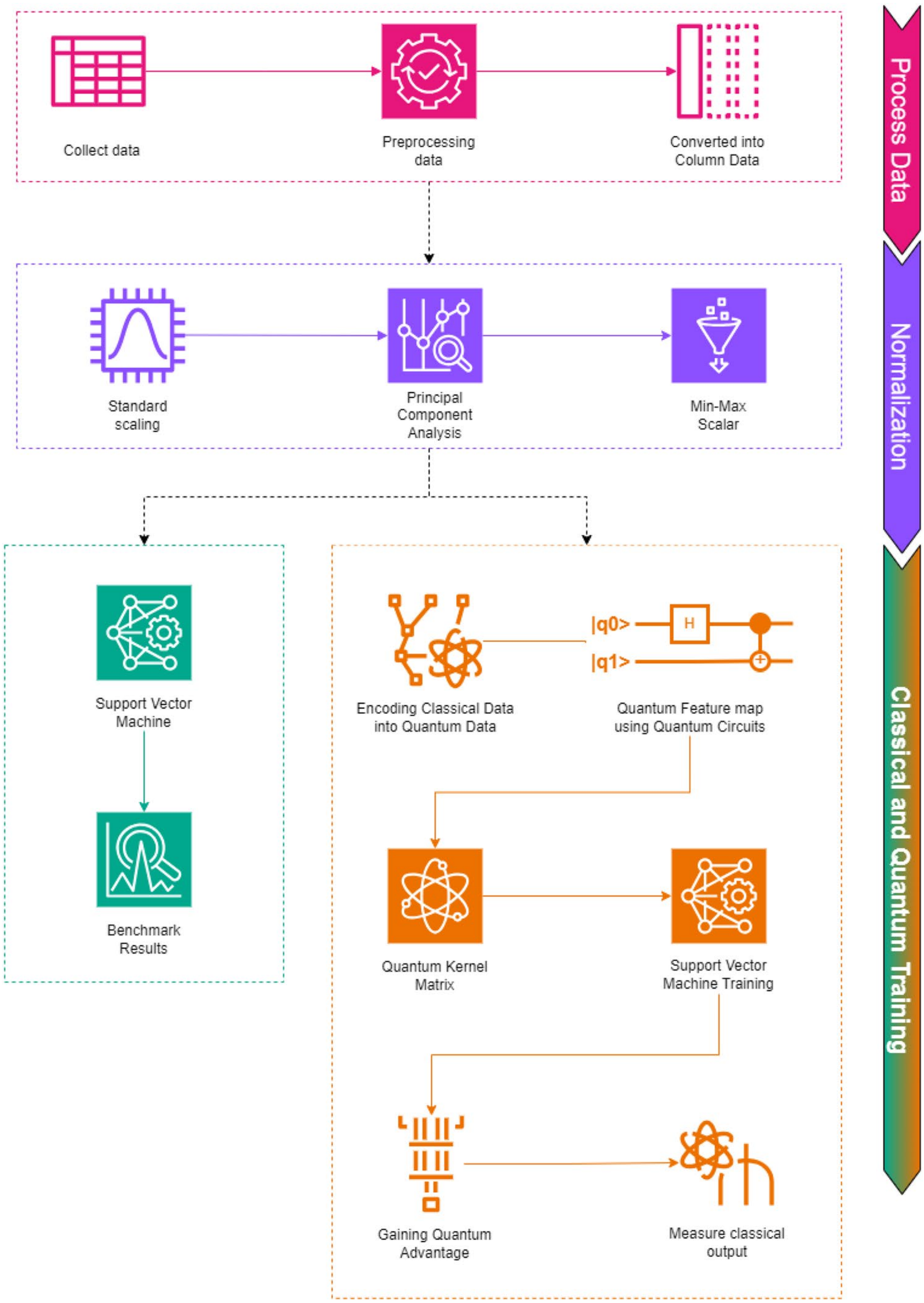

Figure 1. Flowchart illustrating the implementation of both classical and quantum approach utilizing SVM.

Date	Country	City	Species	Count	Minimum	Maximum	Median	Variance
2/5/2022	UG	Kampala	wind-gust	2	11.8	15.4	11.8	64.8
2/25/2022	UG	Kampala	wind-gust	2	12.8	12.8	12.8	0
1/9/2022	UG	Kampala	wind-gust	3	12.8	12.8	12.8	0
1/31/2022	GB	London	Co	96	1	15.7	3.1	176.17
2/7/2022	GB	London	Co	96	9.2	21.7	13.3	122.35
2/8/2022	GB	London	Co	96	8.3	11.9	9.8	11.85
2/26/2022	GB	London	Co	168	0.6	8.3	4.4	49.63
2/16/2022	GB	London	pm25	439	1	114	25	2590.35
2/19/2022	GB	London	pm25	471	2	85	25	2274.02
1/19/2022	GB	London	pm25	505	2	89	49	2871.53
2/3/2022	GB	London	pm25	437	2	74	25	1483.11
1/25/2022	GB	London	pm25	459	25	137	78	3631.95
2/12/2022	GB	London	pm25	488	5	72	34	1683.76
1/30/2022	GB	London	pm25	453	3	91	36	1633.97
2/13/2022	GB	London	pm25	481	5	78	31	1657.11
2/21/2022	GB	London	pm25	487	2	91	38	2339.31

Table 1. A portion of the dataset containing more than 400,000 rows.

The "Date" column of collected data indicates the date of collection. The two-digit country code, like GB, can be found in the "Country" field. City includes the name of the city, while "Species" details the several air quality categories, such as "aqi," "co," "dev," "humadity," "mepaqi," "neph," "no2," "03," "pm1," "pm10," "pm25," "precipitation," "pressure," "so2," "temperature," "uvi," "wd," "wind-gust," and "wind-speed". The "Count" column shows the total number of records for a single day. The minimal value for that day is displayed in the "Minimum" column. The highest value for the day is given as the "Maximum" column. The "Median" statistic represents the midpoint of the data set, and the "Variance" statistic represents the standard deviation of the dataset. The dataset contains a total of almost 400,000 records.

