

## Research Article

# Quantum-Based Feature Selection for Multiclassification Problem in Complex Systems with Edge Computing

Wenjie Liu <sup>1</sup>, Junxiu Chen,<sup>2</sup> Yuxiang Wang,<sup>2</sup> Peipei Gao,<sup>2</sup> Zhibin Lei,<sup>3</sup> and Xu Ma <sup>4</sup>

<sup>1</sup>Engineering Research Center of Digital Forensics, Ministry of Education, School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing 210044, China

<sup>2</sup>School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing 210044, China

<sup>3</sup>Hong Kong Applied Science and Technology Research Institute (ASTRI), Hong Kong 999077, China

<sup>4</sup>School of Software, Qufu Normal University, Shandong 273165, China

Correspondence should be addressed to Xu Ma; [xma@xidian.edu.cn](mailto:xma@xidian.edu.cn)

Received 3 December 2019; Revised 2 February 2020; Accepted 15 February 2020; Published 31 March 2020

Guest Editor: Xuyun Zhang

Copyright © 2020 Wenjie Liu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The complex systems with edge computing require a huge amount of multifeature data to extract appropriate insights for their decision making, so it is important to find a feasible feature selection method to improve the computational efficiency and save the resource consumption. In this paper, a quantum-based feature selection algorithm for the multiclassification problem, namely, QReliefF, is proposed, which can effectively reduce the complexity of algorithm and improve its computational efficiency. First, all features of each sample are encoded into a quantum state by performing operations  $CMP$  and  $R_y$ , and then the amplitude estimation is applied to calculate the similarity between any two quantum states (i.e., two samples). According to the similarities, the Grover–Long method is utilized to find the nearest  $k$  neighbor samples, and then the weight vector is updated. After a certain number of iterations through the above process, the desired features can be selected with regards to the final weight vector and the threshold  $\tau$ . Compared with the classical ReliefF algorithm, our algorithm reduces the complexity of similarity calculation from  $O(MN)$  to  $O(M)$ , the complexity of finding the nearest neighbor from  $O(M)$  to  $O(\sqrt{M})$ , and resource consumption from  $O(MN)$  to  $O(M \log N)$ . Meanwhile, compared with the quantum Relief algorithm, our algorithm is superior in finding the nearest neighbor, reducing the complexity from  $O(M)$  to  $O(\sqrt{M})$ . Finally, in order to verify the feasibility of our algorithm, a simulation experiment based on Rigitte with a simple example is performed.

## 1. Introduction

Complex systems [1] are nonlinear systems composed of agents that can act with local environmental information, which require big data to extract appropriate insights for their decision making. In the cloud computing [2–5], the data transmission delay between the data sources and the cloud centers is problematic for many complex systems where responses are usually required to be time critical or real time. Instead, a recently emerging computation paradigm, edge computing [6–9], is promising to cater for these requirements, as edge computing resources are deployed data sources which support time critical or real-time data processing and analysis. As we all know, the computing resources and storage

resources of most intelligent terminals are very limited, which places higher requirements on the computing performance of algorithms, especially machine learning algorithms, in complex systems with edge computing.

Machine learning [10, 11] continuously improves its performance through “experience,” where experience generally originates from massive data. At present, many machine learning algorithms based on massive data [12–14] have been proposed. In the practical scenario, the amount of data available for training is getting larger and larger, while the characteristics of data are becoming more and more abundant. Those data with redundant or unrelated features will cause the problem of “curse of dimensionality” [15], which greatly increases the computational complexity of the

algorithm. One of the possible solutions is the dimension reduction [16], and the other is the feature selection [17].

Relief algorithm [18] is a well-known feature selection algorithm for the two-classification problem. It is widely used because of its excellent classification effect. However, the limitation of this algorithm is that it can only perform the binary classification, and the efficiency of the algorithm will be greatly affected when the data size and feature size increase. To extend the application of the algorithm, Kononenko [19] proposed a new feature selection algorithm for the multiclassification problem, namely, ReliefF algorithm. It has the advantages of simple principle, convenient implementation, and good results and has been widely applied in various fields [20–22].

On the other hand, since Benioff [23] and Feynman [24] explored the theoretical possibilities of quantum computing, some excellent results have been proposed one after another. For instance, Shor’s algorithm [25] solves the problem of integer factorization in polynomial time. Grover’s algorithm [26] has a quadratic speedup to the problem of conducting a search through some unstructured database. These excellent results have prompted people to think about how to apply this computing power into machine learning algorithms. Thus, a new research hotspot, quantum machine learning [27–32], has gradually formed. Although quantum technology provides a certain improvement in storage and computing power, the “curse of dimensionality” problem still exists in quantum machine learning. Therefore, the quantum-based dimensionality reduction method still has important research value. In 2018, Liu et al. [33] proposed a quantum Relief algorithm (namely, QRelief algorithm) for the two-classification problem, which reduces the complexity of similarity calculation from  $O(MN)$  to  $O(M)$ .

As we know, in the application scenario of edge computing, there are various multiclassification problems based on distributed, massive, and large-feature data. The objective of this study is to design a feasible feature selection method which can effectively get rid of redundant or unrelated features in machine learning, reducing the computation load of intelligent terminals, and thus meet the requirement of real-time data processing and analysis in edge computing. In this paper, we introduce some quantum technologies (such as CMP operation, amplitude estimation, and Grover–Long method) and propose a quantum-based feature selection algorithm, namely QReliefF algorithm, for the multiclassification problem.

The main contributions of our work are as follows:

- (1) A quantum method is proposed to solve the problem of feature selection for the multiclassification problem in complex systems with edge computing. The proposed method fully demonstrates the quantum parallel processing capabilities that classical computing cannot match and significantly reduces the computational complexity of the algorithm.
- (2) The problem of finding nearest neighbor samples is firstly transformed into the similarity calculation of two quantum states (i.e., calculating their inner

product) in quantum computing, and the Grover–Long method is utilized to speed up the search of the targets.

- (3) A simulation experiment based on Rigetti is performed to verify the feasibility of our algorithm.

The outline of this paper is as follows. The classic ReliefF algorithm is briefly reviewed in Section 2, and the proposed quantum ReliefF algorithm is proposed in detail in Section 3. Then, we illustrate the process of the algorithm with a simple example in Section 4 and perform the simulation experiment on Rigetti in Section 5. Subsequently, the efficiency of the algorithm is analyzed in Section 6, and the brief conclusion and discussion are summarized in the last section.

## 2. Review of ReliefF Algorithm

ReliefF algorithm [19] is a feature selection algorithm which is used to handle the multiclassification problem. Before introducing our proposed quantum algorithm, let us review the detailed process of the algorithm.

Without loss of generality, suppose there are  $M$  samples with  $N$  features, and they can be divided into  $P$  classes:

$$C_p = \{v_q \mid v_q \in \mathbb{R}^N, q = 1, 2, \dots, M_p\}, \quad p \in \{1, 2, \dots, P\}, \quad (1)$$

where  $v_q$  is the  $q$ -th  $N$ -feature sample that belongs to Class  $C_p$ ,  $v_q = (v_{q1}, v_{q2}, \dots, v_{qN})^T$ . And the weight vector of  $N$  features  $WT = (wt_1, wt_2, \dots, wt_N)^T$  is initialized to all zeros, the upper limit of iteration is  $T$ , and the relevance threshold (that differentiates the relevant and irrelevant features) is  $\tau$  ( $0 \leq \tau \leq 1$ ). The main steps of ReliefF algorithm are as follows (its pseudocode can be seen in Algorithm 1).

