Binary Classification Quantum Neural Network Model Based on Optimized Grover Algorithm

Wenlin Zhao [1], **Yinuo Wang** [2], **Yingjie Qu** [2], **Hongyang Ma** [2] **and Shumei Wang** [2,*]

1   School of Information and Control Engineering, Qingdao University of Technology, Qingdao 266520, China
2   School of Science, Qingdao University of Technology, Qingdao 266520, China
*   Correspondence: wangshumei@qut.edu.cn

Abstract: We focus on the problem that the Grover algorithm is not suitable for the completely unknown proportion of target solutions. Considering whether the existing quantum classifier used by the current quantum neural network (QNN) to complete the classification task can solve the problem of the classical classifier, this paper proposes a binary quantum neural network classifical model based on an optimized Grover algorithm based on partial diffusion. Trial and error is adopted to extend the partial diffusion quantum search algorithm with the known proportion of target solutions to the unknown state, and to apply the characteristics of the supervised learning of the quantum neural network to binary classify the classified data. Experiments show that the proposed method can effectively retrieve quantum states with similar features. The test accuracy of BQM retrieval under the depolarization noise at the 20th period can reach 97% when the depolarization rate is 0.1. It improves the retrieval accuracy by about 4% and 10% compared with MSE and BCE in the same environment.

Keywords: binary classification; Grover algorithm; QNN

1. Introduction

The Internet has many different kinds of data and information that are intersected and stored on social networks, prompting many different research fields to start to pay attention to social networks. Users on social networks obtain the resources that they need by visiting web pages [1–4]. The key to studying social networks is to analyze how these social networks are used. The data analyzed can be used to improve the social network itself, making it more convenient for users to browse the data required. They can also be used to analyze users' preferences to deliver advertisements at designated points. They can also be used to analyze user behavior and to predict the transactions that users participate in [5,6]. One of the main ways to analyze these results is to perform extensive data analysis on the weblogs of these sites [7,8]. Every time a user requests a page or some of the resources on that page, such as video, sound, etc., a new record is added to the weblog of the site [9]. This information contains information about the user's favourite pages (i.e., the most frequently visited pages), the sequence of visits to ordinary pages, and even hints at the user's characteristics. This information analysis method can be called web page using data mining (WUM) based on weblog information [10–12]. To use WUM, a sequence of interactions between a single user and a web page needs to be extracted. The resulting file should contain at least the following fields: the user's IP address, timestamp, requested resource, code for the result of the operation, the previous web address before entering the web page, and the browser used [13,14]. Using and analyzing this sequence, the pattern of the user's access to the web page can be obtained.

The value of big data is essentially reflected as follows: it provides new thinking and means for a human to understand complex systems. In theory, a virtual digital image of the natural world can be constructed by digitizing the real world on a sufficiently small

scale of time and space that carries the running rules of the real world. On the premise of sufficient computing power and efficient data analysis methods, an in-depth analysis of this virtual digital image will make it possible to understand and to discover the existing complex system's operation behavior, state, and law. Big data provides a new way of thinking and a new means for exploring objective laws and for transforming nature and society for human beings. Due to the high and increasing demand for big data mining and analysis work, the era is forced to gradually use more efficient quantum scientific research technology [15] to meet the gap of technology improvement means, and then to improve the efficiency of information extraction work and to promote the progress of more scientific research. In view of the above requirements and improved thinking, this paper makes use of the special advantages of quantum algorithms compared with traditional iterative algorithms [16], as a new field of quantum neural network, and studies the neural network autonomous learning scheme [17] for network log information extraction, which highlights the high-value prospects of the quantum field in big data mining and analysis.

This search hassle can be carried out in $O(N)$ with the use of Grover's quantum search algorithm. Grover's quantum search algorithm [18] makes use of the amplitude amplification approach in quantum computing to attain quadratic acceleration in unsorted search problems. Grover's quantum search algorithm has been efficiently applied on classical computer systems with the usage of a quantum laptop language (QCL). For an unordered database search, the Grover algorithm achieves quadratic acceleration compared with the classical algorithm. Then, analysis, induction, and variant research are carried out [19,20]. Except for the lower bound, Grover's methodology can be used for the case where the λ fraction of the target term is unknown. The fixed point strategy and the trial and error method are the two quantum search strategies that can be implemented.

The Binary QNN Model in this paper is a kind of model based on the Grover algorithm and the QNN supervised learning algorithm. First of all, this is based on the traditional Grover algorithm being analyzed and improved, using Younes' algorithm to improve the search algorithm efficiency, inserting this into the iterative learning process of the quantum neural network [21,22], and using quantum processes to promote the efficiency of the algorithm and neural network to realize multiple network synchronization searching and learning [23,24] for each iteration algorithm to improve the efficiency of solutions [25]. The quantum neural network learning scheme in this paper can be applied to quickly find the user's IP address in massive weblogs, and then accurately and efficiently identifying and classifying relevant and valuable information such as the IP addresses of network logs. The results can be sorted according to the number of search categories to identify and analyze user behavior accurately. It can not only quickly discover the person's record of specific interests, but also can divide the session of a single user, which ensures the accuracy of person document identification and improves the effectiveness of user activity queries.

This article is structured as follows. Section 2 introduces the trial-and-error method (Younes' algorithm) used in this paper, and the basic principles of the QNN supervised learning division task. Section 3 describes our binary QNN model. Section 4 describes the evolution process of the original dataset, and provides entropy analysis to show the advantages of this model. Section 5 summarizes and discusses the role of the proposed model in the development of user behavior pattern prediction.

2. Basic Conception

2.1. Supervised Learning Classification of QNN

Quantum Neural Network (QNN) is a new research field formed by the intersection of quantum physics, mathematics, computer science, information science, cognitive science, complexity science, and other disciplines. As the natural evolution of the traditional neural computing system, it makes full use of the great power of quantum computing to improve upon the information processing capacity of neural computing. By introducing the idea of the superposition of quantum states into the traditional feedforward neural network, QNN can train the quantum interval and quantify the uncertainty of the input

data of training samples. Different data will map to different magnitudes. A multi-layer excitation function is used to increase the fuzziness of the network, and to improve the accuracy and certainty of network pattern recognition [26]. Therefore, the research on the quantum neural network provides beneficial support for the combination of quantum computing and neural computing. The potential of quantum neural networks is that they take advantage of both quantum computing based on coherent superposition and neural computing based on parallel processing. For example, running a deep neural network (DNN) model on a device with limited computing power can be challenging because of the large computational power and memory requirements on the device. However, to solve this problem, a quantum neural network (QNN) has greater potential, which can save computing costs while ensuring the accuracy of DNN training.