Transform dataset

As rows of data, the "Specie" column of the dataset contains attributes pertaining to air quality, before we can train the data with air quality, we must transform the data into multiple features and then feed them to machine learning models such as SVM. Therefore, we promote the row data from the "Specie" column to a new column, renaming each feature such as temperature to "temperature min", "temperature max", "temperature median", and "temperature variance" and pm25 to "pm25 min", "pm25 max", "pm25 median", and "pm2 variance", etc., increasing the number of columns from nine to ninety-six. The information regarding the classes is in the final column of the data.

In Table 2, the "Class" column takes into account PM2.5's value because it is the principal component consisting of particles with a size of 2.5 microns, such as Covid-19. Also shown in Table 3 is the AQI scale.

Algorithm 1 outlines the process of transforming a dataset in preparation for training a model.

Date	Country	City pm25_count	pm25_min	pm25_max	pm25_median	pm25_variance	Class
1/31/2022	GB	London 419	1	82	26	1929.76	1
2/7/2022	GB	London 477	3	82	33	1853.91	1
2/8/2022	GB	London 442	1	155	19	1451.03	1
2/26/2022	GB	London 460	2	102	50	2579.49	1
3/1/2022	GB	London 383	2	87	36	2265.96	1
1/24/2022	GB	London 464	5	134	64	3228.14	2
1/25/2022	GB	London 459	25	137	78	3631.95	2
1/26/2022	GB	London 438	21	119	64	2564.84	2
12/31/2021	GB	London 387	3	72	34	2143.69	1

Table 2. Data transformation for a single feature.

Class	Daily AQI Color	Levels of concern
1	Green	Good
2	Yellow	Moderate or fair
3	Orange	Unhealthy for sensitive groups or poor
4	Red	Unhealthy
5	Purple	Very unhealthy
6	Maroon	Hazardous

Table 3. The AQI is a metric used to evaluate air pollution levels and their potential impact on human health. It divides air quality into distinct classes, with each class representing a distinct level of concern.

Require: A dataset with rows of unique air quality properties by date and city and columns of minimum, maximum, median, and standard deviation with AQI as the class. All unique air quality values into columns with minimum, maximum, median, standard deviation, and AQI class.

```
ds ← ReadCSV(dataSet.CSV)
nColHd ← AQIattributes
nDatFrm ← pad.DataFrame([nColHd])
city ← ds['City'].unique()
for for each city: do
    nCity ← get one city
    date ← nCity['Date'].unique()
    for for each date: do
        nDate ← each date
        newDataRow ← [None] * 98
        newDataRow ← averagevalues
        for for each specie: do
            nSpecie ← get one specie
            j ← i * 5
            if nSpecie is not empty then
                newDataRow(j + 3) ← nSpecie(0, 4)
                newDataRow(j + 4) ← nSpecie(0, 5)
                newDataRow(j + 5) ← nSpecie(0, 6)
                newDataRow(j + 6) ← nSpecie(0, 7)
                newDataRow(j + 7) ← nSpecie(0, 8)
            end if
        end for
        newDataFrame ← newDataRow
    end for
end for
writenewDataFrameasCSVfile
```

Algorithm 1. An algorithm for dataset transformation.

Standard scaling

Standard scaling, or standardization, is a tried-and-true data preparation technique for machine learning. Its goal is to ensure that all numbers in a dataset have a standard deviation of one and a mean of zero.

$$z = \frac{x - \mu}{\sigma} \tag{1}$$

For preprocessing, we utilize the standard scaler method, as illustrated in Fig. 2. Before scaling, feature 1 is scaled from − 2.5 to 12.5; after scaling, the same data are scaled from − 2 to 2. Similarly, for feature 2, the range of values from 20 to 120 is scaled between − 1 and 4.

Scaling all attributes on the same scale has several advantages:

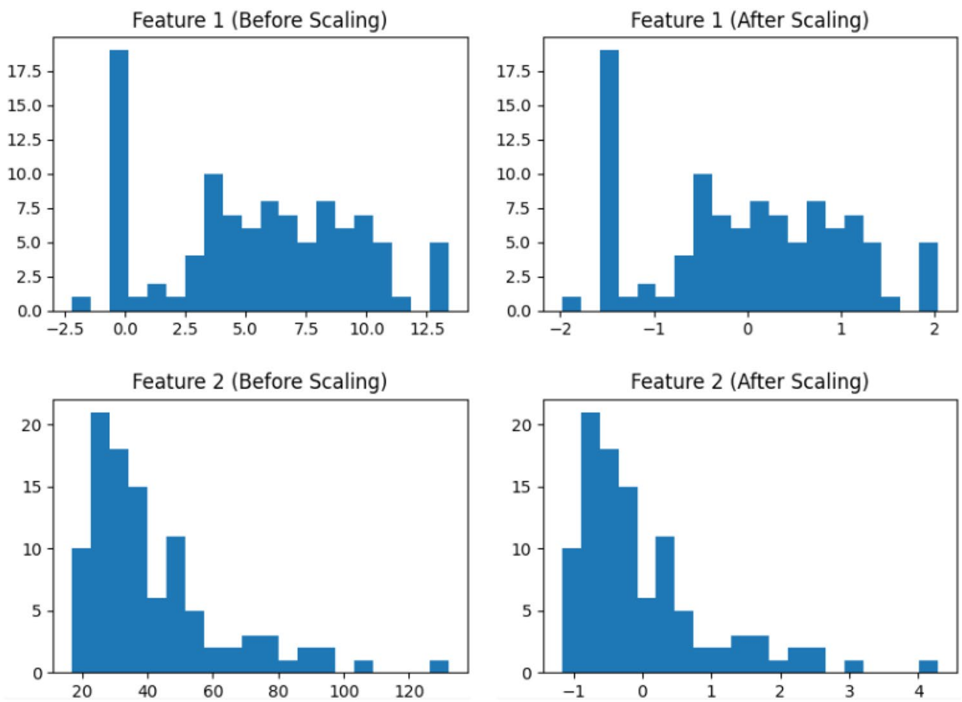

Figure 2. The graph depicts the effects of utilizing standard scaling prior to and following data transformation.

