An Optimizing Method for Performance and ResourceUtilization in Quantum Machine Learning Circuits

Tahereh Salehi
Ferdowsi University of Mashhad
Mariam Zomorodi (✉ zomorodi@pk.edu.pl)
Cracow University of Technology
Paweł Pławiak
Cracow University of Technology
Mina Abbaszade
Isfahan University of Technology
Vahid Salari
Basque Center for Applied Mathematics

Research Article

Keywords: Optimizing, Quantum Machine Learning Circuits, algorithms

Posted Date: January 10th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1214658/v1

An Optimizing Method for Performance and Resource Utilization in Quantum Machine Learning Circuits

Tahereh Salehi[1], **Mariam Zomorodi**[1,2,*], **Pawel Plawiak**[2,3], **Mina Abbaszade**[4], **and Vahid Salari**[5,6]

[1]Department of Computer Engineering, Ferdowsi University of Mashhad, Mashhad, Iran
[2]Department of Computer Science, Faculty of Computer Science and Telecommunications, Cracow University of Technology, Krakow, Poland
[3]Theoretical and Applied Informatics, Polish Academy of Sciences, Gliwice, Poland
[4]Department of Physics, Isfahan University of Technology, Isfahan 84156-83111, Iran
[5]Basque Center for Applied Mathematics (BCAM), Bilbao, Spain
[6]Department of Physics and Astronomy, Howard University, Washington, DC 20059, USA
[*]Corresponding author: zomorodi@pk.edu.pl

ABSTRACT

Quantum computing is a new and advanced topic that refers to calculations based on the principles of quantum mechanics. It makes certain kinds of problems be solved easier compared to classical computers. This advantage of quantum computing can be used to implement many existing problems in different fields incredibly effectively. One important field that quantum computing has shown great results in machine learning. Until now, many different quantum algorithms have been presented to perform different machine learning approaches. In some special cases, the execution time of these quantum algorithms will be reduced exponentially compared to the classical ones. But at the same time, with increasing data volume and computation time, taking care of systems to prevent unwanted interactions with the environment can be a daunting task and since these algorithms work on machine learning problems, which usually includes big data, their implementation is very costly in terms of quantum resources. Here, in this paper, we have proposed an approach to reduce the cost of quantum circuits and to optimize quantum machine learning circuits in particular. To reduce the number of resources used, in this paper an approach including different optimization algorithms is considered. Our approach is used to optimize quantum machine learning algorithms for big data. In this case, the optimized circuits run quantum machine learning algorithms in less time than the original ones and by preserving the original functionality. Our approach improves the number of quantum gates by 10.7% and 14.9% in different circuits and the number of time steps is reduced by three and 15 units, respectively. This is the amount of reduction for one iteration of a given sub-circuit U in the main circuit. For cases where this sub-circuit is repeated more times in the main circuit, the optimization rate is increased. Therefore, by applying the proposed method to circuits with big data, both cost and performance are improved.

Introduction

In recent years, the phenomenon of quantum computing has received global attention[1]. Quantum computational theory was started by Feynman and Deutsch in the 1980s[2] and after that many new quantum computing algorithms have been proposed. Machine learning is the science and art of building computers to learn from data how to solve problems instead of explicitly programming. So, Machine learning and quantum computing are two very important research areas, and by combining these two areas, new solutions for today's challenges are proposed[3]. One usual way to meet these challenges is to arrange these algorithms in the cloud system. With the help of the computing power of the cloud system, the problems are partially solved. However, data storage and management in heterogeneous distributed networks had a number of other problems. Physicists take a different method of quantum computing by exploiting the "Superposition" and "Entanglement" properties[8]. This solution has particularly increased the speed of solving certain problems compared to classical algorithms[4,5]. Problem-solving is very different in quantum and classical systems. In fact, some problems that can be solved in the classical system in several years, it is known that can be solved in a quantum system in a few hours[6]. On the other hand, in recent years many types of research have been carried out on the subject of big data. The challenge is the inefficiency of the computations of classical machine learning algorithms for processing such a large volume of data[7,8]. The unit of quantum processing is the "quantum bit" or "qubit". One of the capabilities of a quantum computer is that by increasing the number of qubits of a quantum computer, the processing power improves exponentially[9]. Quantum algorithms usually express computations by primitive quantum gates. There are different approaches to implementing these algorithms. Therefore, it is useful to find an implementation using the least resource numbers, especially for large-scale quantum circuits with complex designs. To this end, we apply optimization methods which is a fundamental task in almost all areas of science[9, 10]. This work has developed and implemented a framework for quantum circuit optimization algorithms to optimize the desired circuits which are designed particularly for machine learning tasks. We also show how to optimize the repetition of quantum circuits and reduce the required resources for large-scale quantum circuits. While the original functionality of the algorithm is preserved, the final quantum circuit has fewer time steps, execution time, and quantum cost compared to the original circuit. As input, we assumed that the quantum circuit (QC) consists of a set of quantum gates with a certain number of 2 qubits . The ultimate goal of optimizing the quantum circuit of a machine learning algorithm is to reduce the number of gates, time steps, and quantum cost. The quantum cost of a circuit is the number of 1×1 and 2×2 quantum gates in its design[11]. For this purpose, this paper proposes a method to optimize the quantum cost of machine learning algorithms. In principle, it can be said that the operations involved in quantum machine learning circuits can be large and so it is worth reducing them. Quantum circuits typically use single-qubit and two-qubit gates such as NOT, Hadamard, and rotation, and also two-qubit CNOT gates. If there are three-qubit gates such as Bridge, and Swap, and multi-qubit gates, we decomposed them into single-qubit and two-qubit gates in a preprocessing step. In Section 2, we discussed related work in the field of the quantum computation systems for the machine learning algorithm, as well as optimization algorithms for the quantum circuits. Then, in section 3, the proposed method is explained and at the end, in sections 4 and 5, our results and discussion are presented and we conclude the paper.

Quantum gates and circuits

Quantum circuit is made up of a combination of Von Neumann and classical architecture[12], which is executed in the quantum processor. Quantum circuits are shown in such a way that the desired gates are on vertical lines, each of which represents a basic operation. Run time is counted in these circuits from left to right[13] with time steps of the quantum circuit. Quantum gates can be represented as 2^n dimensional matrices that contain the amplitude of the fundamental states 2^n of an n-dimensional quantum system. Figure 1 shows some examples of quantum circuits and their corresponding gates in matrix representation. Figure 1 (a) shows a sample quantum circuit, and Figure 1 (b) represents unitary two-qubit gates with their corresponding matrices[14]. Time advances from left to right.

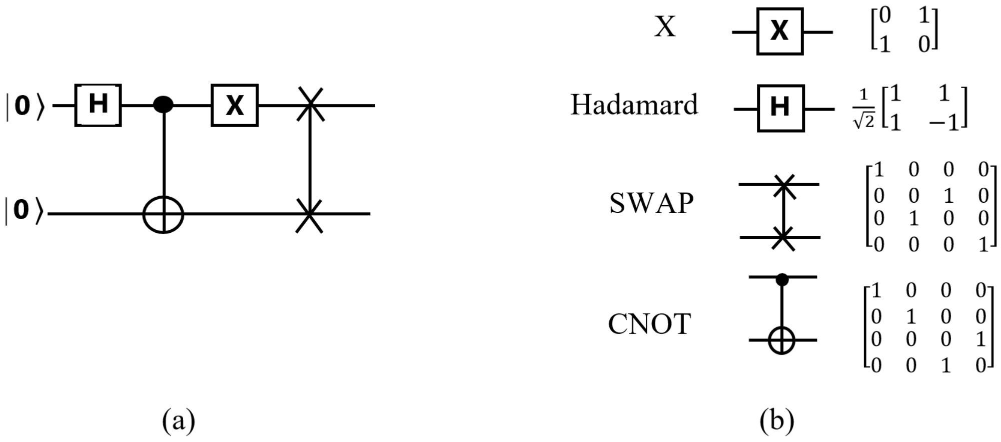

|0⟩
|0⟩
H
X
(a)
X
Hadamard
SWAP
CNOT
X
H
$\begin{bmatrix}0 & 1\\1 & 0\end{bmatrix}$
$\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\1 & -1\end{bmatrix}$
$\begin{bmatrix}1&0&0&0\\0&0&1&0\\0&1&0&0\\0&0&0&1\end{bmatrix}$
$\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&0&1\\0&0&1&0\end{bmatrix}$
(b)

Figure 1. a) An example for a quantum circuit: each line represents a qubit, and operations (gates) are applied on them. b) Representation of unitary two-qubit gates in the quantum circuit with matrix formalism[14]

Quantum Memory

The memory of a classic computer can be easily built by writing an arbitrary bit string in any position. Classical memory performance is not optimal for processing big data. In order to solve the problem of normal and associative memory capacity, quantum memory has been used successfully. In many applications of quantum computers, a quantum register is used instead of classical memory to simulate a physical system. This quantum memory consists of a qubit state tensor in multidimensional Hilbert space that is first prepared in a simple state. For a quantum memory consisting of two qubits $|q_0\rangle$ and $|q_1\rangle$, its state $|q_R\rangle$ is given by Equation (1). The symbol $\otimes$ is the tensor operation. For example, for a 4-qubit quantum register, its state is represented as Equation (2), where the probability of measuring each of its base states is as different four terms illustrated in Equation (3)[15,16]. Giovannetti et al.[17] have demonstrated that how classical data is presented in the language of quantum mechanics through the study of quantum random access memory (QRAM).

