IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed journal & Refereed journal ⁜ Vol. 12, Issue 11, November 2023

DOI: 10.17148/IJARCCE.2023.121112

ILF: A Quantum Semi-Supervised Learning Approach for Binary Classification

Reshma Ahmed Swarna[1], Mohammed Ibrahim Hussain[2], Md. Sadiq Iqbal[3],

Mohammad Mamun[4], Safiul Haque Chowdhury[5]

Lecturer, Department of Computer Science and Engineering, Bangladesh University, Dhaka, Bangladesh[1]

Assistant Professor, Department of Computer Science and Engineering, Bangladesh University, Dhaka, Bangladesh[2]

Associate Professor, Department of Computer Science and Engineering, Bangladesh University, Dhaka, Bangladesh[3]

Student, Department of Computer Science and Engineering, Bangladesh University, Dhaka, Bangladesh[4]

Student, Department of Computer Science and Engineering, Bangladesh University, Dhaka, Bangladesh[5]

Abstract: The lack of enough labeled data is a great issue when designing a real-life scheme. Data labeling is time-consuming as well as costly. Semi-supervised learning (SSL) is a way to solve the issues of data labeling. SSL uses a tiny quantity of labeled data to find labels of massive quantities of unlabeled data. This paper presents a quantum-classical SSL mechanism named "Iterative Labels Finding (ILF)" by combining the Quantum Support Vector Machine algorithm (QSVM) and Ising Models Based Binary Clustering algorithm. The proposed method performs a matching and iteration process to discover the labels of unlabeled data. ILF is designed for binary classification purposes. We have illustrated the experimental result of ILF with a real-time dataset and with a practical example. From experimental results, we have found ILF as a highly efficient approach for quantum SSL.

Keywords: Quantum Computing, QSVM, Semi-Supervised Learning, QML

I. INTRODUCTION

Quantum computing (QC) works based on quantum mechanics. Quantum bit (Qubit) is the basic unit of the QC. A qubit can be 0 or 1 at a time thus it is different from a classical bit. Qubit state is expressed as $a|0\rangle + b|1\rangle$ here a and b are called amplitude and they always hold complex values. With Q quantum bit QC may execute 2^Q states, thus it outperforms the classical computer [1]. Although it has a far way to execute QC absolutely, in recent years a large number of research has been going to improve QC. Hence quantum machine learning achieved an attractable attention in the branch of quantum research [2, 3]. Semi-supervised learning is a classification approach, that uses a small quantity of labeled data with a massive amount of unlabeled data. Semi-supervised learning (SSL) is considered as supervised learning as well as unsupervised learning [4]. In this paper, we have designed an algorithm for SSL by Quantum-Classical machine learning techniques (QCML). QCML allows to simulate the quantum algorithms through classical computers.

Numerous algorithms have been developed and experimented to justify the ability of quantum machine learning. A semi-supervised learning technique for quantum systems has been developed by the authors of the paper [5]. The system is a graph-based quantum annealing technique. Support Vector Machine (SVM) is a well-known excellent algorithm for supervised learning. In paper [6] the authors have proposed a quantum version of classical SVM for binary classification. In the paper [7] the authors have developed a Kernel-based Semi-Supervised learning approach.

The scheme is designed by extending the quantum LS-SVM algorithm. K-means clustering and principal component analysis (PCA) are two major classical unsupervised learning algorithms. In the paper [8] the authors have introduced a quantum version of the classical k-means algorithm. The researchers named the scheme as q-means. In the paper [9] the authors have introduced the quantum version of principal component analysis which is named as QPCA (quantum principal component analysis).

This paper presents an SSL model for QC. We have named this model as "Iterative label finding" or ILF. ILF is a hybrid method and the formation of ILF includes a supervised learning algorithm and an unsupervised learning algorithm. ILF is designed for binary classification. For supervised learning, ILF uses a quantum SVM algorithm. Adiabatic QC-based using models are used as unsupervised learning algorithms in ILF. ILF performs an iterative process to capture the labels of large unlabeled data by using a small portion of labeled data and thus ILF does accurate classification.

The fundamental contribution of ILF includes:

- ILF can able to get the unknown labels of data more accurately.
- ILF provides a unique and new way for quantum semi-supervised learning.
- ILF builds an intersection between supervised learning and unsupervised learning.

II. METHODOLOGY

This section presents the working mechanism of ILF with necessary concepts and the problem statement.