- Due to the varied scales used for each feature, the algorithm could be biased if it gave greater weight to features with higher values. Standard scaling can mitigate this bias by giving all characteristics the same magnitude.
- Supporting algorithms that rely on distance and feature value similarity, such as SVM, which is utilized in this study after preprocessing to ensure that all features contribute equally to distance calculations.

Principal component analysis

Another step in dataset preprocessing is using the principal component analysis (PCA), which rotates the data and attempts to select the top k principal components (eigenvectors) associated with the highest eigenvalues by sorting them in descending order. These p-components capture the data's most significant variance. Figure 3 visualizes the appearance of data before and after PCA is applied.

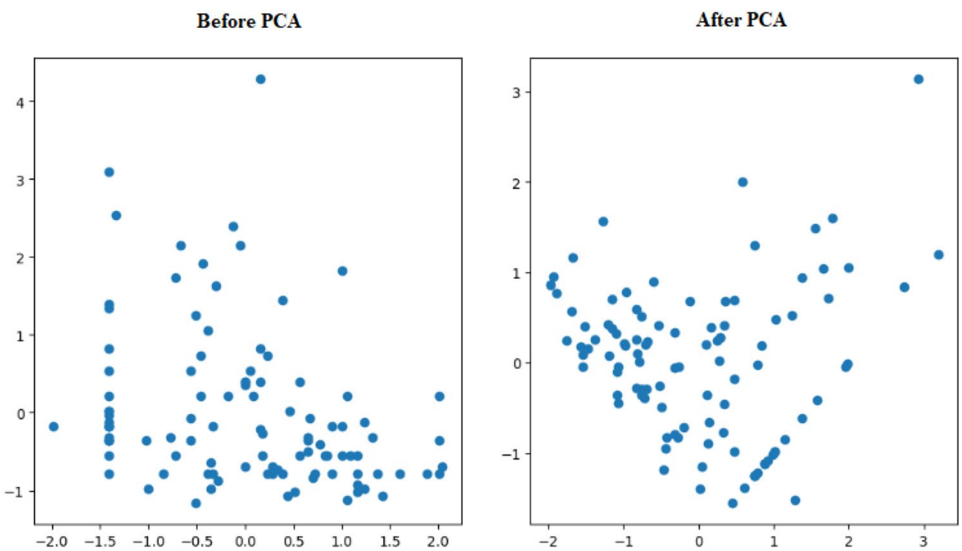

Figure 3. Before and after data transformation, the graph depicts the effects of using the PCA algorithm.

Normalization (Min–Max)
After applying PCA to the dataset, the MinMax function must be used to normalize it. MinMax is used with the following equation

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{2}$$

To improve the performance of the models, a normalization technique is employed to rescale and modify the data. Min-max normalization, also known as feature scaling and min-max scaling, is one such technique. Min-max normalization is used to rescale features of a dataset to a predetermined range between 0 and 1. Figure 4 shows that the data for feature 1 between 2 and 3, is normalized between 0 and 1. Feature 2 data is also within the range 1 to 3, which is normalized between 0 and 1.

Training classical SVM

SVM is a supervised machine learning technique used for regression and classification. It identifies the optimal hyperplane that maximally divides the data into separate classes. As depicted in Fig. 5, support vectors, or the points farthest from the criterion border, are used to establish the hyperplane in the WAQI dataset used in this study. It clearly identifies the hyperplane and support vectors on normalized and 0 to 1 distributed data.

SVM shows good performance but when the number of records and features grow in a dataset, the performance of SVM is affected[7]. The time complexity of SVM is $O(n^2)$ to $O(n^3)$ and an increase in the size of data significantly increases its computational complexity. Algorithm 2 delineates the procedural steps involved in training the conventional SVM model. We benchmark the performance of classical SVM using this algorithm.

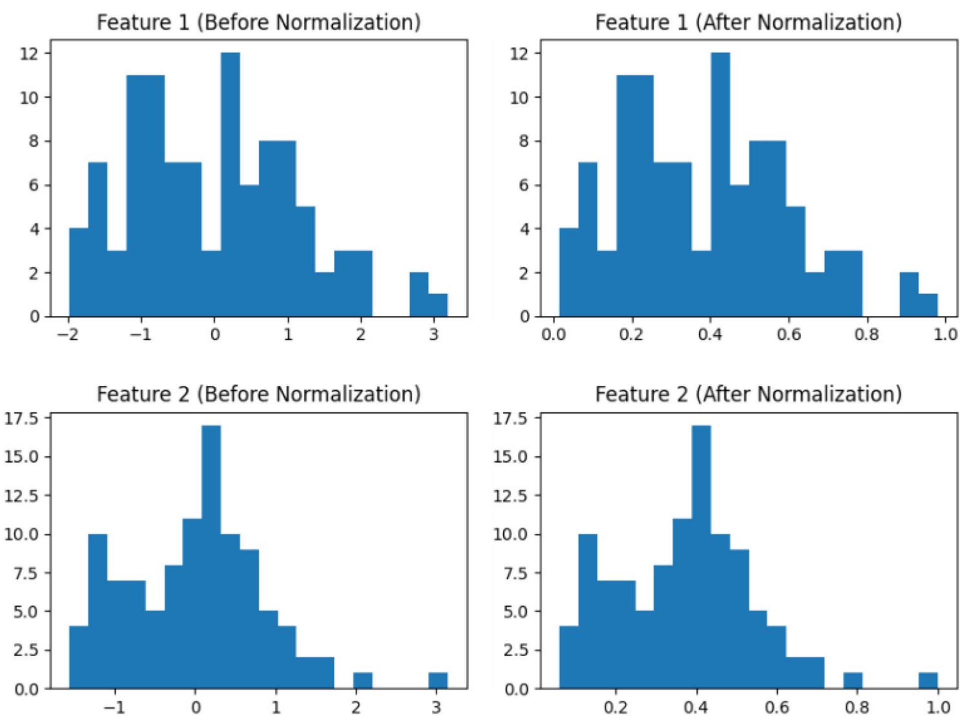

Figure 4. The graph depicts the results of applying Normalization prior to and after data transformation.