QNN comprises quantum state preparation subcircuits and optimization tasks performed by classical controllers [27]. The fact that variable-component subcircuits utilized in QNN produce probability distributions that cannot be efficiently simulated is part of the evidence supporting the claim [28,29]. QNN's main application, similar to DNN's, is to tackle categorization tasks [30]. Practical challenges, such as recognizing handwritten digits and the features of many living creatures, can be categorized as categorization scenes [31,32].

A dataset is given

$$T = \{(x_i, y_i)\}_{i=0}^{N-1} \in (R^{N \times M}, \{0, 1\}^N) \tag{1}$$

According to N examples and M elements in the examples, a QNN is led to research $f_\theta(\cdot)$ to predict the label of a facts set T

$$\min_{\theta} \sum_{i=0}^{N-1} I_{y_i \neq f_\theta(x_i)} \tag{2}$$

where θ is the trainable parameter, and I_z is an indicator function whose value is 1 when the condition z is met; otherwise, it is zero. The quantum classifier realizes the data tag prediction function according to specific rules through the filtered data, and its basic principle is shown in Figure 1. We use quantum classifiers in the research section to specify a QNN for completing the classification task defined in Formula (2). Considering the binary task, it is necessary not only to find a decision rule in Formula (2), but also to output the index j satisfying the pre-determined black box function. Given a trained classifier $f_\theta(\cdot)$, both classical algorithms and previous quantum classifiers require at least $O(T)$ query complexity to find j.

The dataset T is constructed from a given qubit with adjustable interactions, where the qubit composition is represented by x_i and y_i. We learn the interaction from a given training set of each input–output relationship based on the classical backpropagation rule $f_\theta(\cdot)$, and taking x_i as the input to its rule, where the input–output relation is the data pair (x_i, y_i) that constitutes the dataset T. This learning process of qubits is viewed as the desired output algorithm behavior, that is, the quantum network "learns" an algorithm.

A notable theoretical result concerning quantum classifiers is the tradeoff between the computational cost and the training performance shown [33].
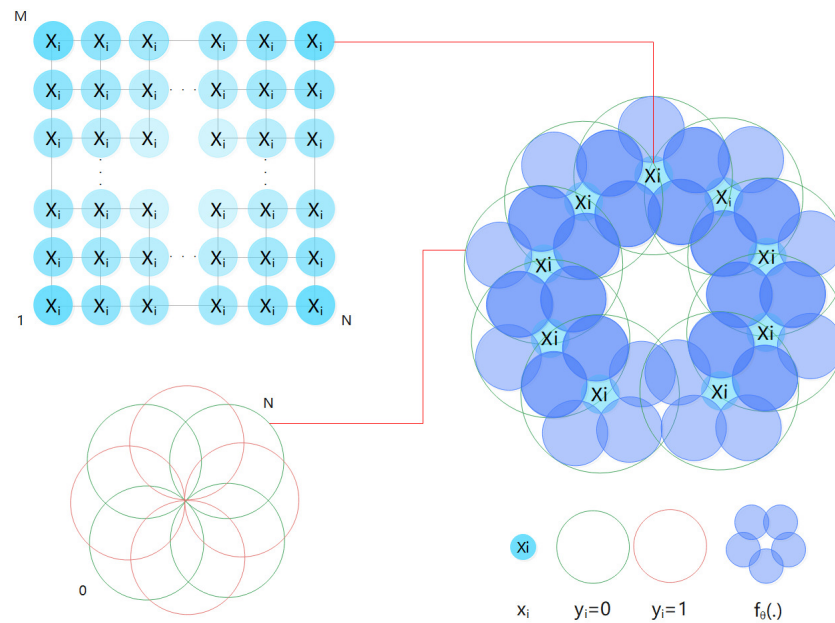
Figure 1. Basic principles of the quantum classifier. x_i randomly generated by the $R^{N \times M}$ matrix, and y_i, which can only be 0 or 1, forms N data pairs and then generates the dataset T. According to the classic backpropagation rule $f_\theta(\cdot)$, x_i is taken as the input to this rule. The qubit obtains the input–output relationship from the data pair (x_i, y_i) in the constructed dataset T, and learns the interaction in the training set of the relationship. In other words, the purpose of the quantum neural network is to use x_i as an input to learn $f_\theta(\cdot)$ rules.

2.2. Younes' Algorithm

The methodology presented by Younes, Rowe J., and Miller J. [34] in Younes' algorithm is used to carry out the quantum search by exploiting the local diffusion operator to overcome the souffle problem in the Grover algorithm. It demonstrates that regardless of whether the number of matches is known, the entire range of $1 \leq M \leq N$ can be consistently handled. It lays the theoretical foundation of the binary QNN model.

In the $|0\rangle$ and $|1\rangle$ states, part of the diffusion operator Q_i system subspace of the entanglement in additional qubit workspace performs about the inverse operation of the mean, and the inverse operation of the phase shift is -1. H is the Hadamard Gates denoted by $H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$. The diagonal representation of Q_i applied to the $n+1$ qubit system is:

$$Q_i = (H^{\otimes n} \otimes I_1)(2|0\rangle\langle 0| - I_{n+1})(H^{\otimes n} \otimes I_1) \tag{3}$$

asthe $|0\rangle$ length is 2^{n+1}, I_1 is the unit of a 2×2 matrix.

Generally, quantum structures of the well-known size $n+1$ can be expressed as:

$$|\psi\rangle = \sum_{p=0}^{N-1} \alpha_p(|p\rangle \otimes |0\rangle) + \sum_{p=0}^{N-1} \beta_p(|p\rangle \otimes |1\rangle) \tag{4}$$

Applying Q_i to $|\psi\rangle$ bits, we obtain

$$\sum_{p=0}^{N-1} (\frac{2}{N} \sum_{p=0}^{N-1} \alpha_p - \alpha_p)(|p\rangle \otimes |0\rangle) - \sum_{p=0}^{N-1} \beta_p(|p\rangle \otimes |1\rangle) \tag{5}$$

where $\frac{1}{N}\sum_{p=0}^{N-1}\alpha_j$ means the mean amplitude of subspace $\sum_{p=0}^{N-1}\alpha_p(|p\rangle\otimes|0\rangle)$. That is, the operator Q_i only performs the inversion of the means in the subspace and only changes the sign of the amplitude.

The H gate is utilized to the first n qubits to produce 2^n values to characterize the list. Then, we iteratively observe the Oracle feature U_f to map the goal in the list to 0 or 1, and we retail the outcomes such as $U_f|x,0\rangle \rightarrow |x,f(x)\rangle$; the partial diffusion operator Q_i is applied, and this step is repeated q times. Finally, the first n qubits are measured.

The variety of iterations q has to be an integer to locate a healthy shut to the change in measurements.