$$|q_R\rangle = |q_0\rangle \otimes |q_1\rangle = |q_0\rangle|q_1\rangle = |q_0 q_1\rangle \tag{1}$$

$$|q_R\rangle = c_0|00\rangle + c_1|01\rangle + c_2|10\rangle + c_3|11\rangle \tag{2}$$

$$|c_0|^2 + |c_1|^2 + |c_2|^2 + |c_3|^2 = 1 \tag{3}$$

Implementing machine learning algorithms in quantum computing has two advantages: storage scale and high execution speed of algorithm[8]. By exploiting the superposition property, it is possible that quantum storage be reduced exponentially. According to equations (4) and (5), all binary numbers from the set $\{0, 1, \cdots, i, \cdots, 2^n - 1\}$ are placed in n qubit quantum memory with probability $|c_i|^2$ for state $|i\rangle$.

$$|\phi_1 \phi_2 \cdots \phi_n\rangle = \sum_{i=0}^{2^n-1} c_i |i\rangle \tag{4}$$

where

$$\sum_{i=0}^{2^n-1} |c_i|^2 = 1 \tag{5}$$

In the classical system for equation (6), the operation must be repeated 2^n times. But in the quantum system, the system can examine all computational states for the variables simultaneously, assuming that the U_f operator understands the function $f(x)$[8].

$$U_f\left(\frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle\right) = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} U_f|x\rangle) = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |f(x)\rangle \tag{6}$$

Related Work

In this section, we first present some works done in implementing a quantum circuit for quantum machine learning algorithms and then methods for quantum circuit optimization are presented.

Quantum circuits of machine learning algorithms

Recently, quantum machine learning is considered as a suitable solution to increase the speed of execution of algorithms. This method has led to the introduction of various quantum algorithms for machine learning using quantum features. In this paper, we first examine the K nearest neighbor algorithm. In this regard, Lioyd et al.[18] and Wiebe et al.[19] use similar approaches such as quantum amplitude estimation or Grover algorithm[20] to obtain the quantum state of the nearest neighbor algorithm. In the next method for implementing the nearest neighbor algorithm, Buhrman et al.[21] use quantum parallelism and the test circuit to calculate the distance between two vectors and provide a quantum solution. The Euclidean distance can be calculated as Euclidean distance= $\sqrt{((2-2|\langle x|y\rangle|))}$. The next method in K nearest neighbor by Ruan et al.[22] is used in document classification, image classification, etc. It works based on the size of the Hamming distance. A natural vector is defined as a bit vector with a hash function and then converted to an equivalent quantum state, after that, then the input vector bits are compared with the training vector. The number of different properties is counted by the Kaye circuit[22] and the distance between the two vectors is estimated. The next algorithm, the support vector machine (SVM), is a supervised algorithm developed by Arodz and Saeedi[23] and also by Rebentrost et al.[24], which classifies vectors in a specific space based on training data. In comparison with the classical support vector machine for binary classification, they achieved a logarithmic acceleration. These methods use Grover algorithm and adiabatic algorithm. The next algorithm is the neural network algorithm. Transfer learning is an interesting technique in neural networks in which a pre-trained model is reused as an input model for a new task. One of the works in quantum neural networks presented in this field was developed by Acar et al[25] and uses the quantum transfer learning method. This method is a hybrid machine learning method consisting of a classical network feature extractors and a diverse quantum classification circuit. There are other works by Zen et al.[26] that have used transmission learning toward scalable quantum neural network states using transmission learning. A protocol was proposed in[44] for machine translation based on quantum long short term memory for translating a sentence from English to Persian. In another work, Mishra et al.[27] used the design and operation of a classical neural network and they designed a quantum neural network capable of working on a 10 qubit system. By demonstrating network performance, they have tried to use the basic principles of machine learning to manage data that can be used in cancer detection.

QUANTUM CIRCUITS OPTIMIZATION TECHNIQUES

There are many different optimization methods for quantum circuits that aim to reduce the number of time steps, gates, depth, and more. For this purpose, quantum circuits can be improved using different approaches in circuit diagram models, or using circuit simplification rules. For example, the quantum circuit can be improved by the quantum Karnaugh map method[28], the exclusive sum of product method[29] and etc. Using circuit simplification rules, there are many different methods that improve circuits by specific rules. The following is a review of the works in this area. Fan et al.[30] proposed a quantum approximate optimization algorithm, which is a standard method for combinatorial optimization with a gate-based quantum computer. The paper introduced a new Gibbs objective function and demonstrated its superiority, and used the architecture of an Ansatz search algorithm to search for the discrete space of a quantum circuit. These changes led to the improvement of various circuits. Using this method, the median has been increased to 244.7% and 44.4% for the grid and complete graph models of quantum computation. Median reduction in the number of two-qubit gates is 33.3% and 20.8%, respectively. In another paper, Alam et al.[31] proposed a method to accelerate the implementation of the quantum approximation optimization algorithm (QAOA). First, a connection is made between the classical optimizer and the quantum computer, and then two parameters named δ and β, with initial values of zero, are inserted into the loop. The classical optimizer for randomly defined variables initially set to some random values. If the values are not ideal, it establishes a connection to the quantum computer. increases the depth of the circuit, which is not good and should be reduced. For this reason, to determine the appropriate distance for the parameters, artificial intelligence techniques are used to achieve the desired result with the acceleration in the process. This method shows that the number of optimization iterations can be reduced 44.9% on average for 264 graphs. Haner et al.[12] optimized the circuit using the Hoare triples[32]. This method checks the accuracy of the execution of specific programs. For each circuit level, a pre-condition defines conditions and post-conditions, and based on the previous level condition, the authors can decide on the operating conditions for the next level operation. When using a Hoare-based optimization strategy, the circuit depth decreases for $n \geq 2$, according to relation $(4(n-2)+n)/n$. In the next method of Childs and Maslov[13], the automatic optimization of large circuits is accomplished using iterative parameters. This method also preserves the main structure of the algorithm and performs better optimizations than state-of-the-art approaches. In fact, it uses a set of exploratory laws that reduces the number of gates. This technique first displays the quantum circuit as a netlist and then preprocesses and simplifies the circuit. Then, it divides the circuit into sub-circuits and optimizes the sub-circuits according to the rules 1-4. In[33], Abdessaied et al. used several algorithms to synthesize reversible functions to quantum circuits and to reduce the number of Hadamard gates. This reduction of the Hadamard gate, reduces the number and depth of T gates, which improves the combined gates. By applying this method, the authors improved the number and depth of T gate by 88% more than other optimization methods. Using different quantum circuit optimization techniques, the aim of this paper is to improve the performance of quantum machine learning circuits and to reduce their cost. To this end we optimized the quantum machine learning circuits in terms of quantum gates and time steps.

Methods

Implementing machine learning algorithms with big data in quantum systems is a major challenge due to the excessive increase in the number of gates, the depth of the circuit, and the execution time of the algorithm. To this end, optimizing quantum circuits is an effective way to overcome these problems. In this section, the details of the optimization algorithm for quantum machine learning circuits are explained. This method is then used to optimize the quantum circuits of two machine learning algorithms, transmission learning and neural networks. Initially, in the preprocessing step, the quantum circuit represented as a list of gates that are applied sequentially. The following transformation rules are then applied to optimize the quantum machine learning circuits.

Rule 1: First, if there is a NOT gate in the circuit, the next gate is checked. In this case, there are three different possibilities for the next gates[13]:

- If the next gate is a TOFFOLI gate: in this case the control qubit of the TOFFOLI is reversed and the NOT gate is removed.
- If the next gate is a NOT gate: in this case the two NOT gates are removed.

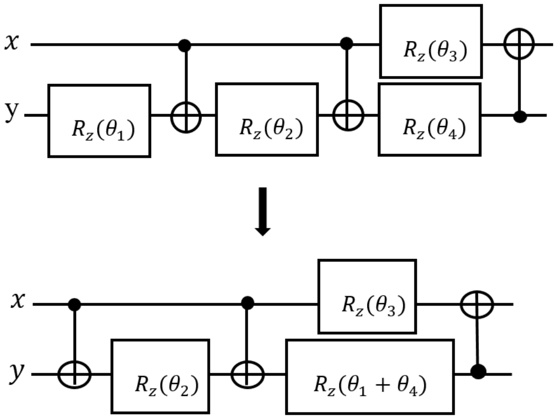


Figure 2. Integration of two rotation gates[13].

- If the next gate is a CNOT gate: in this case the control qubit is reversed and the NOT gate is removed.

Rule 2: Remove gates that are directly adjacent to their inverse. In a two-qubit gate, it is usually possible to simplify or eliminate the gate in the form of quantum circuits by moving it between the gates. In fact, for each U gate in the circuit, the optimizer searches for an instance of $U^{\dagger}$. If present, U is successfully canceled with some instances of $U^{\dagger}$.

Rule 3: For two rotation gates $R_Z(\theta_i)$ and $R_Z(\theta_j)$ that have a shared control line, According to equation (7), we can merge two rotations[34]. For example, in Figure 2 two rotation gates $R_Z(\theta_1)$ and $R_Z(\theta_4)$ can be combined[13]:

$$R(\theta_1).R(\theta_2) = R(\theta_1+\theta_2) \tag{7}$$

$$R(\theta_1).R(\theta_2) = R(\theta_2).R(\theta_1) \tag{8}$$

Rule 4: Because many quantum algorithms can be described using Swap and Bridge gates[35], we transform them into the equivalent circuits consisting of CNOT gates on two consecutive qubits. Figure 3 (a) is the equivalent circuit of Swap gate and Figure 3 (b) is the equivalent circuit of Bridge gate. By breaking down multi-qubit gates into simpler gates, the resulting circuit performs better when using other rules[35].