A. Semi-Supervised Learning (SSL)

Assume D_l , D_u are the set of labeled and unlabeled data respectively. Now Consider any dataset D such that $D = D_l \cup D_u = \{(x_1, y_1), (x_2, y_1), \ldots, (x_n, y_n)\} \cup \{ x_1, x_2, x_3, \ldots, x_m \}$; where x and y presents the data and label respectively. The SSL is a machine learning procedure that works on the dataset D in such a way that it tries to discover the labels of D_u with the help of D_l [10]. Working with unlabeled data is precarious [11, 12]. SSL tries to solve this issue since it combines Labeled data with unlabeled data.

B. Quantum-Classical Machine Learning (QCML)

This method is designed with QCML. The basic steps that a QCML algorithm follows are given in Figure 1 [13, 14, 15, 16]. In QCML the first layer holds the real-life data also called classical data. Classical data can be of different types (integer, floating point, etc). Quantum embedding converts classical data into quantum states so that quantum algorithms can execute them. The second layer of QCML performs embedding. Embedding techniques may be of different types e.g., Amplitude Embedding, Basis Embedding, etc. Consider any classical data x, and x is N-dimensional. If the j-th element of x is x_j then the amplitude-embedding convert $\boldsymbol{x_j}$ into quantum state $|\boldsymbol{\Psi_x}\rangle$ in such ways that

$$|\Psi_x\rangle = \sum_{j=1}^{N} x_j \mid j\rangle,\ N = 2^n \quad \text{(i)}$$

The third and most important step of QCML is the quantum algorithm, this is the core process that performs all the calculations by using the quantum mechanical concept. Measurement converts the quantum state into the classical bit. To get the result from a quantum system measurement is required. The last step of QCML executes measurement operation.

C. Quantum Support Vector Machine (QSVM)

One of the most efficient supervised learning algorithms is SVM [17, 18, 19]. The objective of SVM is to get a decision line that differentiates n-dimensional space into exact classes. The decision line of SVM is known as a hyperplane. The nearest datapoints of the hyperplane is known as support vectors. SVM always tries to maintain a large margin hyperplane for better classification. Figure 2 shows the classification of two different types of data points by hyperplane. In this paper, we have used the quantum version of SVM (known as QSVM) [20] to develop our scheme. QSVM follows the same process as QCML. Quantum feature mapping is performed by a circuit $V(\Psi_x)$ to interpret the classical data x into quantum state $|\Psi_x\rangle$. After that, a kernel is formed by taking the inner product of quantum feature maps $K(x, z) = |\langle \Psi_x | \Psi_z\rangle|^2$. By computing the Kernel matrix in the quantum computer, they can train the QSVM in a similar way to classical SVM. The final output is obtained by applying measurement. Measurement returns value $y \mid y \in \{-1, 1\}$.

D. Ising Models Based Binary Clustering

Clustering is a data grouping task that ensures the same type of data may exist in the same group. Figure 3 shows the clustering concept. The ising model is a statistical mechanics-based mathematical model for physical ferromagnetic substances. For a set of N spins using the model is expressed with the Hamiltonian (H) [21]

$$H = -J \sum_{i,j} s_i s_j - B \sum_{i=1}^{N} s_i \quad \text{(ii)}$$

s_i, J and B are known as magnetic spins, coupling constant, and applied magnetic field respectively in equation (2). Quantum computing that works based on the adiabatic theorem is called Adiabatic Quantum Computing [22]. In this paper, we have used an ising model-based binary clustering for adiabatic quantum computing, inspired from [23].

Figure 3 shows the binary clustering of data using the Ising model. We have evaluated our method using the ising model for two types of distance. Firstly we have used Euclidean distance (d_E) and secondly, we have used Mahalanobis distance(d_m). d_E and d_m for any vectors x and y with covariance matrix S, is defined as,

$$d_E\,(\boldsymbol{x},\boldsymbol{y}) = \sqrt{(\boldsymbol{x}-\boldsymbol{y})^T(\boldsymbol{x}-\boldsymbol{y})} \qquad (iii)$$

$$d_m\,(\boldsymbol{x},\boldsymbol{y}) = \sqrt{(\boldsymbol{x}-\boldsymbol{y})^T s^{-1}(\boldsymbol{x}-\boldsymbol{y})} \qquad (iv)$$

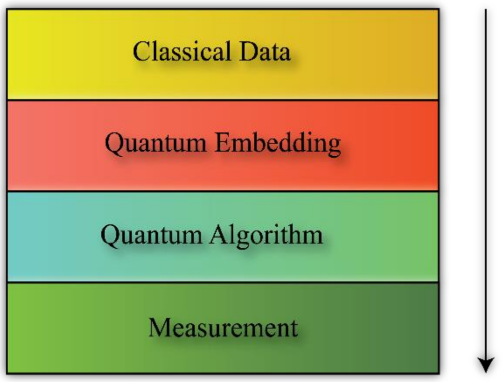


Fig. 1. The fundamental steps of QCML.