Setting $q=\left\lfloor\frac{\pi}{2\theta}\right\rfloor$, as $|q-\bar{q}|\leq\frac{1}{2}$, $0<\theta\leq\frac{\pi}{2}$. As $\cos(\theta)=1-\frac{M}{N}$, we can find

$$\theta\geq\frac{\sqrt{2NM-M^2}}{N} \tag{6}$$

$$q=\left\lfloor\frac{\pi}{2\theta}\right\rfloor\leq O(\sqrt{\frac{N}{M}}) \tag{7}$$

It is proven that the algorithm can be handled in the range of $1\leq M\leq N$ using the $O(\sqrt{\frac{N}{M}})$ fixed operator.

3. The Binary QNN Model

We simulate the creation of a binary analysis algorithm that uses quantum states to process information, as shown in Figure 2. The algorithm proposed in this paper is uniformly represented as a BQM field in the following content. As shown in Figure 2, BQM uses a specified variable component subcircuit U_c, and an H gate to replace the Oracle U_f. The variable component subcircuit U_c, based on the training data, can conditionally flip a flag qubit. The tagged qubit is then used as part of the H gate to guide a Grover search algorithm to identify the index of the specified example; i.e., the state of the tagged qubit, such as "0" or "1", determines the probability of success in identifying the target index. BQM optimizes the trainable parameters of the variable component subcircuit U_c. When the corresponding training instance is positive, the success probability of sampling a target index is maximized. Otherwise, BQM minimizes the probability of the success of sampling target indicators. The design of our algorithm has some advantages in terms of query complexity by inheriting attributes from the Grover search algorithm and performing binary classification tasks on these attributes while allowing the setting of search constraints [35]. Under the above observation, a quantum classifier must have certain advantages [36–38].

3.1. Pretreatment Stage of a Dichotomous Task

In the pretreatment stage, a dichotomous task uses the dataset T defined in Equation (1) as the extended dataset $\hat{T}$.

To apply the Grover search algorithm to obtain index $i=K-1$, for $K\in[N]$, the Kth pair data training rules T_k are as follows.

$$T_K=[(x_k^{(0)},y_k^{(0)}),(x_k^{(1)},y_k^{(1)}),\ldots,(x_k^{(K-1)},y_k^{(K-1)})] \tag{8}$$

The pair of data in T_k is like the Kth pair of $\hat{T}$; this means that $(x_k^{(K-1)},y_k^{(K-1)})=(x_k,y_k)$. The first $K-1$ pair of $T_k=\{(x_k^{(i)},y_k^{(i)})\}_{i=0}^{K-2}$ uniformly samples from a subset of $\hat{T}$, when every label $\{y_k^{(i)}\}_{i=0}^{K-2}$ is the opposite of y_k.

$y_k\in\{0,1\}$, $\hat{T}^{(0)}$ and $\hat{T}^{(1)}$ are constructed, which contain only those examples of $\hat{T}$ with the labels '0' and '1', respectively. When $y_k=0$, the pair samples before K from $\hat{T}^{(1)}$; it is same as for the situation where $y_k=1$, in which the pair samples before K from $\hat{T}^{(0)}$.

Various quantum classifiers encode T_k into quantum states in different ways [39]. For the sake of notation, we indicate that $|\Phi^k\rangle$ that analogously connects with the Kth example is

$$|\Phi^k\rangle = U_{data}|0\rangle = \sum_{i=0}^{K-1} \frac{1}{\sqrt{K}}|g(x_i)\rangle|i\rangle \tag{9}$$

as $g(\cdot)$ is a coding operation.
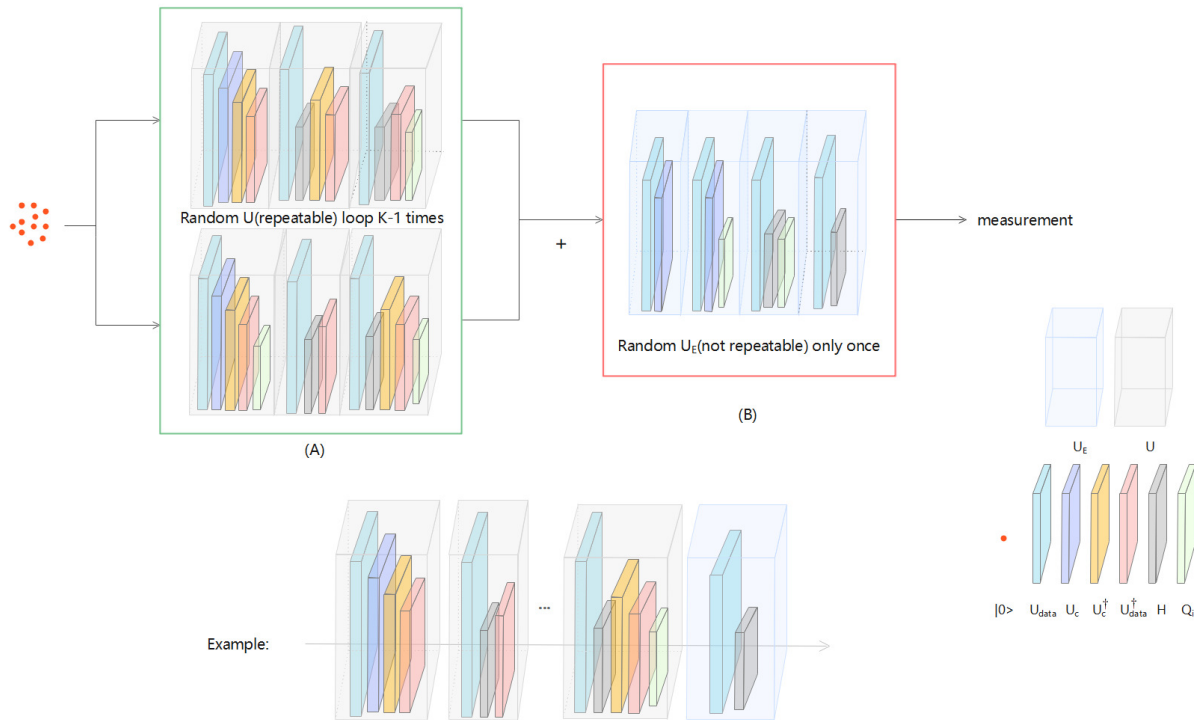
Figure 2. The paradigm of BQM. (**A**) The first K-1 loop uses U, defined in Equation (12), which consists of unitary operators (namely U_{data}, U_c, H, and Q_i). (**B**) The last cycle uses the unitary operation U_E defined in Equation (13). The qubit interacts with U_c and Q_i to form the feature register and data register.

3.2. The Training Process of the Learning Plan

Compared with the traditional Grover algorithm, combining the variational learning method and the Grover search algorithm produces quantum advantages [40–44]. The adopted variable component subcircuit U_c is designed to find a hyperplane to keep the last pair in the T_k away from the pair of samples before K.