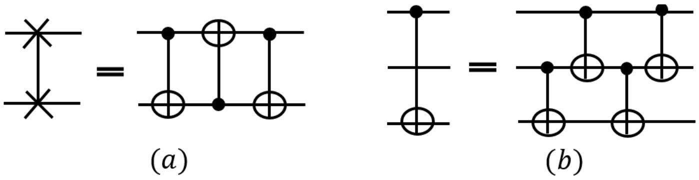
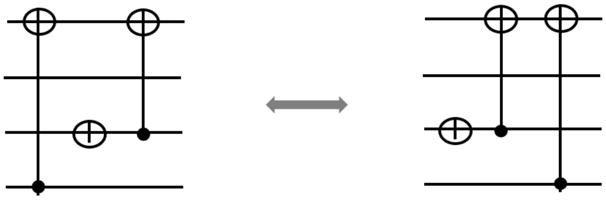


Figure 3. a) Equivalent circuits of Swap gates. b) Equivalent circuits of Bridge gates[35].

In some cases, the gates that can be merged or removed are not placed side by side. By moving the gates according to Figures 4 and 5, the gates are placed side by side and so they can be merged and removed by the above rules. . For this purpose, the Sympy library[36] is used. Sympy is an open source Python library for symbolic mathematics. With the help of this library, complex quantum circuits can be transformed into simple ones. One of the interesting features of this library is producing equivalent circuits. In fact, using library different gates in the circuit are moved around and all locations where the gates can be placed are examined. Then, the rules 1-4 are re-examined by the algorithm and the circuit is simplified if conditions pass.

These defined rules are applied in a loop until no further improvement is obtained. Algorithm 1 and Figure 6 present the steps of our optimization approach using the above rules. Using this framework, we optimized the quantum machine learning



$$(CNOT_{q4,q1} \otimes I \otimes I)(I \otimes I \otimes NOT \otimes I)(CNOT_{q3,q1} \otimes I \otimes I)$$
$$= (NOT \otimes I \otimes I \otimes I)(CNOT_{q3,q1} \otimes I \otimes I)(CNOT_{q4,q1} \otimes I \otimes I)$$

Figure 4. A quantum circuit and its equivalent.

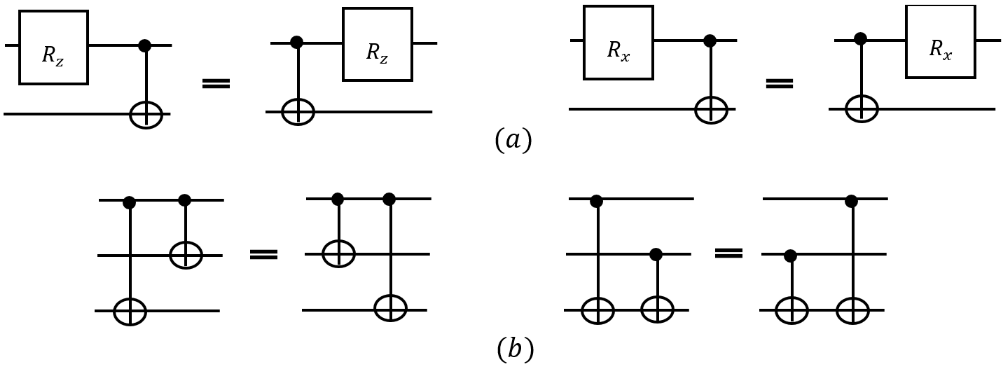


Figure 5. a) Display the equivalent circuit of the rotation gate and the CNOT gate, based on the target qubit and the control qubit of the CNOT. b) Display the equivalent circuit of CNOT gates, based on the target qubit and the control qubit of the CNOT gate[35].

circuit of a classification task for medical diagnosis using quantum transfer learning[25]. This circuit has been tested in several real quantum processors as well as various simulators. This quantum circuit aims at distinguishing a sick person from a healthy person based on computed tomography images. The circuit consists of four steps: The Hadamard gate is first applied to all qubits and then with a the helpf of U operator defined in[25], the classical data is encoded and then entanglement is created The dotted box of Figure 7 shows one application of this operator. Finally, the qubits are measured. Figure 7 demonstrates the quantum circuit of this quantum machine learning algorithm with only one repetition of the coding sub-circuit.

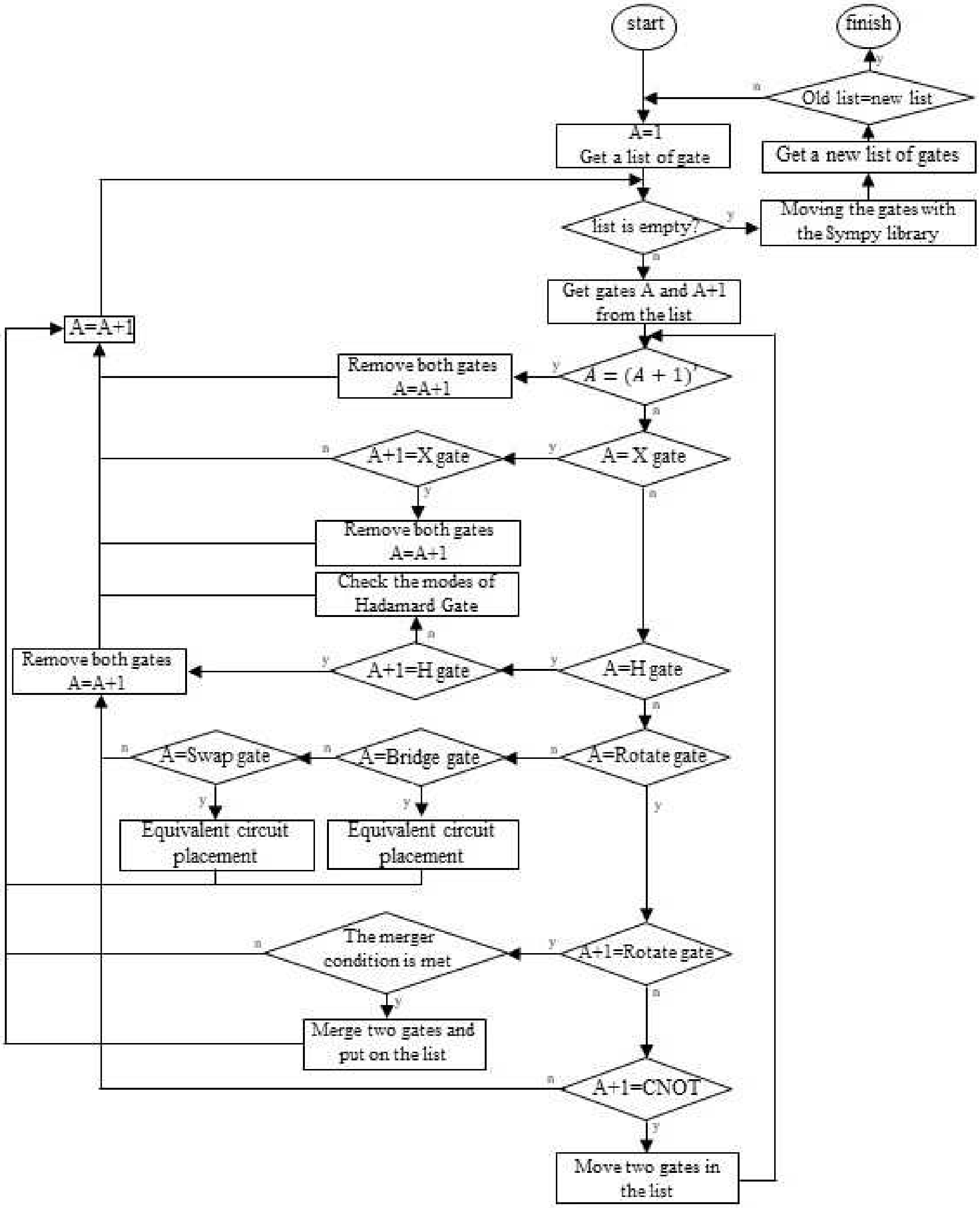


Figure 6. The flowchart of the steps of the optimization approach using rules 1-4.

Results and Discussion

In this section, the experimental results for different quantum circuits and quantum machine learning circuits are presented. The file of these circuits are mostly in QASM format, received as input by Python language and then converted into a matrix by the

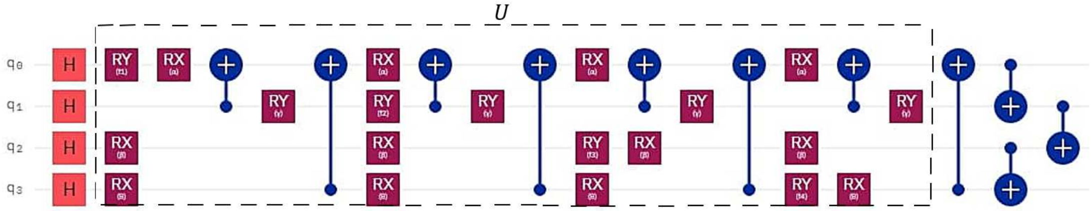


Figure 7. A quantum circuit with a quantum transfer learning method [25].