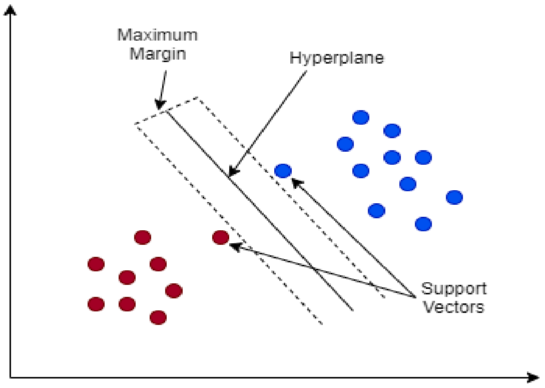
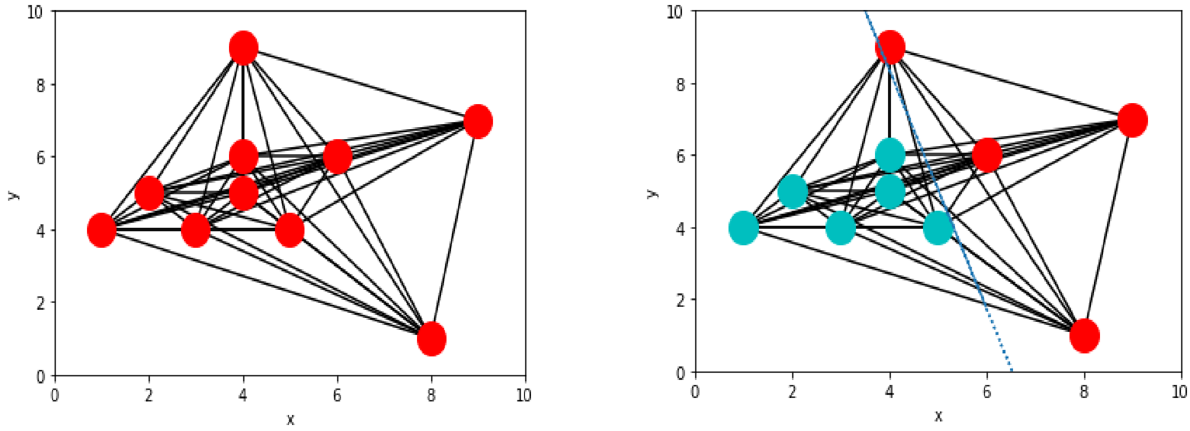


Fig. 2. Hyperplane Concept in SVM



(a) Before clustering.

(b) After clustering.

Fig. 3. Binary clustering using the Ising model.

E. Problem Statement

The core objective of our proposed scheme is to find the labels $y \mid y \in \{-1, 1\}$ and y ensures the labels of D_u . To fulfill this objective, we have tested QSVM (QSVM is trained with D_l before) and clustering (using the ising model) through D_u and matched all the labels that QSVM and ising model produced. If a label is matched, it is considered as a new labeled data for D_l. Secondly, we have applied the same matching process until we have found all the labels of D_u. For this iterative process, we have named the system as ILF. During the iterative process if no matching is found on any iteration we have applied data separation or data adding process to meet the target.

F. Iterative Label Finding(ILF)

This section presents the principal process of our proposed method. Figure 4 shows the core block diagram of our scheme and Step 1-5 explains the process thoroughly.

1. *Execute QSVM:* Train the QSVM algorithm using D_l. Test trained QSVM using D_u. Let y_a be the output of QSVM after testing. Here $y_a \in \{-1, 1\}$ and y_a indicate the labels of data.
2. *Execute clustering model:* Apply ising model-based clustering on D_u. Let y_b be the output of the model. Here $y_b \in \{-1, 1\}$ and y_b indicates the labels of data.
3. *Perform Matching:* Compare each $(y_a)_i$ with each $(y_b)_i$. If $(y_a)_i = (y_b)_i$ occurs then subtract x_i from D_u and update D_l with x_i. After updating *D*, go to step 1 and continue again. Here *i* indicate any arbitrary entity. If there exists no *i* for which $(y_a)_i = (y_b)_i$ occurs then follow step 04.
4. *Perform Data Separation:* Subtract 2^P data from D_u. Here $P \geq 0$ and $P < (n + m)$. Add subtracted data to the dataset D_e, and update D_u. After updating D_u, go to step 1 and continue again. If it is not possible to subtract 2^P data from D_u

DOI: 10.17148/IJARCCE.2023.121112

then check whether all labels are discovered or not. If all labels are generated then go to step 5. If all labels are not generated then follow step 4.