For the variational quantum circuits $U_c(\theta)$ in BQM, a NISQ device scheme consists of a trainable single-qubit gate and two-qubit gates such as CNOT or CZ, which implement generation and discrimination tasks using variational hybrid quantum-classical algorithms. U_c is denoted as $U_c = \prod_{c=1}^{C} U(\theta^c)$, where each layer $U(\theta^c)$ contains $O(poly(N))$ parameterized single-qubit gates and at most, $O(poly(N))$ fixed two-qubit gates with the same layout.

In the optimal situation, given a initial state $|\Phi^k\rangle$ defined in Equation (9), U_c is applied to obtain the following goals:

1. If the pair of samples before K as T_k analogously connects with the label $y_k = 0$, the expected state is

$$(U_c \otimes I)(U_{\text{data}}|0\rangle)_{y_k=0} = \sum_{i=0}^{K-1} \frac{1}{\sqrt{K}}|0\rangle|i\rangle \tag{10}$$

2. If the pair of samples before K as T_k analogously connects with the label $y_k = 1$, the expected state is

$$(U_c \otimes I)(U_{\text{data}}|0\rangle)_{y_k=1} = \sum_{i=0}^{K-1} \frac{1}{\sqrt{K}}|1\rangle|i\rangle \tag{11}$$

The label $y_k = 0$ (or $y_k = 1$) as the quantum state of the R_F characteristics register the first qubits $|0\rangle$ (or $|1\rangle$). As shown in Figure 2A, state $(U_c \otimes I_I)U_{data}|0\rangle$ was prepared. Our binary QNN model iteratively applied the H door to register by the characteristics of the first qubit and index on the index register control, using the U_{data} register and the U_c calculation characteristics, and to finish the first cycle, it applied the diffusion operator Q_i to the index register. All quantum processes, such as U, are part of a period.

$$U = Q_i \circ U_{data}^\dagger \circ (U_c \otimes I)^\dagger \circ H \circ (U_c \otimes I) \circ U_{data} \tag{12}$$

Define $Q_i = \mathrm{I} \otimes (\frac{2}{K} \sum_i |i\rangle\langle i| - I_I)$. Except on a loop, the binary QNN model is repeatedly in the initial state $|0\rangle$ to U, and the application of the unitary operation is replaced with

$$U_E = Q_i \circ H \circ (U_c \otimes I) \circ U_{data} \tag{13}$$

The brown shade in Figure 2B shows this. According to the traditional Grover search algorithm [45], before making quantum measurements, the binary QNN model polls U and U_E for a total of $O(\sqrt{K})$ times.

3.3. The Evolution of the Quantum State

We analyze how quantum states evolve under $y_k = 0$ and $y_k = 1$:

- After interaction with unitary $U_c \otimes I$, using the Equation (10) input state $\Phi_k(y_k = 0)$, this state can be converted to $\frac{1}{\sqrt{K}}\sum_{i=0}|0\rangle|i\rangle$. For all computing in $i \in [K-1]$, this means that the quantum operation $Q_i \circ U_{data}^\dagger \circ (U_c \otimes I_I)^\dagger$ does not change state.

$$\frac{1}{\sqrt{K}}(H^{\otimes n} \otimes I)(2|0\rangle\langle 0| - I_{n+1})(H^{\otimes n} \otimes I)\sum_{i=0}^{K-1}|0\rangle|i\rangle = \sum_{i=0}^{K-1}\frac{1}{\sqrt{K}}|0\rangle|i\rangle \tag{14}$$

When we measure the indicator register of the output state, the sampling $i \in [K-1]$ for calculating the base i is distributed.

- After interaction with unitary $U_c \otimes I$ using the Equation (11) input state $\Phi_k(y_k = 1)$, this state can be converted to $\frac{1}{\sqrt{K}}\sum_{i=0}|1\rangle|i\rangle$.
  Mathematically, the result state is generated after interaction with H

$$H \circ (U_c \otimes I)(U_{data}|0\rangle)_{y_k=1} = \frac{1}{\sqrt{K}}|0\rangle\sum_{i=0}^{K-2}|i\rangle - \frac{1}{\sqrt{K}}|1\rangle|i^*\rangle \tag{15}$$

where $|i^*\rangle$ for calculating the base $|K-1\rangle$. The calculation operation $U_{data}^\dagger \circ (U_c \otimes I)^\dagger$ and the diffusion operation Q_i are used to increase $|i^*\rangle$ probability.
After the first cycle, the generated state is generated

$$U(U_{\text{data}}|0\rangle)_{y_k=1} = \frac{(K-4)}{K\sqrt{K}}|0\rangle\sum_{i=0}^{K-2}|i\rangle + \frac{3K-4}{K\sqrt{K}}|0\rangle|i^*\rangle \tag{16}$$

where Equation (12) defines U. According to Grover's algorithm, the chance of sampling i^* will increase to $\frac{(3K-4)^2}{K^3}$.

3.4. The Loss Function

With the observation above leading to Theorem 1, the proof is given above:

Theorem 1. *For BQM, under the optimal setting, if the label of the last item of T_k is $y_k = 1$, the probability of the sampling resulting in $i^* = K - 1$ is asymptotically 1.*

Proof of Theorem 1. We discussed the case where the last entry in T_k has labels $y_k = 1$ and $y_k = 0$.

In the instance of $y_k = 0$, assuming that the label of the final item in T_k is $y_k = 0$, it is possible to determine from Equation (14) that after the first cycle, the generation state of BQM is

$$U_c|\Phi_k(y_k = 0)\rangle = U|0\rangle = \sum_{i=0}^{K-1} \frac{1}{\sqrt{K}}|0\rangle|i\rangle \tag{17}$$

The chance of picking any index when U (apply to $|0\rangle$) is the same, according to the formula above. After U is applied to $|0\rangle$ via induction, with the migration with time n, the state changes as

$$\prod_{i=0}^{N} U^i|0\rangle = \sum_{i=0}^{K-1} \frac{1}{\sqrt{K}}|0\rangle|i\rangle \tag{18}$$

where the given N is any positive integer, and the probability of sampling $|i^*\rangle$ is $\frac{1}{K}$. In the last loop, the quantum operation U_E defined in Equation (13) is applied to state $\prod_{i=0}^{N} U^i|0\rangle$ and the resulting state is

$$U_E \prod_{i=0}^{N} U^i|0\rangle = Q_i \circ H \circ (U_c \otimes I) \circ U_{data} \sum_{i=0}^{K-1} \frac{1}{\sqrt{K}}|0\rangle|i\rangle$$
$$= \sum_{i=0}^{K-1} \frac{1}{\sqrt{K}}|0\rangle\Lambda \tag{19}$$

where the first equality uses Equation (18); the second equation uses Equation (15) and exploits the application of the diffusion operator $Q_i = (H^{\otimes n} \otimes I_1)(2|0\rangle\langle 0| - I_{n+1})(H^{\otimes n} \otimes I_1)$, then $\Lambda = \frac{1}{\sqrt{K-1}} \sum_{i=0}^{K-2} |i\rangle + |i^*\rangle$.