Algorithm 1 Our proposed algorithm for quantum circuit optimization.

```
        Input: quantum circuit (Qcirc)
        SubCircList = Generate Sub circuit Gateways lists (Qcirc)
        Mtime-step = Gate. Length
        NumSubCirc = SubCircLists. Length
        N=0;
        For t=1 to Mtime-step
            If Gate (t) exist in SubCircLists (N)
                If type-gate(t) = x
                    Function-opt-x (Gate)
                Else if type-gate (t) =Inverse (type-gate (t+1))
                    Gate (t) and Gate (t+1) removed
                Else if type-gate (t) = $R_x||R_y||R_z$
                    Function-opt-$R_{xyz}$ (Gate)
                Else if type-gate (t) = swap
                    Function-opt-swap (gate)
                Else if type-gate (t) = Bridge
                    Function-opt-Bridge (gate)
                Else if quantum circuit not simplified
                    Function-displacement (gate)
          Else if N> NumSubCirc
              N=0
          Else
              N=N+1
          End
    End
```

Qiskit library of Python[43]. Then the proposed optimization techniques are applied to them. The main criterion in evaluating the proposed method is the comparison of the number of gates before and after optimization. Also, another criterion we considered in this work is the execution time of the quantum circuits and the amount of time step reduction. Qiskit library was used in IBM Q quantum computers for output verification. The optimization is then performed and the circuit resulting from the optimization was given to the simulator and the number of gates is calculated after the simulation. Obviously, in order to obtain the simulation results, the pre-optimization and post-optimization simulations must be the same. As the number of gates decreases from the initial value after applying the optimization model, the speed of the algorithm and its implementation cost improve. In this case, the proposed model will be a more efficient model. In order to verify our approach, we first tested our approach on different general quantum circuits and the results are shown in Table 1. In this table, each column is the corresponding quantum circuit and for each circuit we showed the improvement caused by our optimization approach.

Quantum circuit	#of qubits	Pre-optimization						Improvement(%)			Speed Up	Ref
		Execusion time Imp. (S)	#of Time steps	#of Gates	Execution time Imp. (S)	#of Time steps	#of Gates	Execution time Imp. (S)	#of Time steps	#of Gates		
VQC	3	6.4	13	21	6.3	13	18	1.56	0	14.28	0.1	[37]
Grover 1	4	6.4	26	41	6.2	16	21	3.12	38.46	48.78	0.2	[38]
Grover 2	4	6.7	21	39	5.9	17	27	11.94	19.09	30.76	0.8	[38]
QAOA	2	6.7	7	11	6.2	5	7	7.46	28.57	36.36	0.5	[39]
test circuit K Means	3	6.5	14	26	6.2	14	24	4.6	0	7.7	0.3	[40]
TNN	4	6	10	27	5.9	8	14	1.7	20	48.14	0.1	[41]
NNK	4	7.8	20	33	4.7	8	13	39.74	60	60.60	3.1	[42]

Table 1. Results of implementation of the proposed method on quantum circuits.

In the proposed method, assuming that the number of time steps is N and the number of qubits is Q, the time complexity of the algorithm is obtained as O(NQ). As shown in Table 1, applying our method to quantum circuits reduces the number of gates, time steps, and execution time of the quantum circuits significantly. At the second part of the experiments, our optimization approach was applied on the quantum machine learning circuits. One of these circuits uses transmission learning method for a potential application in medical diagnosis. By applying the proposed method to the above quantum circuit, only the U-shaped part of the circuit improves as shown in Figure 8. In Figure 8 (a) it can be seen that the primary circuit has 17 time steps with 28 quantum gates. (b) of this, improved circuit diagram shows that in addition to a 10.7% reduction in the number of gates, the time step is reduced to 14 units. This is the amount of quantum cost reduction for one repetition below the U-circuit in the main circuit. For cases where this sub-circuit is repeated many times in the main circuit, the rate of improvement increases. In this case, by applying the proposed method on circuits with big data, desirable results will be obtained. The results of the implementation of the proposed method on the quantum circuit of transfer learning are shown in Figure 9 before and after optimization.

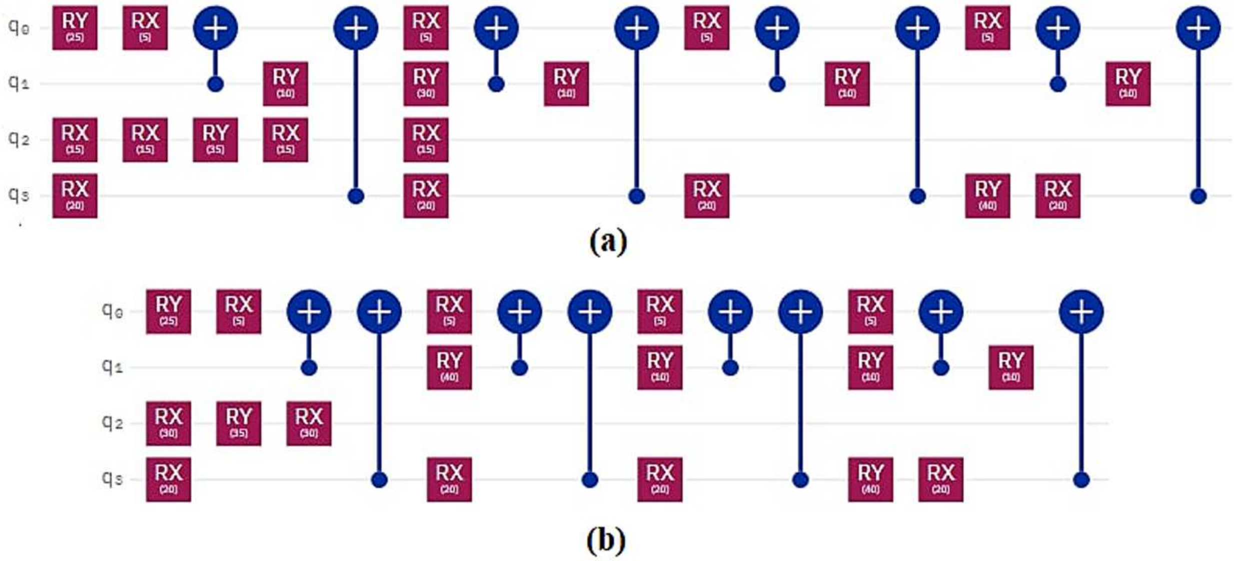


Figure 8. Demonstration of original and improved quantum transfer learning circuits. Diagram (a) shows the non-optimal circuit, performed in 17 time steps, and diagram (b) shows the improved circuit with reducing the number of gates by 10.7%, and the time steps reduced to 14.

We verified the outputs in IBM Q and the results are demonstrated in Figure 10. Figure 10 (a) is the output of the original circuit and Figure 10 (b) is the output after we applied our optimization algorithm. Since the output is the same in both cases, the transformation has done correctly.

The next quantum machine learning circuit that we used in this work is the quantum circuit of the neural network for cancer detection, which used the design and operation of a classical neural network but it is a quantum neural network capable of working on a 10 qubit system. By demonstrating network performance, they have tried to use the basic principles of machine learning to manage data. The graphical representation of this circuit is shown in Figure 11. Figure 11 (a) shows the original circuit, which is implemented in 31 time steps with 67 quantum gates. Figure 11 (b) shows the improved circuit, which in addition to a 14.9% reduction in the number of gates, reduces its time steps to 16. The comparison result of applying the proposed method to this circuit is shown in Figure 12.

The output results of the circuits are shown in Figure 13. Figure 13 (a) is the output of the original circuit and Figure 13 (b) is the output after we applied our optimization algorithm. Since the output is the same in both cases, the transformation of the proposed optimization is correct.

Conclusion

Realizing machine learning algorithms in a quantum system for big data is a real challenge but with remarkable advantages of using quantum computers. In quantum circuits, as the number of gates increases, the number of time steps and execution time is also increased, which is why optimizing quantum circuits is an effective way to overcome these problems. In this study, a new general framework of quantum circuit optimization was presented and in particular, quantum machine learning algorithms for big data were investigated in order to improve their quantum circuit model which in turn leads to the improvement and reduction in the number of required quantum computation resources. In fact, by applying the proposed method, quantum circuits were implemented in less time than the original circuits, with the same functionality of the original design. In addition, applying this method also reduces the quantum costs. Several quantum circuits with different functionality and algorithms were