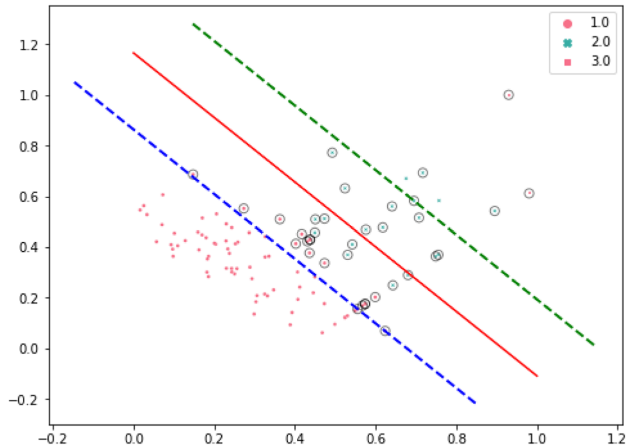

Figure 5. A graphical illustration depicting the separation of data points by a hyperplane following the application of Classical Support Vector Machines (SVM).

Require: Transformed dataset is required for training of classical SVM.
Ensure: Classical SVM results for creating a benchmark for comparison with Quantum SVM.

$dataFrame \leftarrow$ `transformed dataset`
$dataFrame \leftarrow$ `instances for one city`
$x \leftarrow$ `load attributes of one city`
$y \leftarrow$ `load class column`

$scalerX \leftarrow$ `Apply StandardScaler() on x`
$pcaX \leftarrow$ `Apply PCA() on scalerX`
$minMaxX \leftarrow$ `Apply MinMax() on pcaX`

$X, Y \leftarrow trainTestSplit(minMaxX)$
$classifier \leftarrow$ `SVM Kernel`
$fitting(classifier)$
$predictY \leftarrow classifier.predict(testX)$

Algorithm 2. Benchmark using classical SVM.

Training quantum SVM

To classify the data using a quantum computer called quantum classification we used the quantum SVM. After training on classical SVM, now we focus on using the same dataset with quantum SVM. Due to its potential to take advantage of quantum processing benefits, quantum SVM may outperform classical SVM in some use cases[26]. Quantum computers are in their phase of evolution and only a few companies build such machines like IBM, Google, DWave, etc. IBM provides real and simulation of quantum computers for public use cloud technology. So, we implement quantum SVM using IBM Qiskit (open source SDK works with quantum computers). To achieve quantum classification we followed three steps.

1. Convert classical data into quantum data (quantum encoding)
2. Process the quantum data.
3. Measuring the results.

Figure 6 illustrates the process through which a quantum computer operates on a given dataset.

Quantum encoding
We have classical data that we would like to run on a quantum computer. We lack input data collected from quantum devices, so we rely on original classical data. Figure 7 depicts four types of data. classical data executed on classical devices (CC), classical data executed on quantum devices (CQ), quantum data executed on classical devices (QC), and quantum data executed on quantum devices (QQ). In our case, CQ data is used.

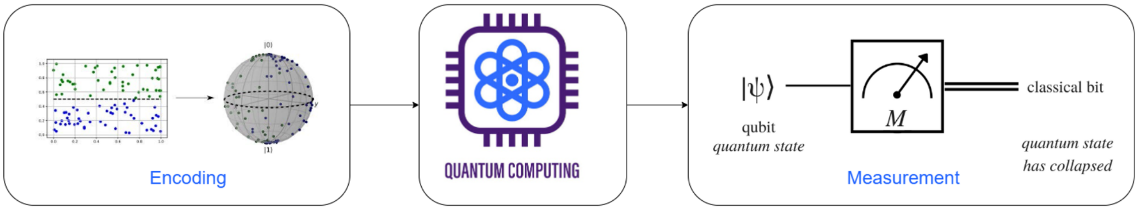

Figure 6. The application of quantum computing on a given dataset is explored in this context.

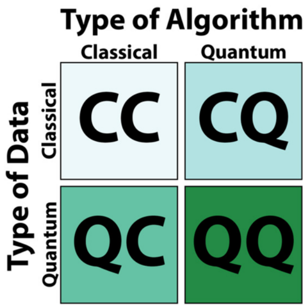

Figure 7. different types of data for quantum computing.

To convert the classical data into quantum data, we use a quantum feature map written as $V(\Phi(x))$, where V is the quantum circuit, and Φ is a classical function acting on classical data x. There are many feature maps available like the Z-Feature Map, ZZ-Feature Map, and Pauli Feature Map. To choose which feature map can be used in our experiment, we need to figure out what feature map best fits on data and what advantage we would like to have from the circuit. The Z-Feature Map only uses a single order gate with no interaction with other qubits. So, we choose ZZ-Feature Map Function 1 which represents two qubits (two data features) and an entangled state between the first and second qubits, as shown in Fig. 8.

Encoding is converting classical data into quantum states in Qiskit's Aqua library ZZFeatureMap is also used to encode classical data into quantum state. ZZFeatureMap constructs a quantum circuit using qubits with controlled Z (CZ) gates. Each qubit represents a single feature in data. Dataset has two features X[0] and X[1] which represent qubits q0 and q1 respectively. The function is used to create a circuit of two qubits-controlled Z gates.

The classical data is incorporated into the quantum state, enabling quantum algorithms to manipulate it within a space of greater dimensions. This feature map is especially valuable in quantum machine learning problems where the interplay between features (qubits) is crucial. This methodology utilizes the principles of quantum mechanics to potentially offer computational advantages for particular workloads in comparison to classical methods.

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

Listing 1: ZZFeatureMap function constructs classical data into Quantum data using qubits

We are utilizing two dimensions, and 'reps' denotes that the circuit is repeated twice with linear entanglement. Each q0 and q1 horizontal line represents two qubits with ?H? representing Hadamard Gate then we have a dot with a vertical line and plus means we use Control Not Gate which is used to Control the target qubit (q1). It acts like a when the $|0>$ ket of q0 then q1 is unchanged and when q0 has $|1>$ then q1 values is changed.

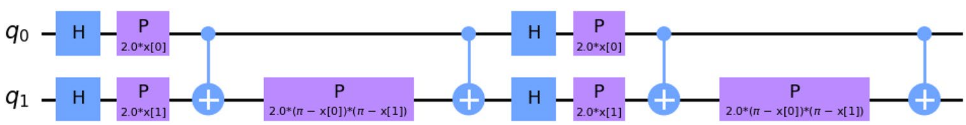

Figure 8. Sample of a quantum circuit designed for a 2-qubit system.