5. *Perform Data Adding:* Add 2^P data to D_u from D_e. If D_e is empty add 2^P data to D_e from D_l randomly(While adding data randomly from D_l to D_e ignore the labeled data that has just been added at the last step.). After updating D_u, go to step 1 and continue again.
6. *Output:* Stop the process and show the result.

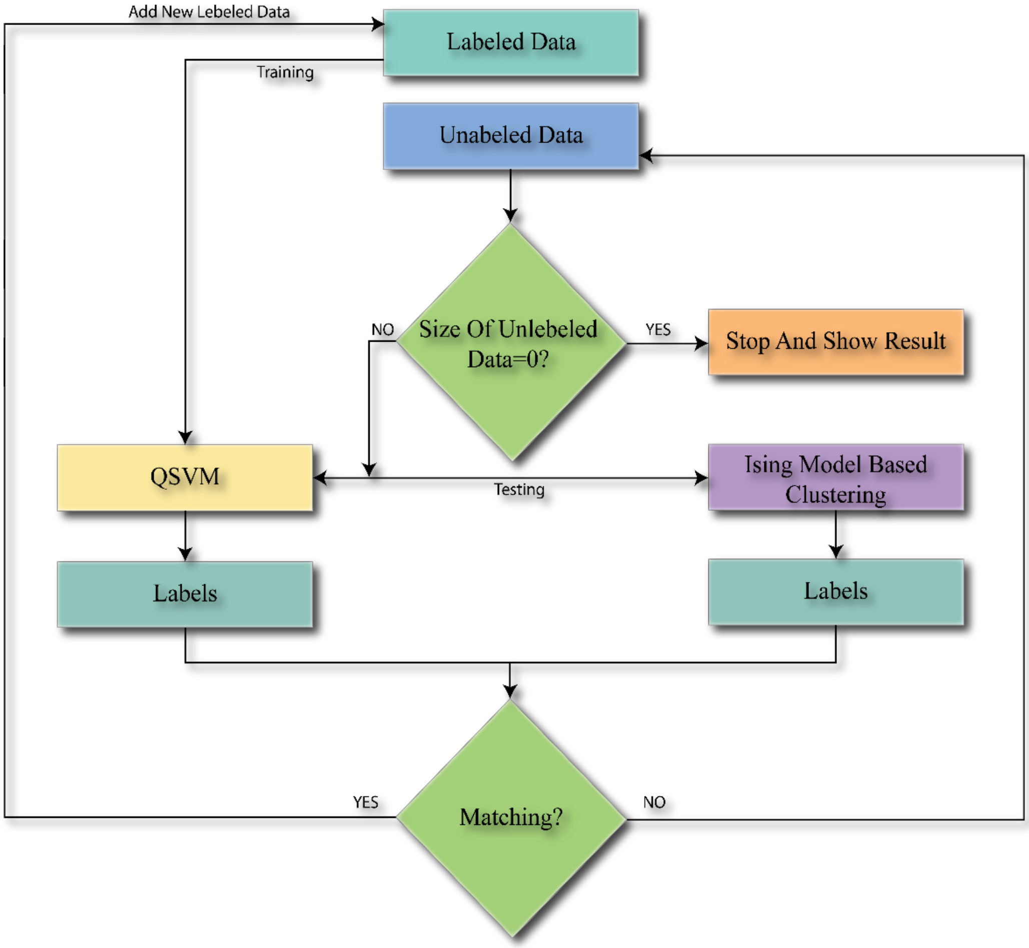


Figure 4. Block diagram of the proposed method.

During the execution of the Ising model, we may consider either Euclidean distance or Mahalanobis distance but it has to be remembered that we may not execute Euclidean and Mahalanobis distance similarly for a dataset. So before applying ILF to any dataset, we have to decide whether we need to use Euclidean distance or Mahalanobis distance. It is also to be remembered that when updating the table D_l we have to discard the duplicate entry. The Entire Algorithm of ILR is given in Algorithm.1

Algorithm 1. Algorithm of ILF

```
$D_l \longleftarrow$ Labeled data
$D_u \longleftarrow$ Unlabeled data
Input: Dataset $D = D_l \cup D_u$
Output: Labels of $D_u$

$A \longleftarrow$ QSVM algorithm
$B \longleftarrow$ Ising model-based binary clustering algorithm
$y_a \longleftarrow$ Labels of $D_u$ generated by executing A
$y_b \longleftarrow$ Labels of $D_u$ generated by executing B
1. Loop: Until all labels of $D_u$ is discovered
```

```
Execute $A(D_l) \mid A(D_u) = y_a$
Execute $B \mid B(D_u) = y_b$
Loop: For every $i$
    If $(y_a)_i = (y_b)_i$ then
        Update $D_l$ with $[x_i, (y_a)_i] \mid D_u = D_u - [x_i, (y_a)_i]$
    Else
        Perform data separation or data adding and update $D_l$, $D_u$
    End if
End Loop
End Loop
```

III. RESULTS AND DISCUSSION

This section presents the result of our method along with a related discussion. Firstly, this section will analyze a practical example of ILF and secondly, the section will provide the outcome analysis and discussion of our experiment.