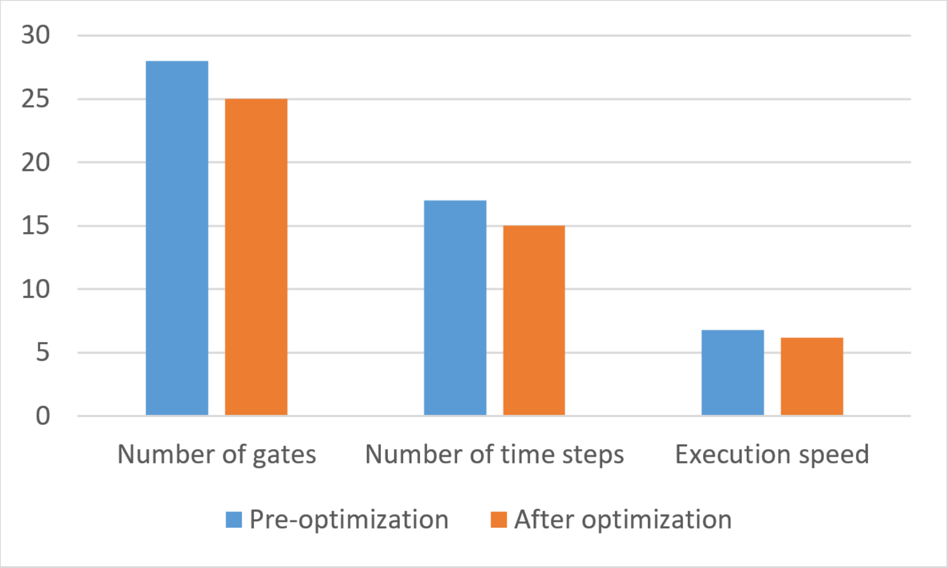
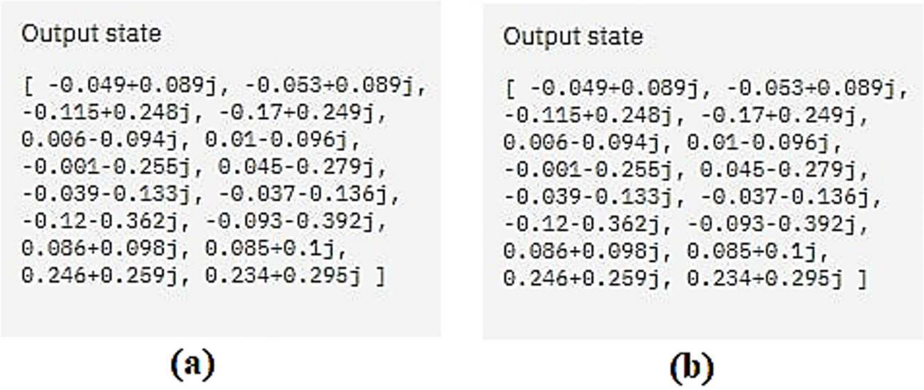


Figure 9. Demonstration of optimized and non-optimized diagrams of quantum transfer learning circuits.



Output state

```
[ -0.049+0.089j, -0.053+0.089j,
-0.115+0.248j, -0.17+0.249j,
0.006-0.094j, 0.01-0.096j,
-0.001-0.255j, 0.045-0.279j,
-0.039-0.133j, -0.037-0.136j,
-0.12-0.362j, -0.093-0.392j,
0.086+0.098j, 0.085+0.1j,
0.246+0.259j, 0.234+0.295j ]
```

(a)

Output state

```
[ -0.049+0.089j, -0.053+0.089j,
-0.115+0.248j, -0.17+0.249j,
0.006-0.094j, 0.01-0.096j,
-0.001-0.255j, 0.045-0.279j,
-0.039-0.133j, -0.037-0.136j,
-0.12-0.362j, -0.093-0.392j,
0.086+0.098j, 0.085+0.1j,
0.246+0.259j, 0.234+0.295j ]
```

(b)

Figure 10. The simulation result of the output of the quantum transfer learning circuit before and after optimization in (a) and (b), respectively. The results are identical.

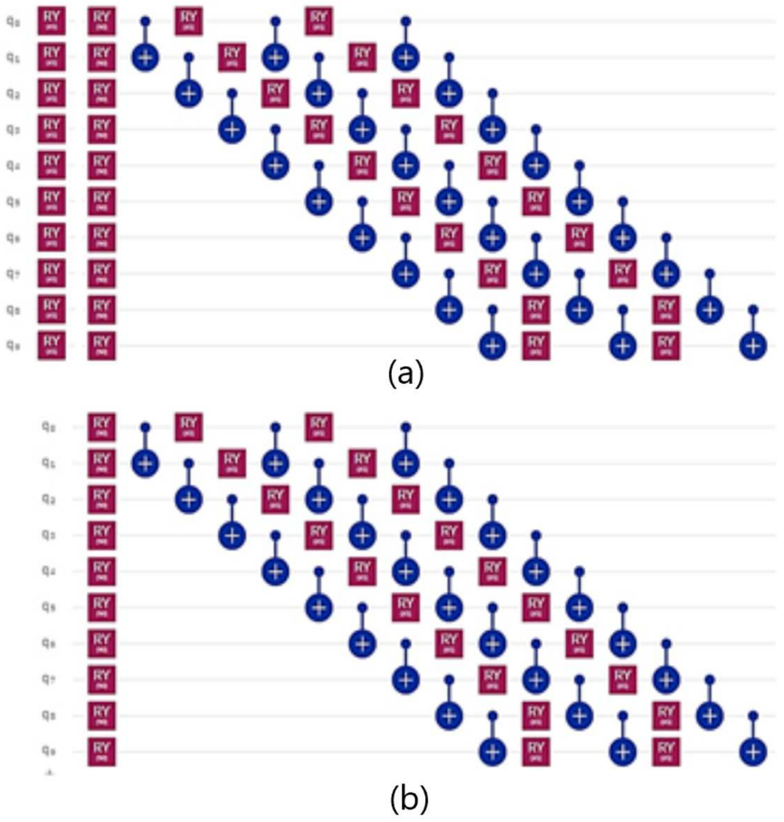


Figure 11. Demonstration of original [27], and (b) improved quantum neural network circuits used for cancer diagnosis. The non-optimal circuit (a) is executed in 31 time steps, but the improved circuit, which has a reduction of 14.9% in the number of gates, its time step is reduced to 16.

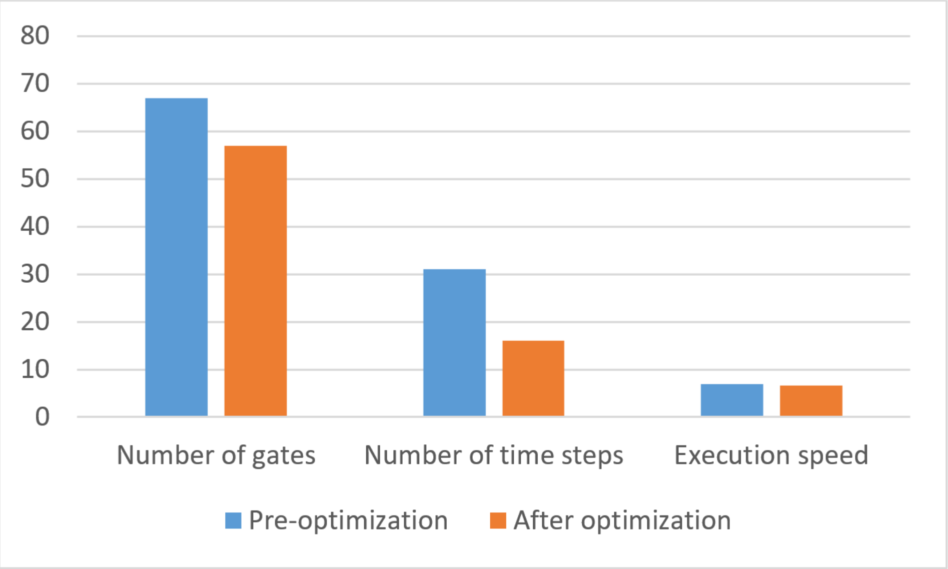
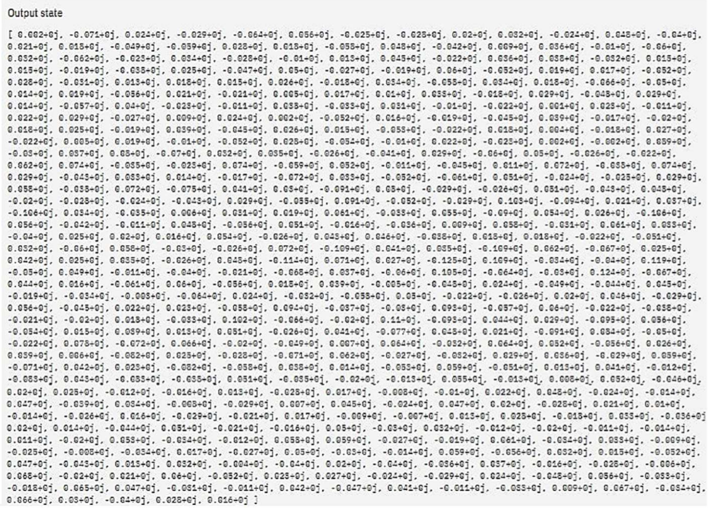


Figure 12. Comparison between optimized and non-optimized quantum neural networks circuits.