Quantum backend
For selecting a quantum computer among the list of computers provided by IBM, we have to provide the quantum instance and get the backend computer or simulator selected from the list of available computers, as shown in Table 4. The function 2 is used to get the backend.

```
backend = QuantumInstance(
provider.get_backend(
"ibmq_qasm_simulator"),
shots=1024, seed_simulator=seed,
seed_transpiler=seed)
```

Listing 2: Creating an instance of a quantum simulator

To implement quantum SVM with the selected dataset, we choose ?ibmq_qasm_simulator? because it is a general-purpose simulator without the need for error correction and available most of the time without any queues of jobs. Most real quantum computers have queues of jobs and are shared between programmers around the world. The waiting time of the real quantum computer is now getting more and more and one has to wait for his turn to complete the job which will execute on a sharing basis.

To implement quantum SVM with the selected dataset, we choose ?ibmq_qasm_simulator? because it is a general-purpose simulator without the need for error correction and available most of the time without any queues of jobs. Most real quantum computers have queues of jobs and are shared between programmers around the world. The waiting time of the real quantum computer is now getting more and more and one has to wait for his turn to complete the job which will execute on a sharing basis.

Quantum kernel
Quantum kernels utilize quantum feature maps to transform classical data into quantum states. To assess similarity, they calculate the inner product between these quantum states. Quantum kernels utilize the multi-dimensional Hilbert space and principles of quantum mechanics to capture intricate patterns in data. They possess the capacity to offer computational benefits for specific machine learning tasks. Quantum kernels are a burgeoning field of study in quantum machine learning, providing novel methods for manipulating and examining data that exploit the distinctive characteristics of quantum computers.

The quantum kernel algorithm generates a kernel matrix using n-dimensional data points and a feature map. This kernel matrix produced by the quantum machine learning kernels function is advantageous for support vector classification and is discussed in Function 3.

```
kernel = QuantumKernel(
feature_map=feature_map,
quantum_instance=backend)
```

Listing 3: The utilization of a quantum kernel feature map for the purpose of implementing a quantum trick

Some elements of the matrix generated by the kernel function are shown on the provided dataset which clearly shows the correlation between samples that on diagonal there is 1 and above and non-diagonal elements are less than 1. The similarity between any two samples is shown by the off-diagonal elements, with values near to 1 indicating a high degree of similarity in Listing 4.

Name	Qubits	Processor
ibm_perth	7	Falcon r5.11H
ibmq_quito	5	Falcon r4T
ibmq_belem	5	Falcon r4T
simulator_mps	100	Matrix Product State
ibmq_qasm_simulator	32	General, context-aware
simulator_statevector	32	Schrodinger wavefunction

Table 4. IBM offers a range of quantum computer processors that are currently accessible. A few of them are listed here.

```
[[1.009033 0.628345 ... 0.89607 0.976777]
 [0.628345 1.009148 ... 0.62932 0.614605]
 [0.967336 0.681156 ... 0.89539 0.966999]
 ......
 ......
 ......
 [0.590631 0.788227 ... 0.47040 0.654444]
 [0.896077 0.629326 ... 1.00768 0.820342]
 [0.976777 0.614605 ... 0.82034 1.013096]]
```

Listing 4: The list that is a compilation of items that are produced by the utilization of the Quantum Kernel Function.

Quantum superposition

Superposition, the idea that quantum states can coexist in a variety of states at once, is central to quantum computing. Quantum superposition enables qubits to concurrently represent and manipulate many states, resulting in a significant benefit of parallelism in quantum computing. Superposition in the setting of QSVMs allows for the efficient encoding of classical data into quantum states, the concurrent processing of data points, and the effective computing of quantum kernels. This can result in machine learning models that are more expressive and potentially more accurate than their traditional counterparts. In order to potentially speed up computation compared to standard SVM, quantum SVM uses superposition to assess the kernel function for several data points simultaneously. The primary objective of quantum SVM is finding the best hyperplane that segregates the data points in the quantum feature space via the solution of an optimization problem.

A quantum version of the sklearn.svm support vector machine from the scikit-learn function is given in Listing 5. SVC classifier adds the quantum_kernel option which returns the QSVC object.

```
qsvc = QSVC(quantum_kernel=kernel)
```

Listing 5: Quantum version of Kernel function

Training

For training of QSVM, we inherit the SVC fit method 6 and pass the training dataset data and label it as y_train.

```
qsvc.fit(X_train_minmax ,y_train)
```

Listing 6: The class acquires its methods, such as fit, through inheritance from the scikit-learn library.

As soon as the quantum state is measured, as shown in Fig. 9, classical outcomes appear. Based on their distance from this best hyperplane, new input data points are classified.

Algorithm 3 outlines the procedural steps required for training the quantum QSVM model.

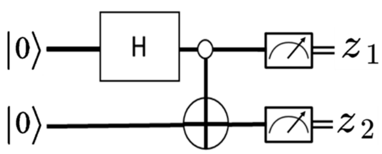

Figure 9. The circuit diagram illustrates the measurement process of a 2-qubit quantum system.

Require: Transformed dataset is required for the training of data on QSVM using a quantum computer.
Ensure: Quantum SVM results for comparing it with classical SVM.

```
import qiskit from IBMQ
seed ← 12345
algorithm_globals.random_seed ← seed
dataFrame ← load transformed dataset
dataFrame ← instances for one city
x ← load attributes of one city
y ← load class column

scalerX ← Apply StandardScaler() on x
pcaX ← Apply PCA() on scalerX
minMaxX ← Apply MinMax() on pcaX

X, Y ← trainTestSplit(minMaxX)
classifier ← SVM Kernel
fitting(classifier)
predictY ← classifier.predict(testX)

featureMap ←
ZZFeatureMap with
dim 2, reps 2 and linear entaglement
IBMQ.enable_account('xxxxxx')
provider ← IBM imbQ provider
backend ← QuantumInstance simulator

kernel ← QuantumKernel feature, backend

airQualitySvc ← SVC kernel
air_quality_svc.fit()
```

Algorithm 3. Implementing quantum SVM.