In the instance of $y_k = 1$, assuming that the label of the last item in T_k is $y_k = 1$, it is possible to determine from Equation (16) that after the first cycle, the generation state of BQM is

$$U_c|\Phi_k(y_k = 1)\rangle = |0\rangle \otimes (\frac{(K-4)}{K\sqrt{K}} \sum_{i=0}^{K-2} |i\rangle + \frac{3K-4}{K\sqrt{K}}|i^*\rangle) \tag{20}$$

The chance of picking any index when U (apply to $|0\rangle$) is the same, according to the formula above. After U is applied to $|0\rangle$ by induction, with the migration with time n, the state changes as follows:

$$\prod_{i=0}^{n} U^i|0\rangle = |0\rangle \otimes \frac{1}{\sqrt{K}} \left[\hbar \sum_{i=0}^{K-2} |i\rangle + \lambda|i^*\rangle \right] \tag{21}$$

where given that n is any positive integer, $\hbar = \cos(2n\alpha) - \frac{1}{\sqrt{K-1}}\sin(2n\alpha)$, $\sin\alpha = \frac{1}{\sqrt{K}}$, $\lambda = \sqrt{K-1}\sin(2n\alpha) + \cos(2n\alpha)$. In the last loop, the quantum operation U_E defined in

Equation (13) is applied to state $\prod_{i=0}^{N} U^i |0\rangle$, and the resulting state is

$$
\begin{aligned}
U_E \prod_{i=0}^{N} U^i |0\rangle &= Q_i \circ H \circ (U_c \otimes I) \circ U_{data} |0\rangle \otimes \frac{1}{\sqrt{K}} \hbar \sum_{i=0}^{K-2} |i\rangle + \lambda |i^*\rangle) \\
&= Q_i \frac{\left[\hbar |0\rangle \sum_{i=0}^{K-2} |i\rangle + \lambda |1\rangle |i^*\rangle \right]}{\sqrt{K}} \\
&= \frac{\left[(K-2)\hbar |0\rangle \sum_{i=0}^{K-2} |i\rangle + 2\sqrt{K-1}\lambda |1\rangle |i^*\rangle \right]}{\sqrt{K^3}}
\end{aligned}
\tag{22}
$$

where the first equality uses Equation (21) and the second equation uses Equation (16) to design the feature register. It uses H to flip the phase of $|i\rangle$ whose first qubit of the feature register is $|1\rangle$, and the last equation comes from the application of the diffusion operator Q_i.

According to Equation (22), in the ideal situation, the probability of sampling i^* is near to 1 when $n \sim O(\sqrt{K})$, and then $(K-1)\sin(2n\alpha) + \sqrt{K-1}\cos(2n\alpha)$ is close to 1.

The result of Equation (19) shows that when $y_k = 0$, the probability of sampling i^* never increases. Thus, we can follow that the sampling probability of the result i^* asymptotically approaches one if and only if the label of the last term of T_k is $y_k = 1$. $\square$

According to Theorem 1 of the BQM's special property, the output distribution is different for different labels of the input T_k while performing the binary classification task. According to the analysis of Theorem 1 mentioned above, the calculation basis $i = K - 1$ will be present in the output state of the BQM; that is, $U_E U^{O(\sqrt{K})} |0\rangle$, which corresponds to $y_k = 1$, and its probability is close to 1. The matching output state for $y_k = 0$ will, however, include the same computational foundation $i \in [K - 1]$.

According to the mechanism of the Grover search algorithm, the loss function of BQM is deduced as

$$
\min_{\theta} L(\theta) = s\left(\frac{1}{2} - y_k\right) Tr((|1\rangle\langle 1|) \otimes H \otimes (|i^*\rangle\langle i^*|)\Delta_{\theta})
\tag{23}
$$

where $s(\cdot)$ is the sign function, $\Delta_{\theta} = U_E U(\theta)^{O(\sqrt{K})} |0\rangle\langle 0| (U_E U(\theta)^{O(\sqrt{K})})^{\dagger}$, and $U(\theta)$ is defined in Equation (12).

The success probability of sampling i^* and obtaining the first feature qubit as '1'('0') is maximized (minimized) when $y_k = 1$ ($y_k = 0$), when faced with the challenge of minimizing the loss function $L(\theta)$.

### 3.5. Gradient-Based Parameter Optimization

The optimization method in this paper uses a multiple-layer parameterized quantum circuit (MPQC), according to the principle that the arrangement of quantum gates in each layer is the same [46], and the operation formed by the layer c is expressed as $U(\theta^c)$, produced by quantum states produced by MPQC

$$
|\omega\rangle = \prod_{c=1}^{C} U(\theta^c) |0\rangle^{\otimes N}
\tag{24}
$$

where C is the total number of layers. BQM uses MPQC to construct U_c

$$
U_c(\theta) = \prod_{c=1}^{C} U(\theta^c)
\tag{25}
$$

The circuit layout of $U(\theta^c)$ at layer l is shown in Figure 3. When the number of layers is C, the total number of trainable parameters of BQM is 2MC.
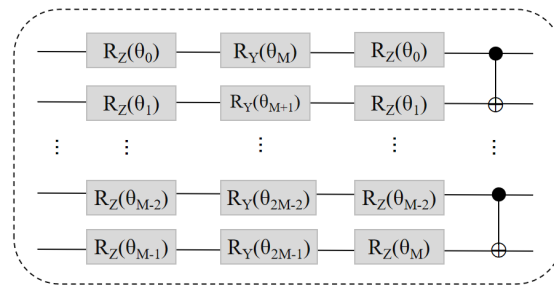
Figure 3. The realization of the lth layer $U(\theta^c)$. It is assumed that the lth layer $U(\theta^c)$ interacts with M qubits. Three trainable parameterized gates R_Z, R_Y, and R_Z are first applied to each qubit, followed by the $M-1$ *CNOT* gates.

The update rules of BQM at the jth iteration are as follows

$$\theta^{(j+1)} = \theta^{(j)} - \zeta \frac{L(\theta^{(j)}, T_j)}{\partial \theta} \tag{26}$$

where ζ is the learning rate. Given the explicit form of $L(\theta)$ in the defined Equation (23), the gradient of $L(\theta^{(j)}, T_j)$ can be rewritten as

$$\frac{\partial L(\theta^{(j)}, T_j)}{\partial \theta} = s(\frac{1}{2} - y_j) \frac{\partial Tr(\prod \Delta_{\theta^{(j)}})}{\partial \theta} \tag{27}$$

where y_j is the label of the last item in T_j, $s(\cdot)$ is the symbol function, and $\prod$ is the measurement operator.