Output state

```
[ 0.062+0j, -0.071+0j, 0.024+0j, -0.029+0j, -0.064+0j, 0.056+0j, -0.025+0j, -0.028+0j, 0.02+0j, 0.032+0j, -0.024+0j, 0.048+0j, -0.04+0j,
0.021+0j, 0.015+0j, -0.049+0j, -0.059+0j, 0.028+0j, 0.018+0j, -0.055+0j, 0.048+0j, -0.042+0j, 0.069+0j, 0.036+0j, -0.01+0j, -0.06+0j,
0.032+0j, -0.062+0j, -0.023+0j, 0.034+0j, -0.028+0j, -0.01+0j, 0.013+0j, 0.045+0j, -0.022+0j, 0.036+0j, 0.038+0j, -0.032+0j, 0.015+0j,
0.015+0j, -0.019+0j, -0.038+0j, 0.025+0j, -0.047+0j, 0.05+0j, -0.027+0j, -0.019+0j, 0.06+0j, -0.052+0j, 0.019+0j, 0.017+0j, -0.052+0j,
0.028+0j, -0.031+0j, 0.013+0j, 0.018+0j, 0.015+0j, 0.026+0j, -0.018+0j, 0.034+0j, -0.055+0j, 0.034+0j, 0.018+0j, -0.066+0j, -0.05+0j,
0.014+0j, 0.019+0j, -0.056+0j, 0.021+0j, -0.021+0j, 0.005+0j, 0.017+0j, 0.01+0j, 0.035+0j, -0.018+0j, 0.029+0j, -0.048+0j, 0.029+0j,
0.014+0j, -0.057+0j, 0.04+0j, -0.023+0j, -0.011+0j, 0.038+0j, -0.033+0j, 0.031+0j, -0.01+0j, -0.022+0j, 0.001+0j, 0.023+0j, -0.011+0j,
0.022+0j, 0.029+0j, -0.027+0j, 0.009+0j, 0.024+0j, 0.002+0j, -0.052+0j, 0.016+0j, -0.019+0j, -0.045+0j, 0.039+0j, -0.017+0j, -0.02+0j,
0.018+0j, 0.025+0j, -0.019+0j, 0.039+0j, -0.045+0j, 0.026+0j, 0.015+0j, -0.053+0j, -0.022+0j, 0.018+0j, 0.004+0j, -0.018+0j, 0.027+0j,
-0.022+0j, 0.005+0j, 0.019+0j, -0.01+0j, -0.052+0j, 0.028+0j, -0.054+0j, -0.01+0j, 0.022+0j, -0.023+0j, 0.002+0j, -0.002+0j, 0.059+0j,
-0.03+0j, 0.037+0j, 0.03+0j, -0.07+0j, 0.032+0j, 0.035+0j, -0.026+0j, -0.041+0j, 0.029+0j, -0.06+0j, 0.05+0j, -0.026+0j, -0.022+0j,
0.062+0j, 0.074+0j, -0.035+0j, -0.023+0j, 0.074+0j, -0.059+0j, 0.052+0j, -0.011+0j, -0.045+0j, 0.011+0j, 0.072+0j, -0.035+0j, 0.074+0j,
0.029+0j, -0.043+0j, 0.033+0j, 0.014+0j, -0.017+0j, -0.072+0j, 0.033+0j, -0.052+0j, -0.061+0j, 0.051+0j, -0.024+0j, -0.025+0j, 0.029+0j,
0.058+0j, -0.038+0j, 0.072+0j, -0.075+0j, 0.041+0j, 0.03+0j, -0.091+0j, 0.05+0j, -0.029+0j, -0.026+0j, 0.051+0j, -0.043+0j, 0.045+0j,
-0.02+0j, -0.028+0j, -0.024+0j, -0.043+0j, 0.029+0j, -0.055+0j, 0.091+0j, -0.052+0j, -0.029+0j, 0.103+0j, -0.094+0j, 0.021+0j, 0.037+0j,
-0.106+0j, 0.034+0j, -0.035+0j, 0.006+0j, 0.031+0j, 0.019+0j, 0.061+0j, -0.033+0j, 0.055+0j, -0.09+0j, 0.054+0j, 0.026+0j, -0.106+0j,
0.056+0j, -0.042+0j, -0.011+0j, 0.048+0j, -0.056+0j, 0.051+0j, -0.016+0j, -0.036+0j, 0.009+0j, 0.058+0j, -0.031+0j, 0.061+0j, 0.033+0j,
-0.04+0j, 0.025+0j, 0.02+0j, 0.016+0j, 0.054+0j, -0.026+0j, 0.043+0j, 0.046+0j, -0.038+0j, 0.018+0j, 0.018+0j, -0.022+0j, -0.051+0j,
0.032+0j, -0.06+0j, 0.058+0j, -0.03+0j, -0.026+0j, 0.072+0j, -0.109+0j, 0.041+0j, 0.035+0j, -0.109+0j, 0.062+0j, -0.067+0j, 0.025+0j,
0.042+0j, 0.025+0j, 0.035+0j, -0.026+0j, 0.045+0j, -0.114+0j, 0.071+0j, 0.027+0j, -0.125+0j, 0.109+0j, -0.034+0j, -0.04+0j, 0.119+0j,
-0.05+0j, 0.049+0j, -0.011+0j, -0.04+0j, -0.021+0j, -0.068+0j, 0.037+0j, -0.06+0j, 0.105+0j, -0.064+0j, -0.03+0j, 0.124+0j, -0.067+0j,
0.044+0j, 0.016+0j, -0.061+0j, 0.06+0j, -0.056+0j, 0.018+0j, 0.039+0j, -0.005+0j, -0.045+0j, 0.024+0j, -0.049+0j, -0.044+0j, 0.045+0j,
-0.019+0j, -0.034+0j, -0.003+0j, -0.064+0j, 0.024+0j, -0.032+0j, -0.053+0j, 0.05+0j, -0.022+0j, -0.026+0j, 0.02+0j, 0.046+0j, -0.029+0j,
0.056+0j, -0.045+0j, 0.022+0j, 0.023+0j, -0.058+0j, 0.094+0j, -0.037+0j, -0.03+0j, 0.093+0j, -0.057+0j, 0.06+0j, -0.022+0j, -0.038+0j,
-0.021+0j, -0.02+0j, 0.018+0j, -0.033+0j, 0.102+0j, -0.066+0j, -0.02+0j, 0.11+0j, -0.093+0j, 0.044+0j, 0.029+0j, -0.095+0j, 0.056+0j,
-0.054+0j, 0.015+0j, 0.059+0j, 0.013+0j, 0.051+0j, -0.026+0j, 0.041+0j, -0.077+0j, 0.048+0j, 0.021+0j, -0.091+0j, 0.084+0j, -0.05+0j,
-0.022+0j, 0.078+0j, -0.072+0j, 0.066+0j, -0.02+0j, -0.049+0j, 0.007+0j, 0.064+0j, -0.032+0j, 0.064+0j, 0.052+0j, -0.056+0j, 0.026+0j,
0.059+0j, 0.066+0j, -0.062+0j, 0.025+0j, -0.028+0j, -0.071+0j, 0.062+0j, -0.027+0j, -0.032+0j, 0.029+0j, 0.036+0j, -0.029+0j, 0.059+0j,
-0.071+0j, 0.042+0j, 0.023+0j, -0.082+0j, -0.058+0j, 0.038+0j, 0.014+0j, -0.053+0j, 0.059+0j, -0.051+0j, 0.013+0j, 0.041+0j, -0.012+0j,
-0.083+0j, 0.043+0j, -0.033+0j, -0.035+0j, 0.051+0j, -0.035+0j, -0.02+0j, -0.013+0j, 0.055+0j, -0.013+0j, 0.008+0j, 0.052+0j, -0.046+0j,
0.02+0j, 0.025+0j, -0.012+0j, -0.016+0j, 0.013+0j, -0.025+0j, 0.017+0j, -0.008+0j, -0.01+0j, 0.022+0j, 0.048+0j, -0.024+0j, -0.014+0j,
0.047+0j, -0.039+0j, 0.034+0j, -0.063+0j, -0.029+0j, 0.007+0j, 0.045+0j, -0.024+0j, 0.047+0j, 0.02+0j, -0.028+0j, 0.021+0j, 0.01+0j,
-0.014+0j, -0.026+0j, 0.016+0j, -0.029+0j, -0.021+0j, 0.017+0j, -0.009+0j, -0.007+0j, 0.013+0j, 0.025+0j, -0.018+0j, 0.033+0j, -0.036+0j
0.02+0j, 0.014+0j, -0.044+0j, 0.051+0j, -0.021+0j, -0.016+0j, 0.05+0j, -0.03+0j, 0.032+0j, -0.012+0j, -0.02+0j, -0.011+0j, -0.014+0j,
0.011+0j, -0.02+0j, 0.053+0j, -0.034+0j, -0.012+0j, 0.055+0j, 0.059+0j, -0.027+0j, -0.019+0j, 0.061+0j, -0.034+0j, 0.033+0j, -0.009+0j,
-0.025+0j, -0.008+0j, -0.034+0j, 0.017+0j, -0.027+0j, 0.05+0j, -0.03+0j, -0.014+0j, 0.059+0j, -0.056+0j, 0.032+0j, 0.015+0j, -0.052+0j,
0.047+0j, -0.043+0j, 0.013+0j, 0.032+0j, -0.004+0j, -0.04+0j, 0.02+0j, -0.04+0j, -0.036+0j, 0.037+0j, -0.016+0j, -0.028+0j, -0.006+0j,
0.068+0j, -0.02+0j, 0.021+0j, 0.06+0j, -0.052+0j, 0.023+0j, 0.027+0j, -0.024+0j, -0.029+0j, 0.024+0j, -0.048+0j, 0.056+0j, -0.053+0j,
-0.018+0j, 0.065+0j, 0.047+0j, -0.031+0j, -0.011+0j, 0.042+0j, -0.047+0j, 0.041+0j, -0.011+0j, -0.033+0j, 0.009+0j, 0.067+0j, -0.054+0j,
0.066+0j, 0.03+0j, -0.04+0j, 0.028+0j, 0.016+0j ]
```