Results

Experiments were conducted using both a conventional computer and a quantum computer simulator. Both of these experiments produce promising results despite the fact that they can only utilize two classes in their results. Classical SVM and quantum SVM are conducted on both classical and quantum computers, which increases the precision of quantum SVM. In the following section, the outcomes of classical SVM and quantum SVM on classical and quantum computers are elaborated. Experiments are carried out using 80% data as training while the remaining 20% is used for testing the models.

Result using classical SVM

Results were obtained from distinct datasets gathered for London and Lahore.

Result for London city (UK)

The classical SVM model demonstrates a classification accuracy of 91% when applied to two distinct classes, as shown in Table 5. Similarly, precision, recall, and F1 scores are also provided for both classes. Overall, class-wise results show better performance of classical SVM for class 1 (Good).

The reported results are achieved with the assistance of 35 data instances, of which 29 are classified as belonging to class 1 (Good) and 6 are classified as belonging to class 2 (Moderate). The graph presented illustrates the accurate prediction of class 1 (Good) and a partial accuracy in predicting class 2 (Moderate), with 3 out of 6 predictions being right. Figure 10 shows the confusion matrix of the classical SVM.

Result for Lahore city (Pakistan)

The classical SVM model demonstrates a classification accuracy of 87% when applied to two distinct classes, as shown in Table 6. Similarly, precision, recall, and F1 scores are also provided for both classes. Overall, class-wise results show better performance of classical SVM for class 5 (Very Unhealthy).

	Precision	Recall	F1-score	Support
1 (Good)	0.91	1	0.95	29
2 (Moderate)	1	0.5	0.67	6
Accuracy	–	–	0.91	35
Macro avg.	0.95	0.75	0.81	35
Weighted avg.	0.92	0.91	0.90	35

Table 5. The outcome of employing the classical SVM algorithm for the classification of two distinct classes for London city.

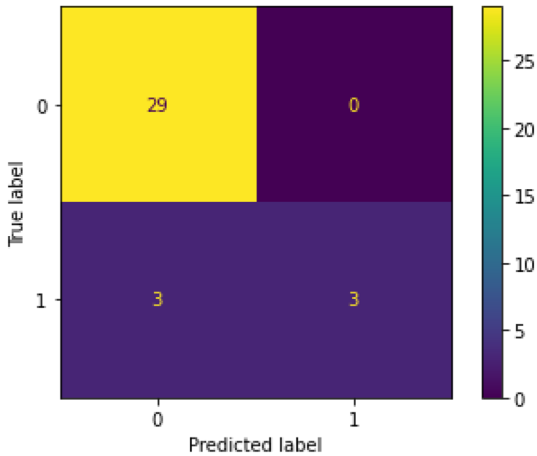

Figure 10. The confusion matrix of the conventional SVM exhibits two unique categories for prediction outcomes, namely "Good" and "Moderate".

	Precision	Recall	F1-score	Support
5 (Very Unhealthy)	0.82	1	0.90	9
6 (Hazardous)	1	0.67	0.80	6
Accuracy	–	–	0.87	15
Macro avg.	0.91	0.83	0.85	15
Weighted avg.	0.89	0.87	0.86	15

Table 6. The outcome of employing the classical SVM algorithm for the classification of two distinct classes for Lahore city.

The reported results are achieved with the assistance of 15 data instances, of which 9 are classified as belonging to class 5 (Very Unhealthy) and 6 are classified as belonging to class 6 (Hazardous). The graph presented illustrates the accurate prediction of class 5 (Very Unhealthy) and a partial accuracy in predicting class 6 (Hazardous), with 2 out of 6 predictions being right. Figure 11 shows the confusion matrix of the classical SVM.

Result on IBM simulator (ibmq_qasm_simulator)

Results were obtained from distinct datasets gathered for London and Lahore.

Result for London city (UK)

The quantum SVM demonstrates a high level of accuracy, reaching 97%, when applied to a binary classification problem involving two classes, as shown in Table 7. Besides better accuracy, precision, recall, and F1 scores of quantum SVM are also better than classical SVM with 1, 0.97, and 0.98 for class 1 precision, recall, and F1 scores and 0.86, 1, and 0.92 for class 2, respectively.

The QSVM model is supported by a dataset consisting of 35 observations. Among these observations, 29 are classified as belonging to class 1 (Good), while the other 6 are classified as belonging to class 2 (Moderate). The graph presented illustrates the accuracy of the predictions for two classes: class 1 (Good) and class 2 (Moderate). It indicates that out of a total of 29 instances in class 1, 28 were accurately predicted. Similarly, for class 2, out of 6 instances, 5 were predicted correctly. Figure 12 shows the confusion matrix of the quantum QSVM.

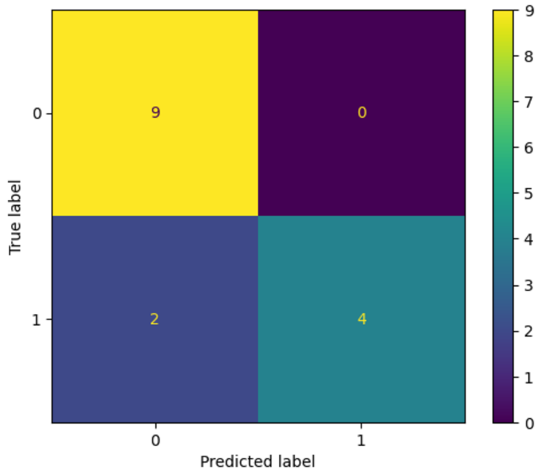

Figure 11. The confusion matrix of the conventional SVM exhibits two unique categories for prediction outcomes, namely "Very Unhealthy" and "Hazardous".

	Percision	Recall	F1-score	Support
1 (Good)	1	0.97	0.98	29
2 (Moderate)	0.86	1	0.92	6
Accuracy	–	–	0.97	35
Macro avg.	0.93	0.98	0.95	35
Weighted avg.	0.98	0.97	0.97	35

Table 7. The outcome of employing the quantum SVM algorithm for the classification of two distinct classes.