BQM employs a gradient-based method, according to the parameter displacement rule, to obtain the gradient $\frac{\partial Tr(\prod \Delta_{\theta^{(j)}})}{\partial \theta}$, to optimize θ. The parameter shift rule [47] iteratively calculates each gradient entry under its guiding principle.

For $e \in [2NC]$, only the eth parameter is rotated by $\pm\frac{\pi}{2}$, i.e.,

$$\theta_{\pm}^{(j)} = [\theta_0^{(j)}, \dots, \theta_{e-1}^{(j)}, \theta_e^{(j)} \pm \frac{\pi}{2}, \theta_{e+1}^{(j)}, \dots, \theta_{2NC-1}^{(j)}] \tag{28}$$

Combining Equations (26)–(28), the update rule of BQM at the eth iteration of the e item is

$$\theta_e^{(j+1)} = \theta_e^j - \zeta s(\frac{1}{2} - y_j) \frac{\partial Tr(\prod \Delta_{\theta_{\pm}^{(j)}})}{2} \tag{29}$$

where $\Delta_{\theta_{\pm}^{(j)}} = U_E U(\theta_+^{(j)})^{O(\sqrt{K})} |0\rangle\langle 0| (U_E U(\theta_-^{(j)})^{O(\sqrt{K})})^{\dagger}$.

3.6. Circuit Implementation of Label Prediction

After the training of BQM is completed, the trained U_c can use the corresponding circuit (as shown in Figure 4) to predict the label of an instance with $O(1)$ query complexity.

Denoting the new input as (x, y), we encode a into quantum states using the same encoding method used during training; i.e., $|\widetilde{x}\rangle = |g(x)\rangle$, then we apply the trained U_c to $|\widetilde{x}\rangle$.

When the size of the dataset loaded by the binary QNN model is K, a well-trained binary QNN model can obtain the index with $O(\sqrt{\frac{K}{MT^2}})$ query complexity.
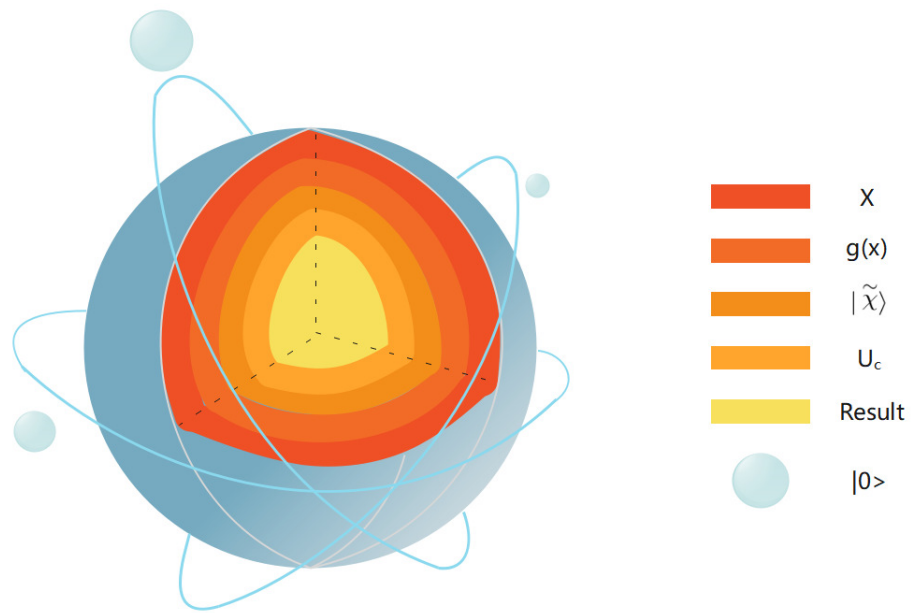
Figure 4. Circuit implementation of BQM prediction. Use g($\cdot$) as in the training process. The encoding method prepares the state $|g(x)\rangle$ and applies the trained variable component subcircuit U_c to $|\widetilde{\chi}\rangle$.

3.7. Synthetic Construction of Datasets

Given the training example $x_i = (\alpha^{(j)}, \beta^{(j)}) \in \mathbb{R}^2$, the embedded function $f(\alpha^{(j)}, \beta^{(j)})$ used to encode x_i into a quantum state is represented as

$$f(\alpha^{(j)}, \beta^{(j)}) = (R(\gamma(\alpha^{(j)}, \beta^{(j)})) \otimes R(\gamma(\alpha^{(j)}, \beta^{(j)})))|0\rangle^{\otimes 2} \tag{30}$$

where $\gamma(\alpha^{(j)}, \beta^{(j)}) = (\alpha^{(j)}, \beta^{(j)})^2$ is a specified mapping function. The above formula means that g (xi) can be converted into a series of quantum operations, the implementation of which is shown in Figure 5a. To encode multiple training examples into quantum states simultaneously, we should treat $f(x_i)$ as a controlled version, the implementation of which is shown in Figure 5b.

3.8. The Details of BQM

The implementation of GBLS is shown in Figure 5c. In it, the data encoding the unitary U_{data} consists of a controlled set of $f(x_i)$ quantum operations. The implementation of encoding unity U_{data} depends on the size of the batch B. For the quantum kernel classifier with BCE loss and MSE loss ($B = M$), it can be seen from Formula (30) that the unitary encoding is

$$U_{data} = R(\gamma(\alpha^{(j)}, \beta^{(j)})) \otimes R(\gamma(\alpha^{(j)}, \beta^{(j)})) \tag{31}$$

For a quantum kernel classifier with MSE loss ($B = M/4$), the implementation of encoding unitary U_{data} is the same as that of BQM, as shown in Figure 5.
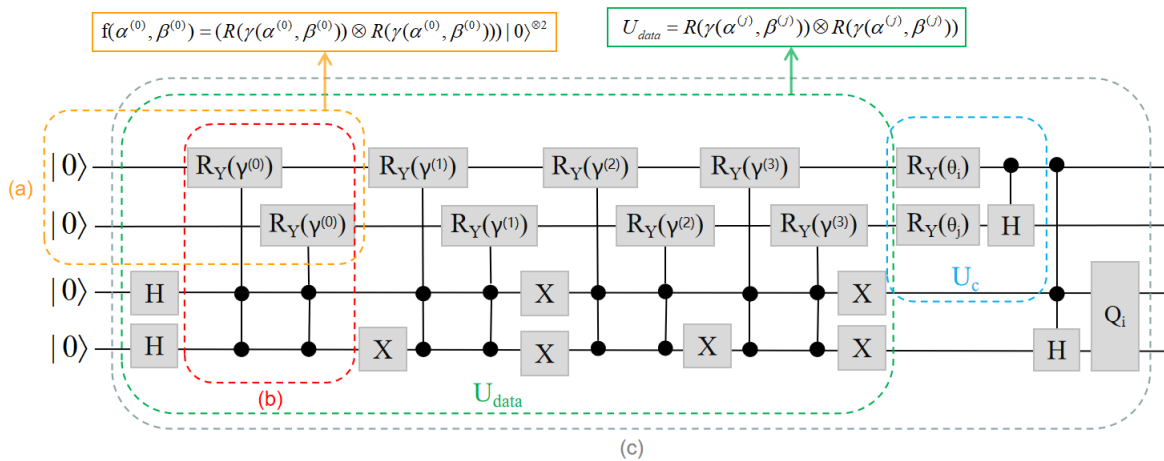
Figure 5. Implementation of BQM in numerical simulation. (**a**) The circuit implementation of the encoded unitary U_{data} corresponding to the feature map $f(x_i)$ is illustrated. (**b**) The realization of quantum operation $f(x_i)$. (**c**) The implementation of BQM, given input $T_k = \{x_i, x_j, x_m, x_n\}$.