A. *Practical Example*

To demonstrate the example, consider the dataset of Table 1.

Table 1. Dataset to illustrate the practical Example of ILF.

	x	**y**	
	1.20	1	D_l
	4	-1	
D_u	2.45		
	1.95		
	7		
	3		

Let $D_l(x)$, $D_l(y)$ present the data and labels of D_l and $D_u(x)$, $D_u(y)$ presents the data and labels of D_u. So initially,

$D_l(x) = \{1.20, 4\}$
$D_l(y) = \{1, -1\}$
$D_u(x) = \{2.45, 1.95, 7, 3\}$
$D_u(y) = \{ y_1, y_2, y_3, y_4\}$

Our objective is to find the values of y_1, y_2, and y_4 correctly. For simplicity assume that *A* is the QSVM algorithm and *B* is the Ising model-based binary clustering algorithm. Again assume A(X, Y), A(X) is the training and testing function of A with data X and labels Y. Similarly, B(X) is the testing function of B. Now follow the given step to execute ILF in Table 1. (Note: We have assumed the result of A(X) and B(X) in the following steps to demonstrate the practical example)

Step 1: Perform $A(D_l(x), D_l(y)) \mid A(D_u(x)) = \{ y_1(A), y_2(A), y_3(A), y_4(A) \} = \{-1, 1, -1, 1\}$
Step 2: Perform $B(D_u(x)) \mid B(D_u(x)) = \{ y_1(B), y_2(B), y_3(B), y_4(B) \} = \{1, 1, -1, -1\}$
Step 3: Perform matching $y_i(A)$ with $y_i(B)$. We have found two matching. y_2 and y_3. It means,

$$y_2(A) = y_2(B) = 1 \text{ and } y_3(A) = y_3(B) = -1$$

Now update $D_l(x)$, $D_l(y)$, $D_u(x)$ and $D_u(y)$,
$D_l(x) = \{1.20, 1.95, 4, 7\}$
$D_l(y) = \{1, 1, -1, -1\}$
$D_u(x) = \{2.45, 3\}$
$D_u(y) = \{ y_1, y_4\}$

Step 4: Perform $A(D_l(x), D_l(y)) \mid A(D_u(x)) = \{ y_1(A), y_4(A) \} = \{-1, 1\}$
Step 5: Perform $B(D_u(x)) \mid B(D_u(x)) = \{ y_1(B), y_4(B) \} = \{1, -1\}$
Step 6: Perform matching $y_i(A)$ with $y_i(B)$. No matching found so Subtract 2^P (with P = 0) data from $D_u(x)$ and add to D_e. Now update $D_l(x)$, $D_l(y)$, $D_u(x)$, and $D_u(y)$ and continue from step 1.
$D_l(x) = \{1.20, 1.95, 7, 4\}$
$D_l(y) = \{1, 1, -1, -1\}$
$D_u(x) = \{2.45\}$
$D_e = \{3\}$

$D_u(y) = \{ y_1\}$

Step 7: Perform $A(D_l(x), D_l(y)) \mid A(D_u(x)) = \{ y_1(A) \} = \{1\}$

Step 8: Perform $B(D_u(x)) \mid B(D_u(x)) = \{y_1(B)\} = \{1\}$

Step 9: Perform matching $y_i(A)$ with $y_i(B)$. One matching found,

$$y_1(A) = y_1(B) = 1$$

Now update $D_l(x)$, $D_l(y)$, $D_u(x)$ and $D_u(y)$,

$D_l(x) = \{1.20, 1.95, 2.45, 7, 4\}$

$D_l(y) = \{1, 1, 1, -1, -1\}$

$D_u(x) = \{3\}$

$D_e = \{\}$

$D_u(y) = \{ y_4\}$

Step 7: Perform $A(D_l(x), D_l(y)) \mid A(D_u(x)) = \{ y_4(A) \} = \{-1\}$

Step 8: Perform $B(D_u(x)) \mid B(D_u(x)) = \{y_4(B)\} = \{1\}$

Step 9: Perform matching $y_i(A)$ with $y_i(B)$. No matching found and subtraction of 2^P (with P = 0) data is not possible. So perform data adding to D_e form $D_l(x)$. We may add any data without 2.45 since it is added at the last iteration. So,

$D_e = \{1.20\}$

Now update $D_l(x)$, $D_l(y)$, $D_u(x)$, $D_u(y)$, and D_e and continue from step 1.