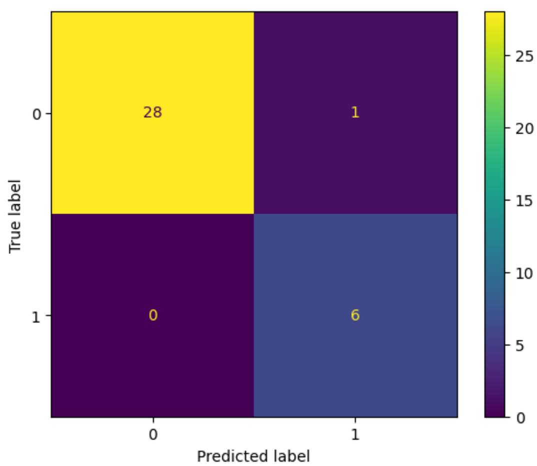

Figure 12. The confusion matrix of the quantum SVM exhibits two unique categories for classical predictions, namely "Good" and "Moderate".

Result for Lahore city (Pakistan)
The quantum SVM demonstrates a high level of accuracy, reaching 94%, when applied to a binary classification problem involving two classes, as shown in Table 8. Besides better accuracy, precision, recall, and F1 scores of quantum SVM are also better than classical SVM with 1, 0.91, and 0.95 for class 5 precision, recall, and F1 scores and 0.95, 0.94, and 0.94 for class 6, respectively.

The QSVM model is supported by a dataset consisting of 35 observations. Among these observations, 16 are classified as belonging to class 5 (Very Unhealthy), while the other 6 are classified as belonging to class 6 (Hazardous). The graph presented illustrates the accuracy of the predictions for two classes: class 5 (Very Unhealthy) and class 6 (Hazardous). It indicates that out of a total of 16 instances in class 5, 10 were accurately predicted.

	Precision	Recall	F1-score	Support
5 (Very Unhealthy)	1	0.91	0.95	10
6 (Hazardous)	0.95	0.94	0.94	6
Accuracy	–	–	0.94	16
Macro avg.	0.92	0.95	0.93	16
Weighted avg.	0.95	0.94	0.94	16

Table 8. The outcome of employing the quantum SVM algorithm for the classification of two distinct classes.

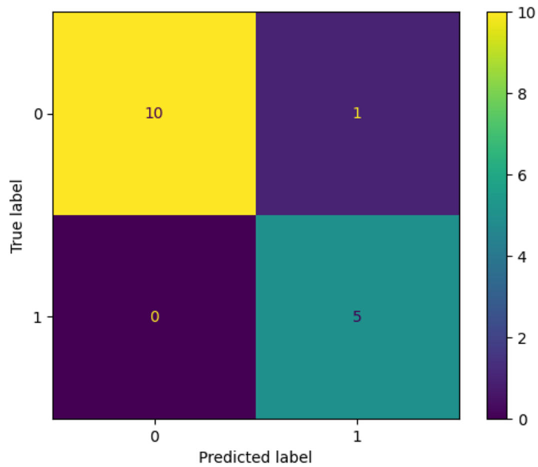

Figure 13. The confusion matrix of the quantum SVM exhibits two unique categories for classical predictions, namely "Very Unhealthy" and "Hazardous".

Similarly, for class 6, out of 5 instances, 5 were predicted correctly. Figure 13 shows the confusion matrix of the quantum QSVM.

In addition to the performance of quantum SVM compared to classical SVM, the computational complexity is another important factor. The time complexity of classical SVM is $O(n^2)$ to $O(n^3)$[7] and the time complexity of quantum SVM is $O(log(n))$[22]. Quantum computers use the power of qubits which can be found in superposition and entanglement which makes them use parallelism to achieve big data tasks more efficiently.

Conclusions

The objective of this study is to analyze the performance of quantum SVM on the WAQI dataset compared to that of conventional SVM. Air quality PM2.5 and temperature are two of the parameters studied in this type of research on specific cities. As quantum computers are still in their infancy, it is recommended to train the dataset using both classical SVM and quantum SVM on the two characteristics of PM2.5 and temperature. We compare and contrast the results of conventional SVM trained with the same features and data as quantum SVM trained with the same features and data. Because quantum computers are in their early stages of development that is why we chose to run our program into a quantum simulator also provided by IBM (ibmq_qasm_simulator). We limit our dataset to two features because of the early development of quantum computers and the limited number of qubits available. After standardization and normalization of data, it is used both with classical and quantum SVM. Results indicate that quantum SVM performs much better than the traditional SVM model. As the number of features increases, quantum computers can perform well in terms of time complexity and accuracy due to the nature of the implementation of parallelism using superposition and entanglement. Classical SVM shows an accuracy of 91% and quantum SVM shows an accuracy of 97%. The study finds that due to the limitation of the early stages of development quantum takes more time but in the near future when qubits are increased with less noise and error corrections quantum computers perform far better results in terms of time and accuracy but this part needs more research in upcoming years.

Data availability

The datasets used and/or analyzed during the present study are available from the corresponding author upon reasonable request

Received: 18 January 2024; Accepted: 7 August 2024
Published online: 22 August 2024