At each iteration, ReliefF randomly selects a sample  $u$  and then searches for  $k$  nearest neighbor samples by cosine distance from each class. The closest same-class sample is called  $H_j$ , and the closest different-class sample is called  $M_j(C_p)$ , where  $j = \{1, 2, \dots, k\}$ . The updating weight vector formula is shown as follows:

$$WT[i] = WT[i] - \sum_{j=1}^k \text{diff}(i, u, H_j) + \sum_{C_p \notin \text{class}(u)} \left[ \frac{p(C_p)}{1 - p(\text{class}(u))} \sum_{j=1}^k \text{diff}(i, u, M_j(C_p)) \right], \quad (2)$$

where  $p(C_p)$  represents the probability of randomly extracting samples of Class  $C_p$ , and the definition of  $\text{diff}(i, u, v)$  function is as follows:

$$\text{diff}(i, u, v) = \begin{cases} \frac{|u[i] - v[i]|}{\max(i) - \min(i)}, & i \text{ is continuous,} \\ 0, & u_i = v_i, \\ 1, & u_i \neq v_i. \end{cases} \quad (3)$$

```

(1) Init WT = (0, . . . , 0)T
(2) for t = 1 to T do
(3)   Pick a sample u randomly
(4)   Find k nearest neighbor samples Hj from the same class of sample u
(5)   for Cp ≠ class(u) do
(6)     Find k nearest neighbor samples Mj(Cp) from the different Class Cp
(7)   end
(8)   for i = 1 to N do
(9)     WT[i] = WT[i] - ∑j=1k diff(i, u, Hj) + ∑Cp≠class(u) [(p(Cp)/(1 - p(class(u)))) ∑j=1k diff(i, u, Mj(Cp))]
(10)  end
(11) end
(12) Select the most relevant features according to WT and τ

```

ALGORITHM 1: ReliefF algorithm.

After iterating  $T$  times, the final weight vector is obtained. Through the relevance threshold  $\tau$ , we can retain relevant features and discard irrelevant features.

ReliefF algorithm is an extension of Relief algorithm that extends the two-classification problem to multiclassification scenario. However, with the increase of category size, sample size, and sample features, ReliefF algorithm will also face with the problem of “dimension disaster,” and the speed of the algorithm will also drop sharply. So, how to improve the efficiency of ReliefF algorithm becomes an urgent problem to be solved.

### 3. The Proposed QReliefF Algorithm

In order to implement the feature selection for the multiclassification problem in complex systems with edge computing, a feasible quantum ReliefF algorithm is introduced in this section. Suppose the sample sets  $C_p = \{v_q = (v_{q1}, v_{q2}, \dots, v_{qN})^T \mid v_q \in \mathbb{R}^N, q = 1, 2, \dots, M_p\}$  ( $p$  represents the category of classification,  $p \in \{1, 2, \dots, P\}$ ), the weight vector WT, the upper limit  $T$ , and the relevance threshold  $\tau$  are the same as classical ReliefF algorithm defined in Section 2. Different from the classical one, all the features of each sample are represented as a quantum superposition state, and thus the problem of finding nearest neighbor samples is transformed into the similarity calculation of two quantum states (i.e., calculating their inner product). And the similarity between any two samples can be calculated in parallel in the way of quantum mechanics. Algorithm 2 describes the process of our algorithm in detail, and the specific steps are as follows.

**3.1. State Preparation.** In order to store classical information in quantum states, we need to normalize the sample sets:

$$C_p \longrightarrow \bar{C}_p = \left\{ \bar{v}_q = (\bar{v}_{q1}, \bar{v}_{q2}, \dots, \bar{v}_{qN})^T \right\}, \quad (4)$$

where

$$\bar{v}_{qi} = \frac{v_{qi}}{\sqrt{\sum_{i=1}^N |v_{qi}|^2}}, \quad i = 1, 2, \dots, N. \quad (5)$$

Obviously,  $\bar{v}_{qi}$  is a real number, and  $\bar{v}_{qi} \in (0, 1)$ . Then, we prepare the initial quantum states as below:

$$|\phi_p\rangle_q = \frac{1}{\sqrt{N}} |q\rangle \sum_{i=0}^{N-1} |i\rangle |1\rangle \left( \sqrt{1 - |\bar{v}_{qi}|^2} |0\rangle + \bar{v}_{qi} |1\rangle \right), \quad (6)$$

where  $|\phi_p\rangle_q$  corresponds to the quantum state of the  $q$ -th sample that belongs to Class  $\bar{C}_p$  and  $\bar{v}_{qi}$  represents the  $i$ -th eigenvalue of the  $q$ -th sample. Assume our initial state is  $|q\rangle |0\rangle^{\otimes n} |1\rangle |0\rangle$  ( $n = \lceil \log_2(N) \rceil$ ); the construction scheme of the quantum state  $|\phi_p\rangle_q$  includes the following steps.

First, we perform Hadamard and CMP operations for  $|0\rangle^{\otimes n}$  and get a new state:

$$|0\rangle^{\otimes n} \xrightarrow{\text{Hadamard and CMP operations}} \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} |i\rangle, \quad (7)$$

and its circuit diagram is shown in Figure 1, where the definition of CMP operation is

$$\text{CMP}|i\rangle|N\rangle|0\rangle = \begin{cases} |i\rangle|N\rangle|0\rangle, & i < N, \\ |i\rangle|N\rangle|1\rangle, & i \geq N. \end{cases} \quad (8)$$

The function of CMP operation is to cut the quantum state larger than  $N$ , and its circuit diagram is shown in Figure 2.  $|i\rangle$  and  $|N\rangle$  represent a single qubit. The implementation of CMP operation needs to repeatedly implement such a circuit  $n$  times. After measurement, if the lowest register is  $|1\rangle$ , it means that  $i > N$ .

Next, we perform the unitary rotation operation  $R_y$ :

$$R_y(2 \sin^{-1} \bar{v}_{qi}) = \begin{bmatrix} \sqrt{1 - |\bar{v}_{qi}|^2} & -\bar{v}_{qi} \\ \bar{v}_{qi} & \sqrt{1 - |\bar{v}_{qi}|^2} \end{bmatrix}, \quad (9)$$

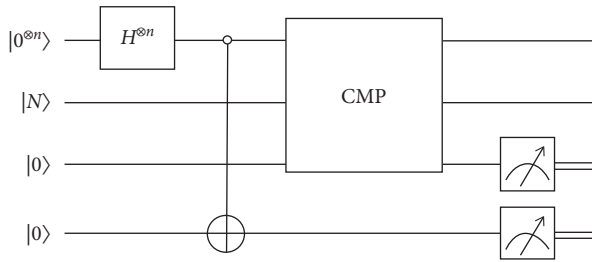
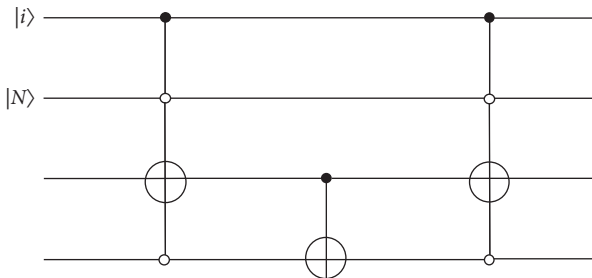
on the last qubit to obtain our target quantum state  $|\phi_p\rangle_q$ :

$$\frac{1}{\sqrt{N}} |q\rangle \sum_{i=0}^{N-1} |i\rangle |1\rangle \left( \sqrt{1 - |\bar{v}_{qi}|^2} |0\rangle + \bar{v}_{qi} |1\rangle \right). \quad (10)$$

**3.2. Similarity Calculation.** After the state preparation, the information of the samples is encoded into the quantum superposition state  $\{|\phi_p\rangle_q\}$ . In this paper, we use the cosine distance to define the similarity between the random sample  $\bar{u}$  and other sample (e.g.,  $\bar{v}_q$ ):