$D_l(x) = \{1.20, 1.95, 7, 4\}$

$D_l(y) = \{1, 1, -1, -1\}$

$D_u(x) = \{3, 1.20\}$

$D_e = \{\}$

$D_u(y) = \{ y_{4,} y_r\}$

Step 7: Perform $A(D_l(x), D_l(y)) \mid A(D_u(x)) = \{ y_4(A), y_r(A) \} = \{1, -1\}$

Step 8: Perform $B(D_u(x)) \mid B(D_u(x)) = \{ y_4(B), y_r(B) \} = \{1, -1\}$

Step 9: Perform matching $y_i(A)$ with $y_i(B)$. Two matchings are found,

$$y_4(A) = y_4(B) = 1 \text{ and } y_r(A) = y_r(B) = -1$$

Now update $D_l(x)$, $D_l(y)$, $D_u(x)$, $D_u(y)$, and D_e (To ignore duplicate value y_r is not considered)

$D_l(x) = \{1.20, 1.95, 2.45, 4, 7, 3\}$

$D_l(y) = \{1, 1, 1, -1, -1, -1\}$

$D_u(x) = \{\}$

$D_e = \{\}$

$D_u(y) = \{ \}$

The final output is shown in Table 2.

Table 2. Result of ILF algorithm for table 1.

x	y
1.20	1
1.95	1
2.45	1
4	-1
7	-1
3	-1

B. Outcome Analysis and Discussion

To develop ILF, we have used code from [24] for QSVM and ising-based clustering model. To evaluate the proposed method, we have used the iris [25] dataset. The Iris dataset has 3 labels and each label has 50 samples so there are a total of 150 rows and 4 columns in the dataset, but the proposed method is designed for binary classification hence we have considered only Iris-setosa and Iris-versicolor classes. We have used 20% data as a labeled data and 80% data as unlabeled data to execute the proposed method. To experiment with the system using Iris-setosa and Iris-versicolor classes firstly we have applied QSVM and Euclidean distance-based ising clustering model the summary of results for these cases is given in Table 3, secondly, we have applied QSVM and Mahalanobis distance-based ising clustering model the summary of results for these cases is given on table 4. In all the cases we have obtained 100% accuracy.

DOI: 10.17148/IJARCCE.2023.121112

Table 3. Summary of experimental results for ILF (QSVM and Euclidean distance based ising clustering model)

iteration	Number of Matching	QSVM Accuracy	Clustering Accuracy (Euclidean distance)
1	87	94%	87%
2	4	30.7%	53.84%
3	0	22.22%	44.44%
4	7	77.78%	100%
5	1	50%	50%
6	0	0%	0%
7	1	100%	100%

Table 4. Summary of experimental results for ILF (QSVM and Mahalanobis distance-based using clustering model)

iteration	Number of Matching	QSVM Accuracy	Clustering Accuracy (Mahalanobis distance)
1	94	94%	98%
2	4	66.67%	100%
3	0	0%	50%
4	2	100%	100%

Figure 5 compares the performance of the proposed method for the used dataset based on Mahalanobis and Euclidean distance. In Figure 5 x-axis presents the number of iterations and the y-axis presents the number of matching data in each iteration.

From Figure 5, it is also clear that the Mahalanobis distance-based model performed better than Euclidean distance.

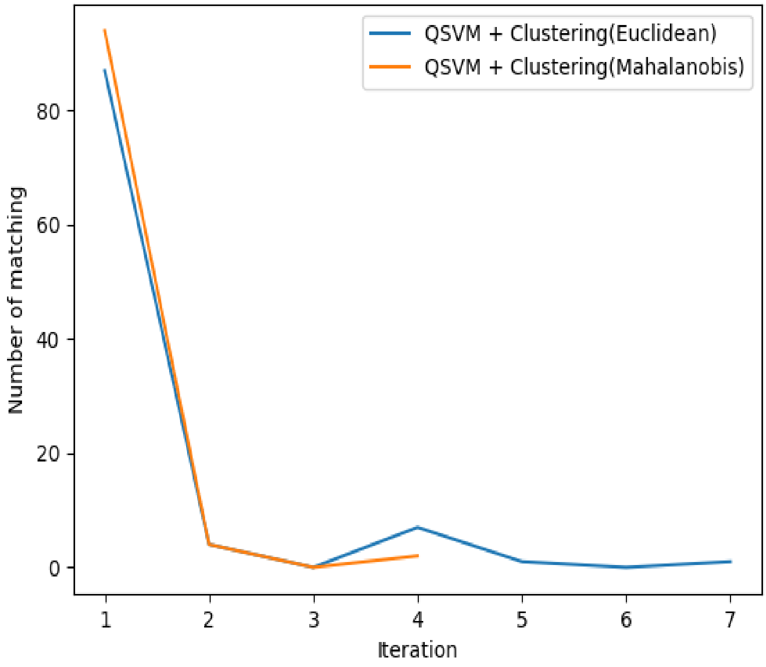


Fig. 5. Comparison of performance of ILF based on Mahalanobis and Euclidean distance