References

1. Agarwal, S. *et al.* Air quality forecasting using artificial neural networks with real time dynamic error correction in highly polluted regions. *Sci. Total Env.* **735**, 139454 (2020).
2. Aithal, P. Advances and new research opportunities in quantum computing technology by integrating it with other icct underlying technologies. *Int. J. Case Stud. Business, IT Educ. (IJCSBE)* **7**(3), 314–358 (2023).
3. Arute, F. *et al.* Quantum supremacy using a programmable superconducting processor. *Nature* **574**(7779), 505–510 (2019).
4. Bindhu, V. Cyber security analysis for quantum computing. *J. IoT Soc. Mobile Anal. Cloud* **4**(2), 133–142 (2022).
5. Carrero, J., Goienaga, N., De Vallejuelo, S.F.-O., Arana, G. & Madariaga, J. Classification of archaeological pieces into their respective stratum by a chemometric model based on the soil concentration of 25 selected elements. *Spectrochim. Acta, Part B* **65**(4), 279–286 (2010).
6. Chen, Y. *et al.* The prospective effects of long-term exposure to ambient pm2.5 and constituents on mortality in rural east china. *Chemosphere* **280**, 130740 (2021).
7. Cristianini, N. & Shawe-Taylor, J. *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods* (Cambridge University Press, Cambridge, 2000).
8. Darrow, L. A. *et al.* Ambient pollen concentrations and emergency department visits for asthma and wheeze. *J. Allergy Clin. Immunol.* **130**(3), 630–638 (2012).
9. Dennekamp, M. & Abramson, M. J. The effects of bushfire smoke on respiratory health. *Respirology* **16**(2), 198–209 (2011).
10. Gupta, N. *et al.* Deleterious effect of air pollution on human microbial community and bacterial flora: A short review. *Int. J. Environ. Res. Public Health* **19**(23), 15494 (2022).
11. Hasnain, A. *et al.* Assessing the ambient air quality patterns associated to the covid-19 outbreak in the yangtze river delta: A random forest approach. *Chemosphere* **314**, 137638 (2023).
12. Hossain, M. T. *et al.* Impact of covid-19 lockdowns on air quality in bangladesh: Analysis and aqi forecasting with support vector regression. In *2023 4th International Conference for Emerging Technology (incet)* 1–6 (2023).
13. Iqbal, M. *Chapter 5-a Cloudless-Sky Atmosphere and its Optics* (Academic Press, UK, 1983).
14. Kapoor, N. R. *et al.* Quantum computing for indoor environmental quality: A leapfrogging technology. In *Handbook of Research on Quantum Computing for Smart Environments* 191–216 (IGI Global, 2023).
15. Kinney, P. L. Climate change, air quality, and human health. *Am. J. Prev. Med.* **35**(5), 459–467 (2008).
16. Kopczyk, D. Quantum machine learning for data scientists. arXiv preprint arXiv:1804.10068 (2018).
17. Kumar, A. *et al.* Detection and identification of shape, size, and concentration of particulate matter in ambient air using bright field microscopy-based system. *Atmos. Pollut. Res.* **2023**, 101913 (2023).
18. Leong, W., Kelani, R. & Ahmad, Z. Prediction of air pollution index (api) using support vector machine (svm). *J. Environ. Chem. Eng.* **8**(3), 103208 (2020).
19. Liu, C.-C., Lin, T.-C., Yuan, K.-Y. & Chiueh, P.-T. Spatio-temporal prediction and factor identification of urban air quality using support vector machine. *Urban Climate* **41**, 101055 (2022).
20. Lu, W. *et al.* Air pollutant parameter forecasting using support vector machines. In *Proceedings of the 2002 International Joint Conference on Neural Networks, ijcnn'02 (cat. no. 02ch37290), vol. 1* 630–635 (2002).
21. Maleki, H. *et al.* Air pollution prediction by using an artificial neural network model. *Clean Technol. Environ. Policy* **21**, 1341–1352 (2019).
22. Outeiral, C. *et al.* The prospects of quantum computing in computational molecular biology. *Wiley Interdiscipl. Rev. Comput. Mol. Sci.* **11**(1), e1481 (2021).
23. Patni, J. C. & Sharma, H. K. Air quality prediction using artificial neural networks. In *2019 international Conference on Automation, Computational and Technology Management (icactm)* 568–572 (2019).
24. Qin, Z., Cen, C. & Guo, X. Prediction of air quality based on knn-lstm. *J. Phys: Conf. Ser.* **1237**, 042030 (2019).
25. Ram, K., Singh, S., Sarin, M., Srivastava, A. & Tripathi, S. Variability in aerosol optical properties over an urban site, kanpur, in the indo-gangetic plain: A case study of haze and dust events. *Atmos. Res.* **174**, 52–61 (2016).
26. Schuld, M. & Petruccione, F. *Supervised Learning with Quantum Computers* 17th edn. (Springer, Uk, 2018).
27. Triantafyllou, E. *et al.* Assessment of factors influencing pm mass concentration measured by gravimetric & beta attenuation techniques at a suburban site. *Atmos. Environ.* **131**, 409–417 (2016).
28. Utku, A. & Can, U. Machine learning-based a comparative analysis for air quality prediction. In *2022 30th Signal Processing and Communications Applications Conference (siu)* 1–4 (2022).
29. Wang, J., Li, H., Yang, H. & Wang, Y. Intelligent multivariable air-quality forecasting system based on feature selection and modified evolving interval type- 2 quantum fuzzy neural network. *Environ. Pollut.* **274**, 116429 (2021).
30. Wang, Z. *et al.* Comparison of real-time instruments and gravimetric method when measuring particulate matter in a residential building. *J. Air Waste Manage. Assoc.* **66**(11), 1109–1120 (2016).
31. WAQ, W. I. World's air pollution: Real-time air quality index. https://waqi.info/ (2020).
32. Watson, J. G., Tropp, R. J., Kohl, S. D., Wang, X. & Chow, J. C. Filter processing and gravimetric analysis for suspended particulate matter samples. *Aerosol Sci. Eng.* **1**, 93–105 (2017).
33. Zheng, K., Van Griensven, J. & Fraser, R. A quantum machine learning approach to spatiotemporal emission modelling. *Atmosphere* **14**(6), 944 (2023).
34. Zhu, J. *et al.* Risk assessment of deep coal and gas outbursts based on iqpso-svm. *Int. J. Environ. Res. Public Health* **19**(19), 12869 (2022).
35. Zimmerman, N. *et al.* A machine learning calibration model using random forests to improve sensor performance for lower-cost air quality monitoring. *Atmos. Meas. Tech.* **11**(1), 291–313 (2018).

Author contributions

OF conceived the idea, performed data analysis and wrote the original draft. MS conceived the idea, performed data curation and wrote the original draft. SA performed data curation, formal analysis, and designed methodology. AA dealt with software, performed visualization and carried out project administration. FI performed visualization, deal with software and supervised the work. YAMV acquired the funding for research, and performed visualization and initial investigation. MALF performed initial investigation, provided resources and performed validation. IA supervised the study, performed validation and review and edit the manuscript. All authors read and approved the final manuscript.

Funding

This research is funded by the European University of Atlantic.

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to A.A., F.I. or I.A.