- (1) Init  $WT = (0, \dots, 0)^T$
- (2) Normalized the sample sets:  $C_p \rightarrow \bar{C}_p$
- (3) Prepare quantum states for all samples by operations CMP and  $R_p$ , respectively.  
 $|\phi_p\rangle_q = (1/\sqrt{N})|q\rangle \sum_{i=0}^{N-1} |i\rangle|1\rangle (\sqrt{1-|\bar{v}_q|^2}|0\rangle + \bar{v}_q|1\rangle)$
- (4) **for**  $t = 1$  **to**  $T$  **do**
- (5) Select a state  $|\phi\rangle$  from  $\{|\phi_p\rangle_q\}$  randomly which corresponds to  $u$
- (6) Perform swap operation on  $|\phi\rangle$  and obtain  $|\varphi\rangle = (1/\sqrt{N})|l\rangle \sum_{i=0}^{N-1} |i\rangle (\sqrt{1-|\bar{u}_i|^2}|0\rangle + \bar{u}_i|1\rangle)|1\rangle$
- (7) The similarity information coded into quantum state  $|\beta\rangle_p = 1/\sqrt{M_p} \sum_{q=1}^{M_p} |q\rangle |\bar{v}_q - \bar{u}\rangle$  through swap test, the inner product and amplitude estimation operations
- (8) The nearest  $k$  samples in each class are obtained by Grover-Long method
- (9) **for**  $i = 1$  **to**  $N$  **do**
- (10)  $WT[i] = WT[i] - \sum_{j=1}^k \text{diff}(i, \bar{u}, H_j) + \sum_{\bar{C}_p \notin \text{class}(\bar{u})} [(p(\bar{C}_p))/(1 - p(\text{class}(\bar{u})))] \sum_{j=1}^k \text{diff}(i, \bar{u}, M_j(\bar{C}_p))$
- (11) **end**
- (12) **end**
- (13)  $\bar{WT} = (1/T)WT$
- (14) **for**  $i = 1$  **to**  $N$  **do**
- (15) **if**  $(\bar{WT}_i \geq \tau)$  **then**
- (16) The  $i$ -th feature is relevant
- (17) **else**
- (18) The  $i$ -th feature is not relevant
- (19) **end**
- (20) **end**

ALGORITHM 2: Quantum ReliefF algorithm.

FIGURE 1: Quantum circuit of getting  $(1/\sqrt{N})\sum_{i=0}^{N-1}|i\rangle$ ;  $H$  is the Hadamard operation and  $\circ$  represents the control qubit conditional being set to zero.FIGURE 2: Quantum circuit of CMP operation;  $\bullet$  represents the control qubit conditional being set to one.

$$s(\bar{u}, \bar{v}_q) \doteq |\cos \theta|^2 = \frac{|\langle \bar{u} | \bar{v}_q \rangle|^2}{|\bar{u}|^2 \cdot |\bar{v}_q|^2}. \quad (11)$$

Referring to equations (4) and (5),  $|\bar{u}|^2$  and  $|\bar{v}_q|^2$  are 1, and equation (11) can be simplified as follows:

$$s(\bar{u}, \bar{v}_q) = |\langle \bar{u} | \bar{v}_q \rangle|^2. \quad (12)$$

First,  $|\phi\rangle$  (i.e., the sample  $\bar{u}$ ) is randomly selected from  $\{|\phi_p\rangle_q\}$  which is the  $l$ -th sample in Class  $\bar{C}_p$ , as shown in the following equation:

$$|\phi\rangle = \frac{1}{\sqrt{N}} |l\rangle \sum_{i=0}^{N-1} |i\rangle|1\rangle (\sqrt{1-|\bar{u}_i|^2}|0\rangle + \bar{u}_i|1\rangle). \quad (13)$$

Then, a swap operation is performed on  $|\phi\rangle$  to get

$$|\varphi\rangle = \frac{1}{\sqrt{N}} |l\rangle \sum_{i=0}^{N-1} |i\rangle (\sqrt{1-|\bar{u}_i|^2}|0\rangle + \bar{u}_i|1\rangle)|1\rangle. \quad (14)$$

Next, a swap test (its circuit is given in Figure 3) is performed on  $(|\varphi\rangle, |\phi_p\rangle_q)$ , and we obtain

$$|\psi\rangle = \frac{1}{2} |0\rangle (|\varphi\rangle |\phi_p\rangle_q + |\phi_p\rangle_q |\varphi\rangle) + \frac{1}{2} |1\rangle (|\varphi\rangle |\phi_p\rangle_q - |\phi_p\rangle_q |\varphi\rangle). \quad (15)$$

From equation (15), we know the probability of measurement result being  $|1\rangle$  is

$$\begin{aligned} P_q^1(1) &= \langle \psi | 1 \rangle \langle 1 | \otimes I \otimes I | \psi \rangle \\ &= \left[ \frac{1}{2} \langle 1 | \left( \langle \varphi | \langle \phi_p |_q - \langle \phi_p |_q \langle \varphi | \right) | 1 \rangle \langle 1 | \otimes I \otimes I \right. \\ &= \left[ \frac{1}{2} \langle 1 | \left( |\varphi\rangle |\phi_p\rangle_q - |\phi_p\rangle_q |\varphi\rangle \right) \right. \\ &= \frac{1}{2} - \frac{1}{2} |\langle \varphi | \phi_p\rangle_q|^2. \end{aligned} \quad (16)$$

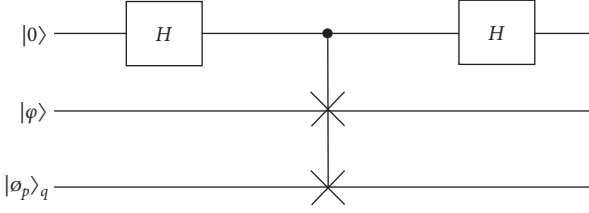


FIGURE 3: Quantum circuit of *swap test* operation; the symbol of two crosses connected by a line represents the *swap* operation.

In addition, the inner product between  $|\varphi\rangle$  and  $|\phi_p\rangle_q$  (i.e., the prepared state) can be calculated as follows:

$$\langle\varphi|\phi_p\rangle_q = \frac{1}{N} \sum_i (\bar{u}_i) * \bar{v}_{qi} = \frac{1}{N} \langle\bar{u}|\bar{v}_q\rangle. \quad (17)$$

Combining equation (16) with equation (17), we can get the similarity between samples  $\bar{u}$  and  $\bar{v}_q$ :

$$s(\bar{u}, \bar{v}_q) = \left(1 - 2P_q^{l(\bar{C}_p)}(1)\right) N^2. \quad (18)$$

Since  $N$  is a constant value and  $\langle\bar{u}|\bar{v}_q\rangle$  is the angle cosine between the random sample  $\bar{u}$  and other sample  $\bar{v}_q$  (e.g., in Class  $\bar{C}_p$ ), then the smaller  $s(\bar{u}, \bar{v}_q)$  is, the smaller cosine distance is, which indicates that these two samples are more similar.

Then, we can rewrite equation (15) as follows:

$$|\alpha\rangle_p = \frac{1}{\sqrt{M_p}} \sum_{q=1}^{M_p} |q\rangle \left( \sqrt{1 - s(\bar{u}, \bar{v}_q)} |0\rangle + \sqrt{s(\bar{u}, \bar{v}_q)} |1\rangle \right). \quad (19)$$

**3.3. Finding the Nearest Neighbor Samples.** First, the quantum amplitude estimation method [34] is applied to store the similarity of the sample  $\bar{u}$  and  $\bar{v}_q$  in the last qubit:

$$|\beta\rangle_p = \frac{1}{\sqrt{M_p}} \sum_{q=1}^{M_p} |q\rangle |s(\bar{u}, \bar{v}_q)\rangle, \quad (20)$$

where  $p \in \{1, 2, \dots, P\}$ , and its quantum circuit diagram is given in Figure 4.

In the Grover-Long method [35], one iteration can be divided into four operations, i.e.,  $G = -WI_0W^{-1}O$ , and its quantum circuit is shown in Figure 5.  $O$  is an oracle operation which performs a phase inversion on the targets:

$$O = e^{i\phi} |v\rangle\langle v| + \sum_{\tau=0, \tau \neq v}^{2^n-1} |\tau\rangle\langle\tau|, \quad (21)$$

where  $v$  is the position of  $e^{i\phi}$  in the diagonal matrix. The position  $v$  of  $e^{i\phi}$  is divided into two cases. If  $v$  is odd, the  $u_1(\phi)$  operation will be applied to the lowest qubit:

$$u_1(\phi) = \text{diag}[1, e^{i\phi}]. \quad (22)$$

If  $v$  is even,  $X$ ,  $u_1(\phi)$ ,  $X$  operations will be applied to the lowest qubit.

Besides,  $I_0$  is a conditional phase shift operation which performs a phase inversion on  $|0\rangle$ :

$$I_0 = e^{i\phi} |0\rangle\langle 0| + \sum_{\tau=1}^{2^n-1} |\tau\rangle\langle\tau| = \text{diag}[e^{i\phi}, 1, \dots, 1]_{2^n}, \quad (23)$$

$$\phi = 2\arcsin\left(\frac{\sin(\pi/(4J+2))}{\sin\eta}\right),$$

where  $\sin\eta = \sqrt{(M/N)}$  and  $J$  represents the number of iteration.