a

Output state

```
[ 0.062+0j, -0.071+0j, 0.024+0j, -0.029+0j, -0.064+0j, 0.056+0j, -0.025+0j, -0.028+0j, 0.02+0j, 0.032+0j, -0.024+0j, 0.048+0j, -0.04+0j,
0.021+0j, 0.015+0j, -0.049+0j, -0.059+0j, 0.028+0j, 0.018+0j, -0.055+0j, 0.048+0j, -0.042+0j, 0.069+0j, 0.036+0j, -0.01+0j, -0.06+0j,
0.032+0j, -0.062+0j, -0.023+0j, 0.034+0j, -0.028+0j, -0.01+0j, 0.013+0j, 0.045+0j, -0.022+0j, 0.036+0j, 0.038+0j, -0.032+0j, 0.015+0j,
0.015+0j, -0.019+0j, -0.038+0j, 0.025+0j, -0.047+0j, 0.05+0j, -0.027+0j, -0.019+0j, 0.06+0j, -0.052+0j, 0.019+0j, 0.017+0j, -0.052+0j,
0.028+0j, -0.031+0j, 0.013+0j, 0.018+0j, 0.015+0j, 0.026+0j, -0.018+0j, 0.034+0j, -0.055+0j, 0.034+0j, 0.018+0j, -0.066+0j, -0.05+0j,
0.014+0j, 0.019+0j, -0.056+0j, 0.021+0j, -0.021+0j, 0.005+0j, 0.017+0j, 0.01+0j, 0.035+0j, -0.018+0j, 0.029+0j, -0.048+0j, 0.029+0j,
0.014+0j, -0.057+0j, 0.04+0j, -0.023+0j, -0.011+0j, 0.038+0j, -0.033+0j, 0.031+0j, -0.01+0j, -0.022+0j, 0.001+0j, 0.023+0j, -0.011+0j,
0.022+0j, 0.029+0j, -0.027+0j, 0.009+0j, 0.024+0j, 0.002+0j, -0.052+0j, 0.016+0j, -0.019+0j, -0.045+0j, 0.039+0j, -0.017+0j, -0.02+0j,
0.018+0j, 0.025+0j, -0.019+0j, 0.039+0j, -0.045+0j, 0.026+0j, 0.015+0j, -0.053+0j, -0.022+0j, 0.018+0j, 0.004+0j, -0.018+0j, 0.027+0j,
-0.022+0j, 0.005+0j, 0.019+0j, -0.01+0j, -0.052+0j, 0.028+0j, -0.054+0j, -0.01+0j, 0.022+0j, -0.023+0j, 0.002+0j, -0.002+0j, 0.059+0j,
-0.03+0j, 0.037+0j, 0.03+0j, -0.07+0j, 0.032+0j, 0.035+0j, -0.026+0j, -0.041+0j, 0.029+0j, -0.06+0j, 0.05+0j, -0.026+0j, -0.022+0j,
0.062+0j, 0.074+0j, -0.035+0j, -0.023+0j, 0.074+0j, -0.059+0j, 0.052+0j, -0.011+0j, -0.045+0j, 0.011+0j, 0.072+0j, -0.035+0j, 0.074+0j,
0.029+0j, -0.043+0j, 0.033+0j, 0.014+0j, -0.017+0j, -0.072+0j, 0.033+0j, -0.052+0j, -0.061+0j, 0.051+0j, -0.024+0j, -0.025+0j, 0.029+0j,
0.058+0j, -0.038+0j, 0.072+0j, -0.075+0j, 0.041+0j, 0.03+0j, -0.091+0j, 0.05+0j, -0.029+0j, -0.026+0j, 0.051+0j, -0.043+0j, 0.045+0j,
-0.02+0j, -0.028+0j, -0.024+0j, -0.043+0j, 0.029+0j, -0.055+0j, 0.091+0j, -0.052+0j, -0.029+0j, 0.103+0j, -0.094+0j, 0.021+0j, 0.037+0j,
-0.106+0j, 0.034+0j, -0.035+0j, 0.006+0j, 0.031+0j, 0.019+0j, 0.061+0j, -0.033+0j, 0.055+0j, -0.09+0j, 0.054+0j, 0.026+0j, -0.106+0j,
0.056+0j, -0.042+0j, -0.011+0j, 0.048+0j, -0.056+0j, 0.051+0j, -0.016+0j, -0.036+0j, 0.009+0j, 0.058+0j, -0.031+0j, 0.061+0j, 0.033+0j,
-0.04+0j, 0.025+0j, 0.02+0j, 0.016+0j, 0.054+0j, -0.026+0j, 0.043+0j, 0.046+0j, -0.038+0j, 0.018+0j, 0.018+0j, -0.022+0j, -0.051+0j,
0.032+0j, -0.06+0j, 0.058+0j, -0.03+0j, -0.026+0j, 0.072+0j, -0.109+0j, 0.041+0j, 0.035+0j, -0.109+0j, 0.062+0j, -0.067+0j, 0.025+0j,
0.042+0j, 0.025+0j, 0.035+0j, -0.026+0j, 0.045+0j, -0.114+0j, 0.071+0j, 0.027+0j, -0.125+0j, 0.109+0j, -0.034+0j, -0.04+0j, 0.119+0j,
-0.05+0j, 0.049+0j, -0.011+0j, -0.04+0j, -0.021+0j, -0.068+0j, 0.037+0j, -0.06+0j, 0.105+0j, -0.064+0j, -0.03+0j, 0.124+0j, -0.067+0j,
0.044+0j, 0.016+0j, -0.061+0j, 0.06+0j, -0.056+0j, 0.018+0j, 0.039+0j, -0.005+0j, -0.045+0j, 0.024+0j, -0.049+0j, -0.044+0j, 0.045+0j,
-0.019+0j, -0.034+0j, -0.003+0j, -0.064+0j, 0.024+0j, -0.032+0j, -0.053+0j, 0.05+0j, -0.022+0j, -0.026+0j, 0.02+0j, 0.046+0j, -0.029+0j,
0.056+0j, -0.045+0j, 0.022+0j, 0.023+0j, -0.058+0j, 0.094+0j, -0.037+0j, -0.03+0j, 0.093+0j, -0.057+0j, 0.06+0j, -0.022+0j, -0.038+0j,
-0.021+0j, -0.02+0j, 0.018+0j, -0.033+0j, 0.102+0j, -0.066+0j, -0.02+0j, 0.11+0j, -0.093+0j, 0.044+0j, 0.029+0j, -0.095+0j, 0.056+0j,
-0.054+0j, 0.015+0j, 0.059+0j, 0.013+0j, 0.051+0j, -0.026+0j, 0.041+0j, -0.077+0j, 0.048+0j, 0.021+0j, -0.091+0j, 0.084+0j, -0.05+0j,
-0.022+0j, 0.078+0j, -0.072+0j, 0.066+0j, -0.02+0j, -0.049+0j, 0.007+0j, 0.064+0j, -0.032+0j, 0.064+0j, 0.052+0j, -0.056+0j, 0.026+0j,
0.059+0j, 0.066+0j, -0.062+0j, 0.025+0j, -0.028+0j, -0.071+0j, 0.062+0j, -0.027+0j, -0.032+0j, 0.029+0j, 0.036+0j, -0.029+0j, 0.059+0j,
-0.071+0j, 0.042+0j, 0.023+0j, -0.082+0j, -0.058+0j, 0.038+0j, 0.014+0j, -0.053+0j, 0.059+0j, -0.051+0j, 0.013+0j, 0.041+0j, -0.012+0j,
-0.083+0j, 0.043+0j, -0.033+0j, -0.035+0j, 0.051+0j, -0.035+0j, -0.02+0j, -0.013+0j, 0.055+0j, -0.013+0j, 0.008+0j, 0.052+0j, -0.046+0j,
0.02+0j, 0.025+0j, -0.012+0j, -0.016+0j, 0.013+0j, -0.025+0j, 0.017+0j, -0.008+0j, -0.01+0j, 0.022+0j, 0.048+0j, -0.024+0j, -0.014+0j,
0.047+0j, -0.039+0j, 0.034+0j, -0.063+0j, -0.029+0j, 0.007+0j, 0.045+0j, -0.024+0j, 0.047+0j, 0.02+0j, -0.028+0j, 0.021+0j, 0.01+0j,
-0.014+0j, -0.026+0j, 0.016+0j, -0.029+0j, -0.021+0j, 0.017+0j, -0.009+0j, -0.007+0j, 0.013+0j, 0.025+0j, -0.018+0j, 0.033+0j, -0.036+0j,
0.02+0j, 0.014+0j, -0.044+0j, 0.051+0j, -0.021+0j, -0.016+0j, 0.05+0j, -0.03+0j, 0.032+0j, -0.012+0j, -0.02+0j, -0.011+0j, -0.014+0j,
0.011+0j, -0.02+0j, 0.053+0j, -0.034+0j, -0.012+0j, 0.055+0j, 0.059+0j, -0.027+0j, -0.019+0j, 0.061+0j, -0.034+0j, 0.033+0j, -0.009+0j,
-0.025+0j, -0.008+0j, -0.034+0j, 0.017+0j, -0.027+0j, 0.05+0j, -0.03+0j, -0.014+0j, 0.059+0j, -0.056+0j, 0.032+0j, 0.015+0j, -0.052+0j,
0.047+0j, -0.043+0j, 0.013+0j, 0.032+0j, -0.004+0j, -0.04+0j, 0.02+0j, -0.04+0j, -0.036+0j, 0.037+0j, -0.016+0j, -0.028+0j, -0.006+0j,
0.068+0j, -0.02+0j, 0.021+0j, 0.06+0j, -0.052+0j, 0.023+0j, 0.027+0j, -0.024+0j, -0.029+0j, 0.024+0j, -0.048+0j, 0.056+0j, -0.053+0j,
-0.018+0j, 0.065+0j, 0.047+0j, -0.031+0j, -0.011+0j, 0.042+0j, -0.047+0j, 0.041+0j, -0.011+0j, -0.033+0j, 0.009+0j, 0.067+0j, -0.054+0j,
0.066+0j, 0.03+0j, -0.04+0j, 0.028+0j, 0.016+0j ]
```

b

Figure 13. The simulation result of the output of the quantum neural network circuit before and after optimization in (a) and (b) respectively. The results are identical.

used to evaluate the proposed method. The results of the improved circuits showed that the number of quantum gate, the time steps, and the execution time in the evaluated circuits were reduced. In particular, the proposed method was investigated on the quantum circuits of transfer learning and neural network. Our approach reduced the number of time steps and the gates by 17.65% and 10.71% respectively in transfer learning circuit and also reduced the number of time steps and the gate by 48.38% and 14.9% respectively in neural network circuit. More importantly, this was the amount of reduction for one iteration of the U-subcircuit in the main circuit of the transfer learning algorithm. So, for the cases where this sub-circuit was repeated more often in the main circuit, the optimization is even more. So, by applying the proposed method on circuits with big data, better results would be obtained.