4. Results

We will analyze the security of the proposed BQM, intuitively express the evolution and generation process of the dataset by using the dot plot, and evaluate the algorithm's performance through entropy analysis and testing [48–50].

4.1. Dataset Evolution

This algorithm's dataset generation process is shown in Figure 6. It uses the top $K-1$ pair and the Kth pair in the original dataset for label classification and uniform sampling. It divides all data pairs into sub-datasets labeled as 1 (or 0) according to the y value of 0 (or 1).

The yellow cube in the figure represents the data point in the data pair whose value of y is 1, the blue cube represents the data point in the data pair whose value of y is 0, and the point whose circle in red represents the Kth data point. The specific rules are as follows:

1. Represents the original dataset when $y_k = 1$ as the 'A' cube or when $y_k = 0$ as the 'B' cube in the Kth pair of data (x_k, y_k);
2. Represents the uniform sampling from the sub-dataset labeled 1 in Figure 6a to generate a new dataset in Figure 6b;
3. Represents the uniform sampling from the sub-dataset labeled 0 in Figure 6a to generate a new dataset in Figure 6c.

4.2. Stability and Convergence Analysis

Define a utility-bound R as a utility measure to evaluate the distance between the optimization result and the stationary point in the optimized environment.

$$R = \mathbb{E}[||\nabla_\theta L(\theta^{(j)})||]^2 \leq \varepsilon(j) \tag{32}$$

For the BQM quantum classifier with a depolarization noise setting, the utility bound of output $\theta^{(j)} \in R^l$ after j iterations is

$$\varepsilon(j) = O(poly(\frac{l}{j(1-P)^d}, \frac{l}{BK(1-P)^d}, \frac{l}{(1-P)^d})) \tag{33}$$

where P is the depolarization rate, l is the total number of trainable parameters, K is the number of measurements to estimate the quantum expected value, d is the circuit depth of the variable component sub-circuit, and B is the number of batches.

We use the decay rate of $log(\varepsilon(j))$ to define the asymptotic convergence rate of this optimization algorithm [51,52]. According to Equation (33), the attenuation rate of $log(\varepsilon(j))$ is slower than that of $-j$, which proves that this algorithm has a sublinear convergence rate.

When $B = M$, we input each sample T_j in turn to variable component subcircuits to obtain $\nabla L(\theta, T_j)$. Once the set $\{\nabla L(\theta, T_j)\}_{j=1}^M$ is collected, the gradient $\nabla L(\theta, T)$ can be estimated by $\frac{1}{M}\sum_{j=1}^M \nabla L(\theta, T_j)$. Assuming that the number of measurements required to estimate the derivative of the JTH parameter θ_j is K, the total number of measurements obtained is MK for $\frac{1}{M}\sum_{j=1}^M \nabla L(\theta, T_j)$. Therefore, the estimate of $\nabla L(\theta, T_j)$ with l parameters requires MKl measurements.

For the above definition of utility bound R, the results show that a large number of lot B can guarantee a better realization of utility bound R by increasing the total number of measurements.
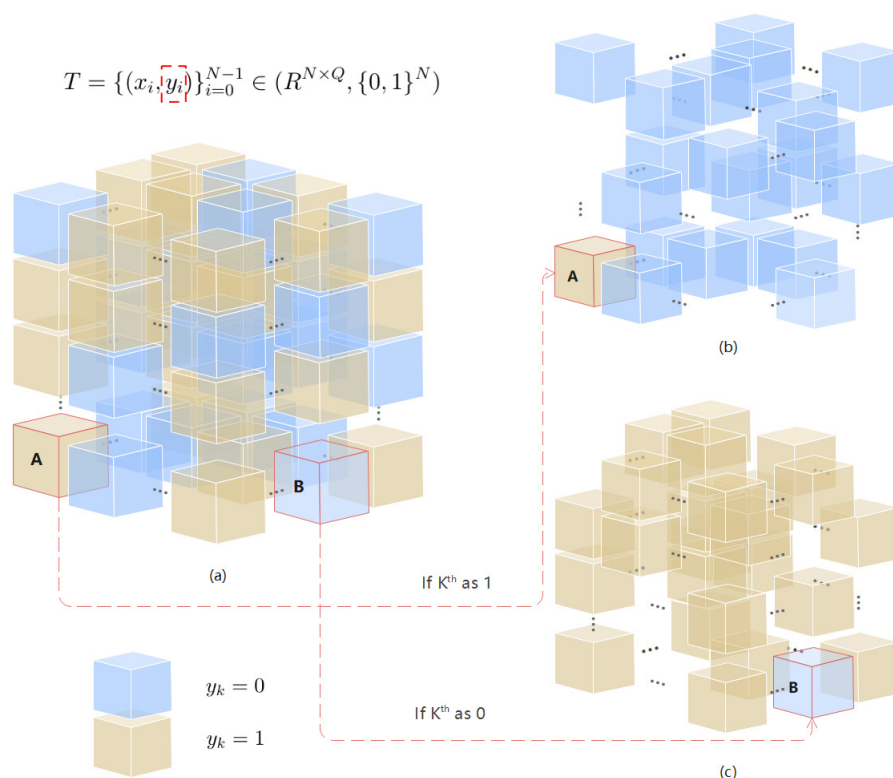
Figure 6. Dataset evolution. The yellow cube in the figure represents the data point in the data pair whose $y = 1$, the blue cube represents the data point in the data pair whose $y = 0$, and the point circle in red represents the Kth data point. (**a**) The traditional dataset when defined as T, which is defined in Equation (1). (**b**) When $K^{th} = 1$ (that is, the red grid labeled A), according to the generation formula, K^{th} is combined with the first $K - 1$ labels with $y = 0$, and the resulting dataset. (**c**) When $K^{th} = 0$ (that is, the red grid labeled B), according to the generation formula, K^{th} is combined with the first $K - 1$ labels with $y = 1$, and the resulting dataset.
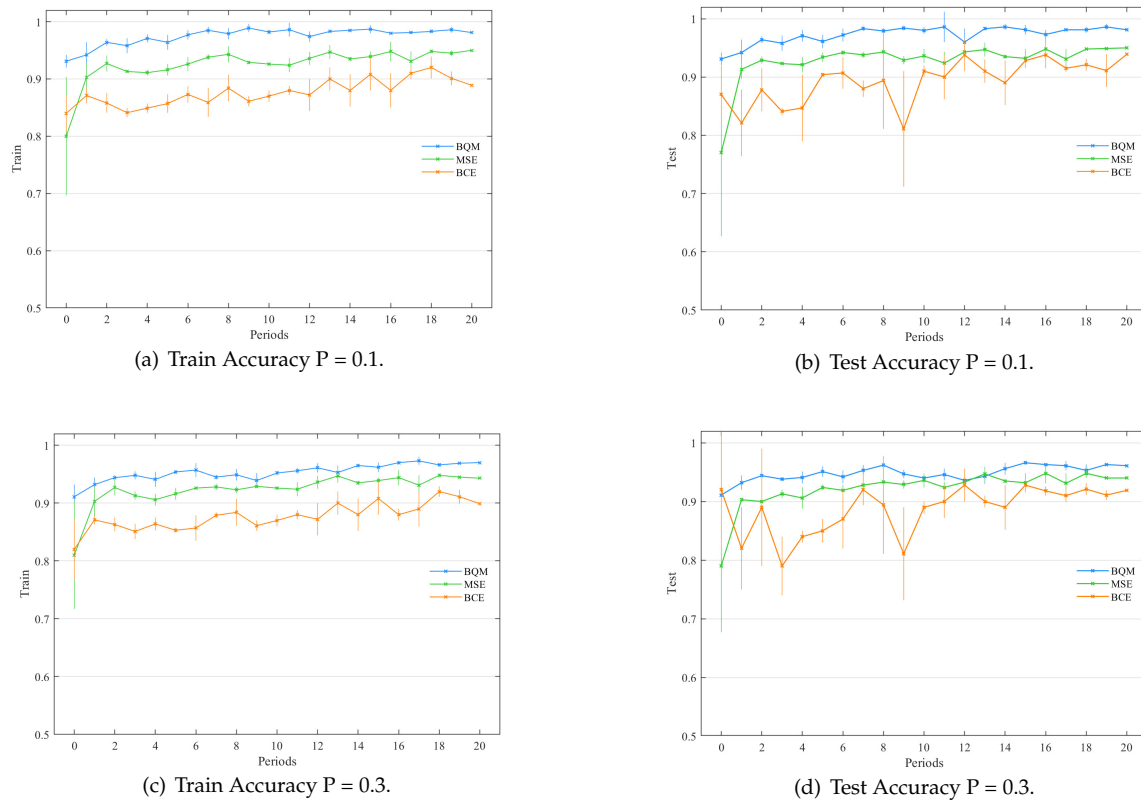
4.3. Performance Analysis under Depolarization Noise