Having obtained the state  $|\beta\rangle_p$  (see equation (20)) through the amplitude estimation, we introduce a quantum minimum search algorithm [37] to find  $k$  nearest neighbor samples from Class  $\bar{C}_p$  with the time complexity of  $O(\sqrt{kM_p})$ , and its quantum circuit is shown in Figure 6.

Suppose the set  $K = \{K_1, K_2, \dots, K_k\}$  represents the  $k$  nearest neighbor samples, we should prepare  $\lceil\sqrt{k}\rceil$  auxiliary qubits. As shown in Figure 6, the operator  $W_s$  represents  $W_s|\beta\rangle_p|0^{\otimes k}\rangle = |\beta\rangle_p|K_1\rangle|K_2\rangle \dots |K_k\rangle$ , and  $u_1(\phi)$  is the operator defined in equation (22). Let  $d_0 = s(\bar{u}, \bar{v}_1)$ ; we can mark  $K_x$  when  $d_0 > s(\bar{u}, \bar{v}_{K_x})$ ,  $x \in [1, k]$ . Next, we can replace  $d_0$  after one iteration, where  $d_0$  is  $\min\{s(\bar{u}, \bar{v}_{K_x})\}$ ,  $x \in [1, k]$ . We repeat the above steps several times until all samples in Class  $\bar{C}_p$  are compared. Finally, all index of  $k$  nearest neighbor samples in Class  $\bar{C}_p$  can be obtained according to the similarity.

**3.4. Updating Weight Vector.** After the above steps, we obtain the nearest neighbor samples (i.e.,  $H_j$  and  $M_j(\bar{C}_p)$ ) of the random sample  $\bar{u}$ . Then, we update the weight vector according to the updating weight vector formula as follows:

$$\begin{aligned} \text{WT}[i] = & \text{WT}[i] - \sum_{j=1}^k \text{diff}(i, \bar{u}, H_j) \\ & + \sum_{\bar{C}_p \notin \text{class}(\bar{u})} \left[ \frac{P(\bar{C}_p)}{1 - P(\text{class}(\bar{u}))} \sum_{j=1}^k \text{diff}(i, \bar{u}, M_j(\bar{C}_p)) \right], \end{aligned} \quad (24)$$

where  $i \in [1, N]$ .

**3.5. Feature Selection.** After iterating the above steps, i.e., similarity calculation, we find the nearest neighbor samples and update weight vector  $T$  times, and we jump out of the algorithm's loop. Then, we get a final weight vector  $\text{WT}$ . And the average weight vector is

$$\overline{\text{WT}} = \frac{1}{T} \text{WT}. \quad (25)$$

Then, we make feature selection based on the final  $\overline{\text{WT}}$  and threshold  $\tau$ . Here,  $\tau$  can be chosen to retain relevant features and discard irrelevant features [38], that is to say, those features whose weight is greater than  $\tau$  will be selected, and those less than  $\tau$  will be discarded. Here, the value of  $\tau$  is determined with regards to the user's requirements and the



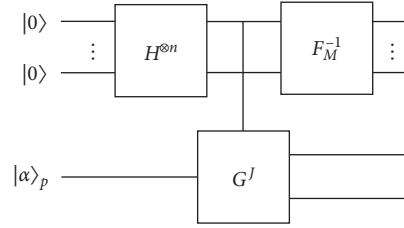


FIGURE 4: Quantum circuit of amplitude estimation operation;  $G^J$  represents  $J$  iterations of Grover-Long method [35] and  $F_M^{-1}$  represents the inverse Fourier transform [36].

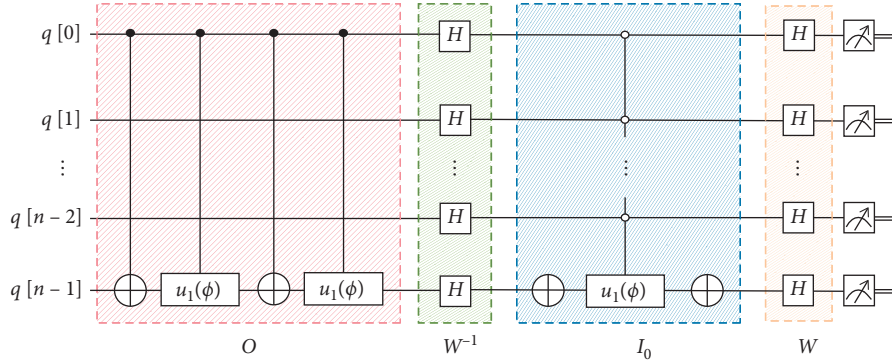


FIGURE 5: Quantum circuit with one iteration in Grover-Long method [35];  $q[0]$  denotes the highest qubit and  $q[n-1]$  denotes the lowest qubit.

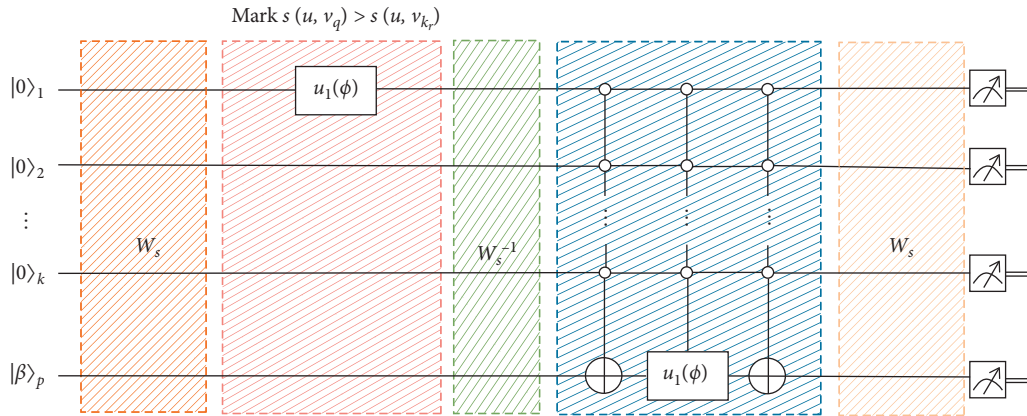


FIGURE 6: Quantum circuit of finding  $k$  nearest neighbor samples.

characteristics of the problem itself (e.g., the distribution of samples and the number of features).

#### 4. Example

Suppose that there are four samples (see Table 1);  $S_0 = (1, 0, 0, 1, 0, 0)$ ,  $S_1 = (1, 0, 0, 0, 1, 0)$ ,  $S_2 = (0, 1, 0, 0, 0, 1)$ ,  $S_3 = (0, 1, 0, 1, 0, 0)$ ,  $S_4 = (0, 0, 1, 0, 1, 0)$ , and  $S_5 = (0, 0, 1, 0, 0, 1)$ , and thus  $n$  is 3, and they belong to two classes:  $A = \{S_0, S_1\}$ ,  $B = \{S_2, S_3\}$ , and  $C = \{S_4, S_5\}$ , which is illustrated in Figure 7.

First, the four initial quantum states are prepared as follows:

$$\begin{cases} |\psi\rangle_{S_0} = |000\rangle|0\rangle^{\otimes 3}|1\rangle|0\rangle, \\ |\psi\rangle_{S_1} = |001\rangle|0\rangle^{\otimes 3}|1\rangle|0\rangle, \\ |\psi\rangle_{S_2} = |010\rangle|0\rangle^{\otimes 3}|1\rangle|0\rangle, \\ |\psi\rangle_{S_3} = |011\rangle|0\rangle^{\otimes 3}|1\rangle|0\rangle, \\ |\psi\rangle_{S_4} = |100\rangle|0\rangle^{\otimes 3}|1\rangle|0\rangle, \\ |\psi\rangle_{S_5} = |100\rangle|0\rangle^{\otimes 3}|1\rangle|0\rangle. \end{cases} \quad (26)$$

Next, we take  $|\psi\rangle_{S_0}$  as an example, and then the  $H^{\otimes 3}$  operation is applied on the third and fourth qubits:

TABLE 1: The feature values of four samples.