Author contributions statement

T.S. contributed to implementation of the research and the main conceptual ideas, and performing the experiments. M.Z. contributed to the original idea, took the lead in writing the manuscript with input from all authors, and were in charge of overall direction. P.P. supervised and commented on the manuscript and contributed to the final version. M.A. contributed to the final version of the manuscript and commented on the manuscript. V.S. contributed to the analysis of the results and also contributed to the writing of the manuscript.

References

1. Humble TS, Thapliyal H, Munoz-Coreas E, Mohiyaddin FA, Bennink RS. Quantum computing circuits and devices. IEEE Design & Test. 2019 Apr 3;36(3):69-94.
2. Benenti G, Casati G, Strini G. Principles of Quantum Computation and Information-Volume II: Basic Tools and Special Topics. World Scientific Publishing Company; 2007 Mar 21.
3. Schuld M, Petruccione F. Supervised learning with quantum computers. Berlin: Springer; 2018 Aug 30.
4. Grover LK. A fast quantum mechanical algorithm for database search. InProceedings of the twenty-eighth annual ACM symposium on Theory of computing 1996 Jul 1 (pp. 212-219).
5. Humble TS, Thapliyal H, Munoz-Coreas E, Mohiyaddin FA, Bennink RS. Quantum computing circuits and devices. IEEE Design & Test. 2019 Apr 3;36(3):69-94.
6. Gyongyosi L, Imre S. A survey on quantum computing technology. Computer Science Review. 2019 Feb 1;31:51-71.
7. Dang Y, Jiang N, Hu H, Ji Z, Zhang W. Image classification based on quantum K-Nearest-Neighbor algorithm. Quantum Information Processing. 2018 Sep;17(9):1-8.

8. Ruan Y, Xue X, Liu H, Tan J, Li X. Quantum algorithm for k-nearest neighbors classification based on the metric of hamming distance. International Journal of Theoretical Physics. 2017 Nov;56(11):3496-507.

9. Savchuk MM, Fesenko AV. Quantum Computing: Survey and Analysis. Cybernetics and Systems Analysis. 2019 Jan;55(1):10-21.

10. Gilyén A, Arunachalam S, Wiebe N. Optimizing quantum optimization algorithms via faster quantum gradient computation. InProceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms 2019 (pp. 1425-1444). Society for Industrial and Applied Mathematics.

11. Thapliyal H, Ranganathan N. Design of reversible sequential circuits optimizing quantum cost, delay, and garbage outputs. ACM Journal on Emerging Technologies in Computing Systems (JETC). 2010 Dec 21;6(4):1-31.

12. Häner T, Hoefler T, Troyer M. Using Hoare logic for quantum circuit optimization. ArXiv e-prints. 2018 Sep.

13. Childs AM, Maslov D. Automated optimization of large quantum circuits with continuous parameters. arXiv preprint arXiv:1710.07345. 2017 Oct.

14. Schuld M, Sinayskiy I, Petruccione F. An introduction to quantum machine learning. Contemporary Physics. 2015 Apr 3;56(2):172-85.

15. Hagouel PI, Karafyllidis IG. Quantum computers: Registers, gates and algorithms. In2012 28th International Conference on Microelectronics Proceedings 2012 May 13 (pp. 15-21). IEEE.

16. Soklakov AN, Schack R. Efficient state preparation for a register of quantum bits. Physical review A. 2006 Jan 9;73(1):012307.

17. Giovannetti V, Lloyd S, Maccone L. Quantum random access memory. Physical review letters. 2008 Apr 21;100(16):160501.

18. Lloyd S, Garnerone S, Zanardi P. Quantum algorithms for topological and geometric analysis of data. Nature communications. 2016 Jan 25;7(1):1-7.

19. Wiebe N, Granade C, Ferrie C, Cory D. Quantum Hamiltonian learning using imperfect quantum resources. Physical Review A. 2014 Apr 17;89(4):042314.

20. Grover LK. A fast quantum mechanical algorithm for database search. InProceedings of the twenty-eighth annual ACM symposium on Theory of computing 1996 Jul 1 (pp. 212-219).

21. Buhrman H, Cleve R, Watrous J, De Wolf R. Quantum fingerprinting. Physical Review Letters. 2001 Sep 26;87(16):167902.

22. Kaye P. Reversible addition circuit using one ancillary bit with application to quantum computing. arXiv preprint quant-ph/0408173. 2004 Aug 28.

23. Saeedi S, Arodz T. Quantum sparse support vector machines. arXiv preprint arXiv:1902.01879. 2019 Feb 5.

24. Rebentrost P, Mohseni M, Lloyd S. Quantum support vector machine for big data classification. Physical review letters. 2014 Sep 25;113(13):130503.

25. Acar E, Yilmaz I. COVID-19 detection on IBM quantum computer with classical-quantum transfer learning. Turkish Journal of Electrical Engineering & Computer Sciences. 2021 Jan 27;29(1):46-61.

26. Zen R. et al.. Transfer learning for scalability of neural-network quantum states. Physical Review E. 2020 May 5;101(5):053301.

27. Mishra, N., Bisarya, A., Kumar, S., Behera, BK., Mukhopadhyay, S., Panigrahi PK. Cancer Detection Using Quantum Neural Networks: A Demonstration on a Quantum Computer. arXiv preprint arXiv:1911.00504. 2019 Nov 1.

28. Bae JH, Alsing PM, Ahn D, Miller WA. Quantum circuit optimization using quantum Karnaugh map. Scientific reports. 2020 Sep 24;10(1):1-8.

29. Basak A, Sadhu A, Das K, Sharma KK. Cost Optimization Technique for Quantum Circuits. International Journal of Theoretical Physics. 2019 Sep;58(9):3158-79.

30. Li L, Fan M, Coram M, Riley P, Leichenauer S. Quantum optimization with a novel gibbs objective function and ansatz architecture search. Physical Review Research. 2020 Apr 24;2(2):023074.

31. Alam M, Ash-Saki A, Ghosh S. Accelerating quantum approximate optimization algorithm using machine learning. In2020 Design, Automation & Test in Europe Conference & Exhibition (DATE) 2020 Mar 9 (pp. 686-689). IEEE.

32. Hoare CA. An axiomatic basis for computer programming. Communications of the ACM. 1969 Oct 1;12(10):576-80.

33. Abdessaied N, Soeken M, Drechsler R. Quantum circuit optimization by Hadamard gate reduction. InInternational Conference on Reversible Computation 2014 Jul 10 (pp. 149-162). Springer, Cham.

34. Zomorodi-Moghadam M., Navi K. Rotation-based design and synthesis of quantum circuits. Journal of Circuits, Systems and Computers. 2016 Dec 25;25(12):1650152.

35. Itoko T, Raymond R, Imamichi T, Matsuo A. Optimization of quantum circuit mapping using gate transformation and commutation. Integration. 2020 Jan 1; 70:43-50.

36. Curry M. Symbolic quantum circuit simplification in SymPy.

37. Variational Quantum Classifier - Syed Farhan (born-2learn.github.io)

38. Mandviwalla A, Ohshiro K, Ji B. Implementing Grover's algorithm on the IBM quantum computers. In2018 IEEE International Conference on Big Data (Big Data) 2018 Dec 10 (pp. 2531-2537). IEEE.

39. Karalekas PJ, Tezak NA, Peterson EC, Ryan CA, da Silva MP, Smith RS. A quantum-classical cloud platform optimized for variational hybrid algorithms. Quantum Science and Technology. 2020 Mar 26;5(2):024003.

40. Larose R. Overview and comparison of gate level quantum software platforms. Quantum. 2019 Mar 25; 3:130.

41. https://cds.cern.ch/record/2716204/plots.

42. LaBorde ML, Rogers AC, Dowling JP. Finding broken gates in quantum circuits: exploiting hybrid machine learning. Quantum Information Processing. 2020 Aug;19(8):1-8.

43. McKay, David C., et al. "Qiskit backend specifications for openqasm and openpulse experiments." arXiv preprint arXiv:1809.03452 (2018).

44. Abbaszade M, Salari V, Mousavi SS, Zomorodi M, Zhou X. Application of quantum natural language processing for language translation. IEEE Access. 2021 Aug 30;9:130434-48.

45. Salari V., et. al. Quantum Face Recognition Protocol with Ghost Imaging. preprint: arXiv:2110.10088770 [quant-ph]