DOI: 10.17148/IJARCCE.2023.121112

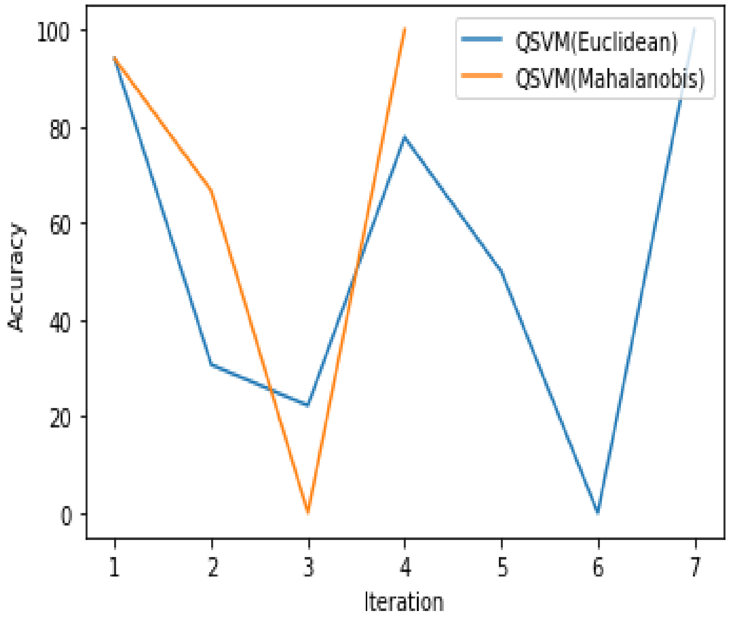


Fig. 6. Comparison of performance of QSVM (when QSVM is used with Mahalanobis or Euclidean distance.)

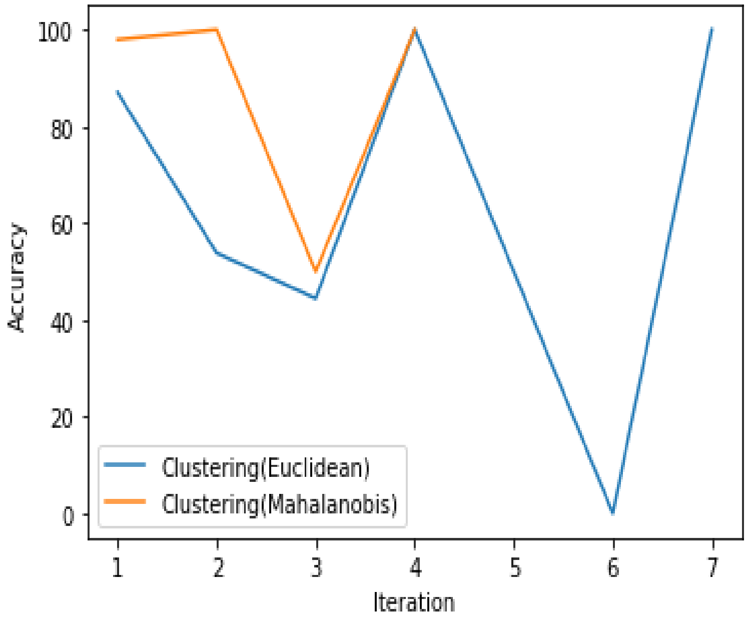


Fig. 7. Comparison of performance of ising-based clustering model based on Mahalanobis and Euclidean distance.

The complexity of ILF is O(i(p+q)), here i is the number of iterations, p is the complexity of QSVM and q is the complexity of a based clustering model. One of the major problems with ILF is that if the QSVM and ising-based clustering model make a false prediction for the same data on the same iteration then the system observes that sample correctly. However, IF executes two algorithms and performs matching hence it is much more efficient than other systems.

IV. CONCLUSION

The concept of quantum computing (QC) is derived from the core idea of quantum physics. QC is considered to be the next-generation computer hence nowadays day QC is a trending topic of research in the branch of computing and communication. Machine learning (ML) is the most important branch of artificial intelligence. To cope with upcoming technology, researchers try to apply ML with quantum computing, thus the concept of quantum machine learning has

DOI: 10.17148/IJARCCE.2023.121112

been produced. Classification is an important concept in the ML branch. Several classification processes for quantum machine learning (QML) have been developed by researchers. This research introduces a technique for classification, we have named our scheme as ILF. Semi-supervised learning is a classification method. Semi-supervised learning works with little labeled data and tries to find the labels of large unlabeled data. Mainly ILF is a semi-supervised learning approach. ILF consists of two existing QML algorithms:- quantum support vector machine (QSVM) and Ising models based on binary clustering. ILF performs its process iteratively hence it is a time-consuming method but as it combines two methods to capture the output it is highly accurate. We have developed the system using the quantum-classical machine learning method.