	$F_0$	$F_1$	$F_2$	$F_3$	$F_4$	$F_5$
$S_0$	1	0	1	0	0	0
$S_1$	1	0	0	0	1	0
$S_2$	0	1	0	0	0	1
$S_3$	0	1	0	1	0	0
$S_4$	0	0	1	0	1	0
$S_5$	0	0	1	0	0	1

Each row represents all the feature values of a certain sample, while each column denotes a certain feature value of all the samples.

$$|000\rangle|0\rangle^{\otimes 3}|1\rangle|0\rangle \xrightarrow{H^{\otimes 3}} \frac{1}{2}|000\rangle \sum_{i=0}^3 |i\rangle|1\rangle|0\rangle. \quad (27)$$

Then, we perform  $R_y$  rotation (see equation (9)) on the last qubit, and we can get

$$|\phi\rangle_{S_0} = \frac{1}{2}|000\rangle \sum_{i=0}^3 |i\rangle|1\rangle \left( \sqrt{1-|\bar{v}_{0i}|^2} |0\rangle + \bar{v}_{0i}|1\rangle \right). \quad (28)$$

The other quantum states are prepared in the same way and they are listed as follows:

$$\left\{ \begin{array}{l} |\phi\rangle_{S_0} = \frac{1}{2}|000\rangle \sum_{i=0}^3 |i\rangle|1\rangle \left( \sqrt{1-|\bar{v}_{0i}|^2} |0\rangle + \bar{v}_{0i}|1\rangle \right), \\ |\phi\rangle_{S_1} = \frac{1}{2}|001\rangle \sum_{i=0}^3 |i\rangle|1\rangle \left( \sqrt{1-|\bar{v}_{1i}|^2} |0\rangle + \bar{v}_{1i}|1\rangle \right), \\ |\phi\rangle_{S_2} = \frac{1}{2}|010\rangle \sum_{i=0}^3 |i\rangle|1\rangle \left( \sqrt{1-|\bar{v}_{2i}|^2} |0\rangle + \bar{v}_{2i}|1\rangle \right), \\ |\phi\rangle_{S_3} = \frac{1}{2}|011\rangle \sum_{i=0}^3 |i\rangle|1\rangle \left( \sqrt{1-|\bar{v}_{3i}|^2} |0\rangle + \bar{v}_{3i}|1\rangle \right), \\ |\phi\rangle_{S_4} = \frac{1}{2}|100\rangle \sum_{i=0}^3 |i\rangle|1\rangle \left( \sqrt{1-|\bar{v}_{4i}|^2} |0\rangle + \bar{v}_{4i}|1\rangle \right), \\ |\phi\rangle_{S_5} = \frac{1}{2}|100\rangle 1 \sum_{i=0}^3 |i\rangle|1\rangle \left( \sqrt{1-|\bar{v}_{5i}|^2} |0\rangle + \bar{v}_{5i}|1\rangle \right). \end{array} \right. \quad (29)$$

Second, we randomly select a sample (assume  $|\phi\rangle_{S_0}$  is  $\bar{u}$ ) and perform similarity calculation with other samples (i.e.,  $|\phi\rangle_{S_1}$ ,  $|\phi\rangle_{S_2}$ ,  $|\phi\rangle_{S_3}$ ,  $|\phi\rangle_{S_4}$ ,  $|\phi\rangle_{S_5}$ ). Next, we take  $|\phi\rangle_{S_0}$  and  $|\phi\rangle_{S_1}$  as an example and perform a *swap* operation between the last two qubits of  $|\phi\rangle_{S_0}$ :

$$|\phi\rangle_{S_0} \xrightarrow{\text{swap}} |\varphi\rangle = \frac{1}{2}|000\rangle \sum_{i=0}^3 |i\rangle \left( \sqrt{1-|\bar{v}_{0i}|^2} |0\rangle + \bar{v}_{0i}|1\rangle \right) |1\rangle. \quad (30)$$

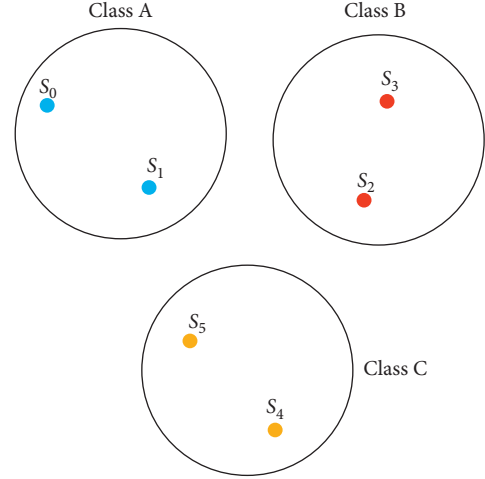


FIGURE 7: The simple example with six samples in classes A, B, and C.

After that, the *swap test* operation is applied on  $(|\varphi\rangle, |\phi\rangle_{S_1})$ :

$$\frac{1}{2}|0\rangle \left( |\varphi\rangle |\phi\rangle_{S_1} + |\phi\rangle_{S_1} |\varphi\rangle \right) + \frac{1}{2}|1\rangle \left( |\varphi\rangle |\phi\rangle_{S_1} - |\phi\rangle_{S_1} |\varphi\rangle \right). \quad (31)$$

We perform a *swap test* operation to obtain a quantum state that encodes similarity in amplitude:

$$\left\{ \begin{array}{l} |\alpha\rangle_A = \frac{1}{\sqrt{2}} \sum_{q=1}^2 |q\rangle \left( \sqrt{1-s(\bar{u}, \bar{v}_q)} |0\rangle + \sqrt{s(\bar{u}, \bar{v}_q)} |1\rangle \right), \\ |\alpha\rangle_B = \frac{1}{\sqrt{2}} \sum_{q=1}^2 |q\rangle \left( \sqrt{1-s(\bar{u}, \bar{v}_q)} |0\rangle + \sqrt{s(\bar{u}, \bar{v}_q)} |1\rangle \right), \\ |\alpha\rangle_C = \frac{1}{\sqrt{2}} \sum_{q=1}^2 |q\rangle \left( \sqrt{1-s(\bar{u}, \bar{v}_q)} |0\rangle + \sqrt{s(\bar{u}, \bar{v}_q)} |1\rangle \right). \end{array} \right. \quad (32)$$

Then, through the amplitude estimation, we can obtain the quantum states:

$$\left\{ \begin{array}{l} |\beta\rangle_A = \frac{1}{\sqrt{2}} \sum_{q=1}^2 |q\rangle |\bar{v}_q - \bar{u}\rangle, \\ |\beta\rangle_B = \frac{1}{\sqrt{2}} \sum_{q=1}^2 |q\rangle |\bar{v}_q - \bar{u}\rangle, \\ |\beta\rangle_C = \frac{1}{\sqrt{2}} \sum_{q=1}^2 |q\rangle |\bar{v}_q - \bar{u}\rangle. \end{array} \right. \quad (33)$$

Next, we perform an oracle operation on the quantum states obtained in the above steps to obtain the  $k$  nearest neighbor samples.

## 5. Simulation Experiment

Quantum Cloud Services (QCS<sup>TM</sup>) is Rigetti’s quantum-first cloud computing platform. At the end of 2017, a 19-qubit processor named “Acorn” was launched, which can be used in QCS through a quantum programming toolkit named Forest [39]. The chip of “Acorn” is made of 20 superconducting qubits but for some technical reasons, qubit 3 is offline and cannot interact with its neighbors, so it is treated as a 19-qubit device whose coupling map is shown in Figure 8.

In order to obtain the result and also verify our algorithm, we choose Rigetti to perform the quantum processing. However, since the Rigetti platform limits the length of the entire circuit and noise has a great influence on the preparation of quantum states [40], we only show one of the ideal experiment circuits of similarity calculation in QRelief algorithm running on Rigetti platform. We successfully stored the characteristic information in the sample into the amplitude of the quantum state and then extracted the amplitude information into the quantum state through the phase estimation algorithm. Figure 9 gives the schematic diagram of our experimental circuit. The corresponding code of the circuit in Rigetti is shown in Algorithm 3. After running Algorithm 3 8 times, the result can be seen in Figure 10. We can get  $|1\rangle$  with the average probability of 0.435125. Then, we successfully stored the characteristic information in the sample into the amplitude of the quantum state. According to equation (32), we can get  $\sqrt{s(\bar{u}, \bar{v}_q)} \approx \sqrt{0.435}$ , i.e.,  $s(\bar{u}, \bar{v}_q) \approx 0.435$ , and then we extracted the amplitude information into the quantum state through the phase estimation algorithm.

After all the steps have been performed, we obtain the quantum states  $S_1 (H)$ ,  $S_2 (M(B))$ , and  $S_5 (M(C))$  of the nearest neighbor samples of the quantum state  $S_0 (\bar{u})$  in each class of the random sample which can be seen in Figure 11. Then, the weight vectors are updated according to equation (24) and the result of WT is listed in the second row of Table 2 after the first iteration. The algorithm iterates  $T$  times (in our example,  $T=4$ ) as above steps and obtains all the WT results as shown in Table 2. After  $T$ -th iterations,  $WT = [4, 4, 4, -2, 0, -2]$ , and then  $\overline{WT} = [1, 1, 1, -1/2, 0, -1/2]$ . In this paper, the value of  $\tau$  in the example is assumed to be 0.5 according to the updated result of WT in Table 2. Since the threshold  $\tau = 0.5$ , the selected features are  $F_0, F_1$ , and  $F_2$ , i.e., the first, second, and third features. The result of quantum feature selection is consistent with the classical Relief algorithm after being verified by Python.

In the final weight value comparison, considering the large amount of data in the complex system and the corresponding eigenvalues, the calculation amount required for the comparison after the final result is obtained is also large. In order to meet the requirements of big data and result accuracy, we adopted an optimized quantum maximum and minimum value search algorithm [37] when comparing weights in the last step to help us quickly and accurately

select the features we want, so as to better solve the multiclassification problem in complex systems.

In circumstances when we can exactly estimate the ratio of the number of solutions  $M$  and the searched space  $N$ , this algorithm can improve the successful probability close to 100%. Furthermore, it shows an advantage in complexity with large databases and in the operation complexity of constructing oracles.

**5.1. Efficiency Analysis.** In order to evaluate the efficiency of QRelief algorithm, three algorithms (i.e., classical Relief, classical ReliefF, and quantum Relief algorithms) are selected to compare with our algorithm from three indicators: complexity of similarity calculation (CSC), complexity of finding the nearest neighbor (CFNN), and resource consumption (RC).

In the classic Relief algorithm, it takes  $O(N)$  time to calculate the distance between randomly selected samples and any other samples, and then it finds the nearest neighbors related to  $M$ . This process needs to iterate  $T$  times, so CSC is  $O(TMN)$ . Since  $T$  is a constant, CSC in the classic Relief algorithm is  $O(MN)$ . As we know there are totally  $M$  samples, each with  $N$  features, CFNN is  $O(M)$  and RC in the classic Relief algorithm is  $O(MN)$  bits. The classical ReliefF algorithm is similar to the classical Relief algorithm. Since it finds  $k$  nearest neighbors at once time, the time complexity is  $O(kTMN)$ . Then, we can simplify CSC to  $O(MN)$  because  $k$  and  $T$  are constants. Besides, CFNN for finding  $k$  nearest neighbors is  $O(M)$ . In terms of resource consumption, there are  $M$  samples, and each sample has  $N$  features, so the resource consumption of the classic Relief algorithm is  $O(MN)$  bits.

In QRelief and QReliefF algorithms, the quantum property is used to calculate the distance from  $O(N)$  to  $O(1)$ , so their CSCs are all  $O(TM)$ . Since  $T$  is constant, their CSCs can be simplified to  $O(M)$ . CFNN of QRelief is  $O(kM)$ , and then it can be simplified to  $O(M)$  as  $k$  is constant, while CFNN of QReliefF is  $\sqrt{kM}$  because it uses the quantum Grover–Long method to find  $k$  nearest neighbor samples which holds a quadratic acceleration. Since  $k$  is constant, CFNN of QReliefF is  $O(\sqrt{M})$ . RC of similarity calculation, finding the nearest neighbor samples and updating weight vector are  $O(TM \log N)$ ,  $O(TN)$  and  $O(N)$ , respectively. Therefore, the total complexity is  $O(TM \log N + TN + N)$ ; since  $T$  is constant, RC of QRelief and QReliefF is  $O(M \log N + N)$ . For multifeature big data in complex systems with edge computing, there is  $M \gg N$ , so  $M \gg (N/N - \log N)$ , and then RC of QRelief and QReliefF can be simplified into  $O(M \log N)$ .

For convenience, we list the efficiency comparison of classic Relief algorithm, ReliefF algorithm, quantum Relief algorithm, and our algorithm in terms of CSC, CFNN, and RC in Table 3. Obviously, our algorithm is superior to classical algorithms (i.e., Relief and ReliefF) in terms of CSC, CFNN, and RC and better than quantum algorithm (i.e., QRelief) in terms of CFNN.



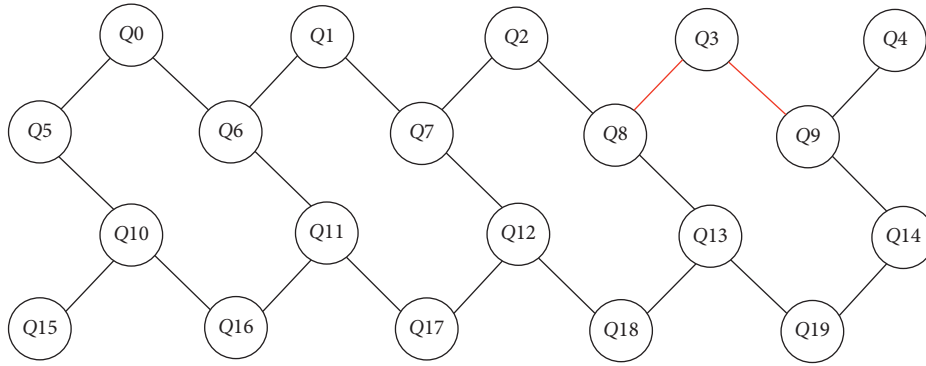


FIGURE 8: The coupling map picture: Rigetti's 19-qubit processor "Acorn." Lines indicate the two-qubit connection ruled by a controlled Z operation.