REFERENCES

[1]. Zhang, Y., & Ni, Q. (2020). Recent advances in quantum machine learning. Quantum Engineering, 2(1), e34.
[2]. Dunjko, V., & Briegel, H. J. (2017). Machine learning\& artificial intelligence in the quantum domain. arXiv preprint arXiv:1709.02779.
[3]. Ciliberto, C., Herbster, M., Ialongo, A. D., Pontil, M., Rocchetto, A., Severini, S., & Wossnig, L. (2018). Quantum machine learning: a classical perspective. Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences, 474(2209), 20170551.
[4]. Zhu, X. J. (2005). Semi-supervised learning literature survey.
[5]. Zheng, Y. L., Zhang, W., Zhou, C., & Geng, W. (2021). Quantum annealing for semi-supervised learning. Chinese Physics B, 30(4), 040306.
[6]. Rebentrost, P., Mohseni, M., & Lloyd, S. (2014). Quantum support vector machine for big data classification. Physical review letters, 113(13), 130503.
[7]. Saeedi, S., Panahi, A., & Arodz, T. (2021). Quantum semi-supervised kernel learning. Quantum Machine Intelligence, 3, 1-11.
[8]. Kerenidis, I., Landman, J., Luongo, A., & Prakash, A. (2019). q-means: A quantum algorithm for unsupervised machine learning. In Advances in Neural Information Processing Systems (pp. 4134-4144).
[9]. Lloyd, S., Mohseni, M., & Rebentrost, P. (2014). Quantum principal component analysis. Nature Physics, 10(9), 631-633.
[10]. Learning, S. S. (2006). Semi-Supervised Learning. CSZ2006. html.
[11]. Li, Y. F., & Liang, D. M. (2019). Safe semi-supervised learning: a brief introduction. Frontiers of Computer Science, 13, 669-676.
[12]. Singh, A., Nowak, R., & Zhu, J. (2008). Unlabeled data: Now it helps, now it doesn't. Advances in neural information processing systems, 21.
[13]. Zhang, Y., & Ni, Q. (2020). Recent advances in quantum machine learning. Quantum Engineering, 2(1), e34.
[14]. Biamonte, J., Wittek, P., Pancotti, N., Rebentrost, P., Wiebe, N., & Lloyd, S. (2017). Quantum machine learning. Nature, 549(7671), 195-202.
[15]. Kopczyk, D. (2018). Quantum machine learning for data scientists. arXiv preprint arXiv:1804.10068.
[16]. Carrasquilla, J. (2020). Machine learning for quantum matter. Advances in Physics: X, 5(1), 1797528.
[17]. Pisner, D. A., & Schnyer, D. M. (2020). Support vector machine. In Machine learning (pp. 101-121). Academic Press.
[18]. Awad, M., Khanna, R., Awad, M., & Khanna, R. (2015). Support vector machines for classification. Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers, 39-66.
[19]. Braun, A. C., Weidner, U., & Hinz, S. (2011, June). Support vector machines, import vector machines and relevance vector machines for hyperspectral classification—A comparison. In 2011 3rd Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS) (pp. 1-4). IEEE.
[20]. Havlíček, V., Córcoles, A. D., Temme, K., Harrow, A. W., Kandala, A., Chow, J. M., & Gambetta, J. M. (2019). Supervised learning with quantum-enhanced feature spaces. Nature, 567(7747), 209-212.
[21]. The phycomp website. [Online]. Available: http://phycomp.technion.ac.il/~lior/ising3d.html [Last accessed on December 1, 2023]
[22]. Farhi, E., Goldstone, J., Gutmann, S., & Sipser, M. (2000). Quantum computation by adiabatic evolution. arXiv preprint quant-ph/0001106.
[23]. Bauckhage, C., Brito, E., Cvejoski, K., Ojeda, C., Sifa, R., & Wrobel, S. (2017). Adiabatic quantum computing for binary clustering. arXiv preprint arXiv:1706.05528.
[24]. The renom website. [Online]. Available: https://www.renom.jp/packages/renomq/rsts/example/Quantum_machine_learning/quantum_machine_learning.html [Last accessed on December 1, 2023]
[25]. The kaggle website. [Online]. Available: https://www.kaggle.com/uciml/iris [Last accessed on December 1, 2023]