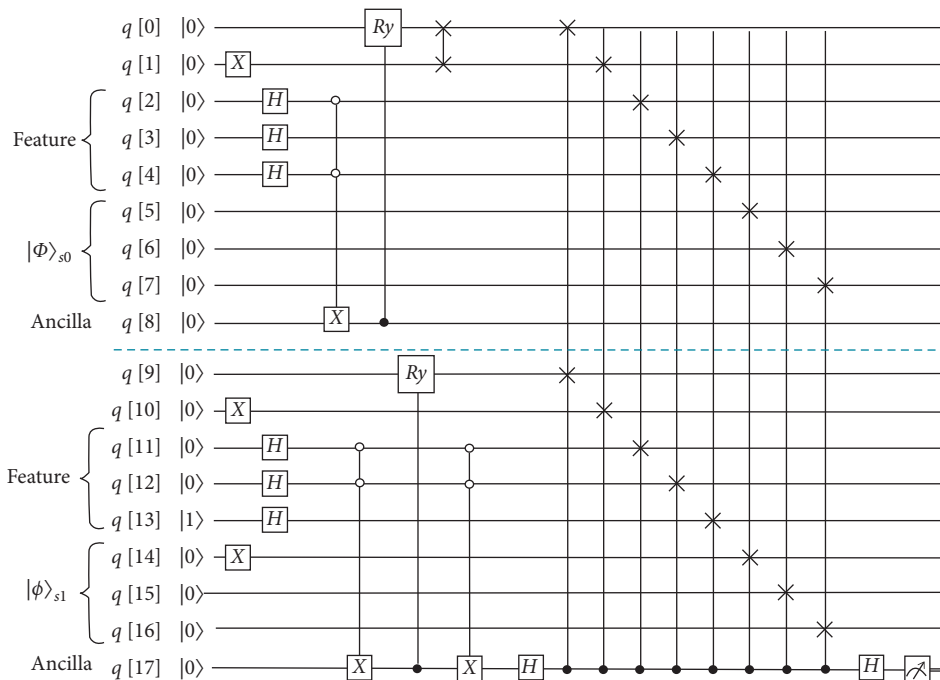


FIGURE 9: One of the ideal experiment circuits of similarity calculation in QReliefF algorithm running on Rigetti platform.  $q[0] - q[7]$  represents the randomly selected quantum state  $|\phi\rangle_{s_0}$ ,  $q[9] - q[16]$  represents  $|\phi\rangle_{s_1}$ , and  $q[17]$  is the resultant qubit. X is the Not operation, and  $R_y$  is  $R_y$  operation which can be expressed as a matrix in equation (9).

```

(1) # Define the new gate from a matrix
(2) theta = Parameter("theta")
(3) cry = np.array([
(4)     [1, 0, 0, 0]
(5)     [0, 1, 0, 0]
(6)     [0, 0, quil_sqrt(1 - theta * theta), -theta * theta]
(7)     [0, 0, theta * theta, quil_sqrt(1 - theta * theta)]
(8) ])
(9) gate_definition = DefGate("CRY", cry, [theta])
(10) CRY = gate_definition.get_constructor()

```

ALGORITHM 3: Continued.

```

(11) # Create our program and use the new parametric gate
(12) p = Program(
(13)   gate_definition, X(1), H(2), H(4), H(5), X(2), X(5)
(14)   CCNOT(2, 5, 18), X(2), X(5), CRY(1) (18, 0)
(15)   SWAP(0, 1), X(10), H(11), H(12), H(13), X(14)
(16)   X(11), X(12), CCNOT(11, 12, 17), X(11), X(12)
(17)   CRY(1) (17, 9), H(19), CSWAP(19, 0, 9)
(18)   CSWAP(19, 1, 10), CSWAP(19, 2, 11)
(19)   CSWAP(19, 4, 12), CSWAP(19, 5, 13)
(20)   CSWAP(19, 6, 14), CSWAP(19, 7, 15)
(21)   CSWAP(19, 8, 16), H(19)
(22) )
(23) # print the circuit
(24) print(p)
(25) # get a QPU, 20q – Acorn is just a string naming the device
(26) qc = get_qc("20q – Acorn")
(27) # run and measure
(28) result = qc.run_and_measure(p, trials = 1024)

```

ALGORITHM 3: Similarity calculation in QReliefF algorithm running on Rigetti.

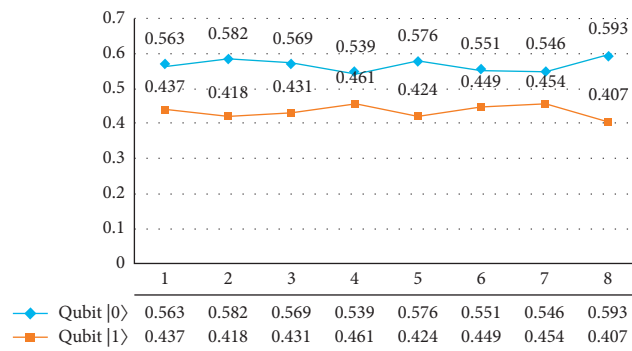


FIGURE 10: The measurement result of  $|0\rangle$  and  $|1\rangle$  after running Algorithm 3 8 times on Rigetti.

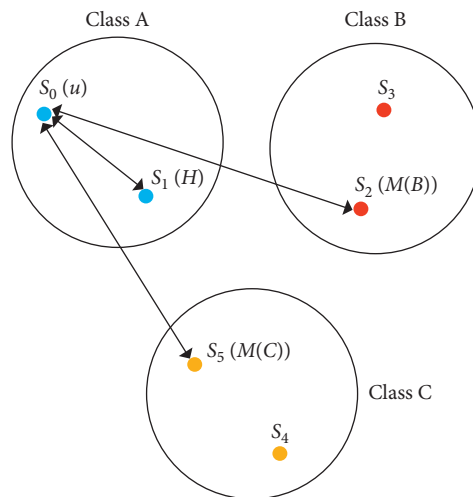


FIGURE 11: Finding the nearest neighbor samples ( $S_1$ ,  $S_2$ , and  $S_5$ ) of the sample  $S_0$ .

TABLE 2: The updated result of WT.

Iteration times ( $T$ )	Weight vector ( $WT$ )
1	[1, 1, 1, 0, 0, -1]
2	[2, 2, 2, -1, 0, -1]
3	[3, 3, 3, -1, 0, -2]
4	[4, 4, 4, -2, 0, -2]

TABLE 3: Efficiency comparison between classical Relief, classical ReliefF, QRelief, and QReliefF algorithms.

	Multiclassification	CSC	CFNN	RC
Relief	No	$O(MN)$	$O(M)$	$O(MN)$
ReliefF	Yes	$O(MN)$	$O(M)$	$O(MN)$
QRelief	No	$O(M)$	$O(M)$	$O(M\log N)$
QReliefF	Yes	$O(M)$	$O(\sqrt{M})$	$O(M\log N)$

## 6. Conclusion and Discussion

With the rapid development of edge computing technology and quantum machine learning algorithms, researchers began to pay attention to the combination and application of these two fields. In this paper, we use quantum technology to solve the multiclassification problem of feature selection in the complex systems with edge computing and propose a quantum ReliefF algorithm. Compared to the classic ReliefF algorithm, our algorithm reduces the complexity of similarity calculation from  $O(MN)$  to  $O(M)$  and the complexity of finding the nearest neighbor from  $O(M)$  to  $O(\sqrt{M})$ . In addition, from the perspective of resource consumption, our algorithm consumes  $O(M\log N)$  qubit, while the classic ReliefF algorithm consumes  $O(MN)$  bit. Obviously, our algorithm is superior in terms of computational complexity and resource consumption.

It should be noted that our work aims to improve the algorithm efficiency, while the privacy protection of sensitive data is not taken into account. At present, data security has become a focus of attention in the field of artificial intelligence, and some solutions for data privacy protection in complex systems with edge computing have been proposed [41–44]. In our future work, how to improve the efficiency of quantum machine learning algorithms while ensuring the privacy protection of sensitive data, such as [45–48], will become our direction.

## Data Availability

The specific data items and corresponding experimental codes used to support the findings of this study are included within the article.

## Disclosure

An earlier version of the manuscript has been presented in “2019 International Conference on IEEE Cyber, Physical and Social Computing (CPSCom).”

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

The support of all the members of the quantum research group of NUIST and SEU is especially acknowledged; their professional discussion and advices have helped us a lot. This study was supported by the Natural Science Foundation of Jiangsu Province (grant no. BK20171458). This study was also supported in part by the Natural Science Foundation of China (grant nos. 61672290 and 61802002), the Natural Science Foundation of Jiangsu Higher Education Institutions of China (grant no. 19KJB520028), and the Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD).

## References

- [1] R. Alfieri and L. Milanese, *Complex System*, Springer, New York, NY, USA, 2013.
- [2] M. Armbrust, I. Stoica, M. Zaharia et al., “A view of cloud computing,” *Communications of the ACM*, vol. 53, no. 4, pp. 50–58, 2010.
- [3] X. Chen, L. Jiao, W. Li, and X. Fu, “Efficient multi-user computation offloading for mobile-edge cloud computing,” *IEEE/ACM Transactions on Networking*, vol. 24, no. 5, pp. 2827–2840, 2016.
- [4] X. Xu, S. Fu, L. Qi et al., “An IoT-oriented data placement method with privacy preservation in cloud environment,” *Journal of Network and Computer Applications*, vol. 124, pp. 148–157, 2018.
- [5] L. Qi, Yi Chen, Y. Yuan et al., *A QoS-Aware Virtual Machine Scheduling Method for Energy Conservation in Cloud-Based Cyber-Physical Systems*, World Wide Web, Geneva, Switzerland, 2019.
- [6] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, “A survey on mobile edge computing: the communication perspective,” *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2322–2358, 2017.
- [7] X. Xu, Y. Li, T. Huang et al., “An energy-aware computation offloading method for smart edge computing in wireless metropolitan area networks,” *Journal of Network and Computer Applications*, vol. 133, pp. 75–85, 2019.
- [8] L. Qi, X. Zhang, W. Dou et al., “A two-stage locality-sensitive hashing based approach for privacy-preserving mobile service recommendation in cross-platform edge environment,” *Future Generation Computer Systems*, vol. 88, pp. 636–643, 2018.
- [9] X. Xu, X. Zhang, H. Gao, X. Yuan, L. Qi, and W. Dou, “BeCome: blockchain-enabled computation offloading for IoT in mobile edge computing,” *IEEE Transactions on Industrial Informatics*, vol. 10, 2019.
- [10] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of Machine Learning*, MIT Press, Cambridge, MA, USA, 2012.
- [11] X. Xu, Y. Chen, X. Zhang et al., *A Blockchain-Based Computation Offloading Method for Edge Computing in 5G Networks*, Software: Practice and Experience, Hoboken, NJ, USA, 2019.
- [12] G. Manogaran, D. Lopez, and N. Chilamkurti, “In-mapper combiner based MapReduce algorithm for processing of big climate data,” *Future Generation Computer Systems*, vol. 86, pp. 433–445, 2018.
- [13] L. Qi, R. Wang, C. Hu, S. Li, Q. Xu, and X. Hu, “Time-aware distributed service recommendation with privacy-preservation,” *Information Sciences*, vol. 480, pp. 354–364, 2019.

- [14] X. Xu, Q. Liu, Y. Luo et al., "A computation offloading method over big data for IoT-enabled cloud-edge computing," *Future Generation Computer Systems*, vol. 95, pp. 522–533, 2019.
- [15] A. Ronald, "Howard, dynamic programming," *Management Science*, vol. 12, no. 5, pp. 317–348, 1966.
- [16] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.
- [17] H. Liu, *Feature Selection for Knowledge Discovery and Data Mining*, Springer, New York, NY, USA, 2012.
- [18] K. Kira and A. Larry, "A practical approach to feature selection," in *Proceedings of the Ninth International Workshop on Machine Learning*, Aberdeen, Scotland, UK, 1992.
- [19] I. Kononenko, "Estimating attributes: analysis and extensions of Relief," in *Machine Learning: ECML-94*, Springer, Berlin, Heidelberg, 2005.
- [20] Y. Zhang, C. Ding, and T. Li, "Gene selection algorithm by combining ReliefF and MRMR," *BMC Genomics*, vol. 9, no. 2, p. S27, 2008.
- [21] D. Kong, C. Ding, H. Huang, and H. Zhao, "Multi-label ReliefF and F-statistic feature selections for image annotation," *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 2352–2359, 2012.
- [22] H. Jason, "Epistasis analysis using ReliefF," *Epistasis: Methods and Protocols*, pp. 315–325, Springer, New York, NY, USA, 2015.
- [23] P. Benioff, "The computer as a physical system: a microscopic quantum mechanical Hamiltonian model of computers as represented by turing machines," *Journal of Statistical Physics*, vol. 22, no. 5, pp. 563–591, 1980.
- [24] P. Richard, "Simulating physics with computers," *International Journal of Theoretical Physics*, vol. 21, no. 6-7, pp. 467–488, 1982.
- [25] W. Peter, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM Review*, vol. 41, no. 2, pp. 303–332, 1999.
- [26] L. K. Grover, "Quantum mechanics helps in searching for a needle in a haystack," *Physical Review Letters*, vol. 79, no. 2, pp. 325–328, 1997.
- [27] Z. Qu, T. Zhu, J. Wang, and X. Wang, "A novel quantum steganography based on Brown states," *CMC-Computers, Materials & Continuum*, vol. 56, no. 1, pp. 47–59, 2018.
- [28] W. Liu, P. Gao, Y. Wang, W. Yu, and M. Zhang, "A unitary weights based one-iteration quantum perceptron algorithm for non-ideal training sets," *IEEE Access*, vol. 7, pp. 36854–36865, 2019.
- [29] W. Liu, P. Gao, Z. Liu, H. Chen, and M. Zhang, "A quantum-based database query scheme for privacy preservation in cloud environment," *Security and Communication Networks*, vol. 2019, p. 14, 2019.
- [30] L. Lamata, M. Sanz, and E. Solano, "Quantum machine learning and bioinspired quantum technologies," *Advanced Quantum Technologies*, vol. 2, pp. 7-8, 2019.
- [31] Z. Qu, Z. Li, G. Xu, S. Wu, and X. Wang, "Quantum image steganography protocol based on quantum image expansion and grover search algorithm," *IEEE Access*, vol. 7, pp. 50849–50857, 2019.
- [32] Z. Qu, Z. Cheng, and X. Wang, "Matrix coding-based quantum image steganography algorithm," *IEEE Access*, vol. 7, pp. 35684–35698, 2019.
- [33] W. Liu, P. Gao, W. Yu, Z. Qu, and C. Yang, "Quantum Relief algorithm," *Quantum Information Processing*, vol. 17, no. 10, p. 280, 2018.
- [34] A. Michael and L. Isaac, *Quantum Computation and Quantum Information*, p. 221, Cambridge University Press, Cambridge, United Kingdom, 2000.
- [35] G. Long, "Grover algorithm with zero theoretical failure rate," *Physical Review A*, vol. 64, no. 2, 2001.
- [36] S. Zhou, T. Loke, J. A. Izaac, and J. Wang, "Quantum fourier transform in computational basis," *Quantum Information Processing*, vol. 16, no. 3, 2017.
- [37] Y. Chen, S. Wei, X. Gao, C. Wang, J. Wu, and H. Guo, "An optimized quantum maximum or minimum searching algorithm and its circuits," 1908.
- [38] K. Kira and A. Larry, "Feature selection problem: traditional methods and a new algorithm," in *Proceedings of the Tenth National Conference on Artificial Intelligence*, pp. 129–134, San Jose, CA, USA, July 1992.
- [39] R. S. Smith and W. J. Zeng, "A practical quantum instruction set architecture," 2017.
- [40] Z. Qu, S. Wu, M. Wang, and X. Wang, "Effect of quantum noise on deterministic remote state preparation of an arbitrary two-particle state via various quantum entangled channels," *Quantum Information Processing*, vol. 16, no. 306, pp. 1–25, 2017.
- [41] X. Xu, C. He, Z. Xu, L. Qi, S. Wan, and M. Z. A. Bhuiyan, "Joint optimization of offloading utility and privacy for edge computing enabled IoT," *IEEE Internet of Things Journal*, vol. 34, 2019.
- [42] H. Xiong, Y. Zhao, L. Peng, H. Zhang, and K.-H. Yeh, "Partially policy-hidden attribute-based broadcast encryption with secure delegation in edge computing," *Future Generation Computer Systems*, vol. 97, pp. 453–461, 2019.
- [43] J. Zhang, B. Chen, Y. Zhao, X. Cheng, and F. Hu, "Data security and privacy-preserving in edge computing paradigm: survey and open issues," *IEEE Access*, vol. 6, pp. 18209–18237, 2018.
- [44] X. Xu, Y. Xue, L. Qi et al., "An edge computing-enabled computation offloading method with privacy preservation for internet of connected vehicles," *Future Generation Computer Systems*, vol. 96, pp. 89–100, 2019.
- [45] W. Liu, Z. Chen, J. Liu, Z. Su, and L. Chi, "Full-blind delegating private quantum computation," *CMC-Computers Materials & Continuum*, vol. 56, no. 2, pp. 211–223, 2018.
- [46] W. Liu, Y. Xu, W. Wang, and Z. H. Lei, "Quantum searchable encryption for cloud data based on full-blind quantum computation," *IEEE Access*, vol. 7, pp. 186284–186295, 2019.
- [47] W. Liu, Y. Xu, J. Yang, W. Yu, and L. Chi, "Privacy-preserving quantum two-party geometric intersection," *Computers, Materials & Continuum*, vol. 60, no. 3, pp. 1237–1250, 2019.
- [48] W. Liu, Y. Xu, M. Zhang, J. Chen, and C.-N. Yang, "A novel quantum visual secret sharing scheme," *IEEE Access*, vol. 7, pp. 114374–114384, 2019.