We employ depolarization channels to mimic the system noise, since the number of measurements and the quantum system's noise are both constrained. We next examine how well BQM performs in the presence of depolarization noise [53].

If a quantum state is ω, we define the depolarizing channel ω_P acting on this quantum state as

$$\omega_P(\omega) = (1 - P)\omega + P\frac{I_l}{l} \tag{34}$$

where P is the depolarization rate, and l is the total number of trainable parameters.

We compare the performance of BQM and two other quantum kernel classifiers when quantum system noise and measurement delays are considered. Among them, BQM stands for a binary classification quantum neural network model based on the optimized Grover algorithm proposed in this paper. The two classifiers compared with BQM are defined as "BCE" and "MSE", respectively. "BCE" stands for the quantum kernel classifier with binary cross-entropy loss, and "MSE" means the quantum kernel classifier with the mean square error loss (B = N). We simulated the statistics for each of the three classifiers by repeating the values 10 times. Figure 7 illustrates the simulation findings. After 20 periods, BQM, BCE, and MSE quantum classifiers achieve the same performances. It can be observed that the quantum classifiers with MSE loss have lower convergence speeds and larger variances than the BQM and BCE classifiers. This phenomenon reflects that using BQM for classification tasks with different batches is meaningful. In Table 1, we compare the average training and testing accuracies of the BQM, BCE, and MSE quantum classifiers in the last stage. Considering the measurement error and quantum gate noise, BQM still achieves a stable performance because of its minimal variance.

(a) Train Accuracy P = 0.1.

(b) Test Accuracy P = 0.1.

(c) Train Accuracy P = 0.3.

(d) Test Accuracy P = 0.3.

Figure 7. The performance of different quantum classifiers at the different depolarization rates ($P = 0.1, 0.3$). Depolarizing noise models extracted from quantum hardware are applied to the trainable unitary $U_c(\theta)$ of these three classifiers. The labels 'BQM', 'BCE', and 'MSE' refer to the proposed Grover-based quantum classifier, the quantum kernel classifier with BCE loss, and the quantum kernel classifier with mean square error loss. (**a**,**b**) shows the variation of the train and test accuracies of BQM and the quantum kernel classifier with BCE loss with a P value of 0.1. (**c**,**d**) show the variation of the train and test accuracies of BQM and the quantum kernel classifier with BCE loss when the P value is 0.3. Vertical bars reflect the variance of the train and test accuracy at each iteration.

The binary QNN model based on Grover, quantum kernel classifier BCE loss, and quantum kernel classifier mean square error loss is reflected by the labels 'BQM', 'BCE', and 'MSE'. The train and test accuracies of the BQM quantum classifier are shown in the left and right figures. The vertical bar represents the train and test accuracy variation at

each iteration, where all hyperparameter settings are the same as those used in the above numerical simulation.

According to the numerical simulation results of the three quantum classifiers in Figure 7, BQM can obtain a good utility boundary R through some tests. When they achieve basically comparable performances, BQM reduces the number of measurements required by K = 4 times compared to quantum classifiers with BCE losses and MSE losses (B = N). This result shows that when N is larger, there is a large separation of computational efficiency between BQM and the previous B = N quantum classifier.

The above data demonstrate that, when BQM is compared to the other two quantum classifiers, the number of measurements required by BQM is decreased by four times, demonstrating BQM's efficacy.

Table 1. The average training and testing accuracies of BQM, BCE, and MSE quantum classifiers in the last stage. The value 'a ± b' means that the average precision is a and its variance is b. The labels 'BQM', 'BCE', and 'MSE' refer to the proposed Grover-based quantum classifier, the quantum kernel classifier with BCE loss, and the quantum kernel classifier with mean square error loss.

Algorithm's Name	$P = 0.1 \ (Train)$	$P = 0.1 \ (Test)$	$P = 0.3 \ (Train)$	$P = 0.3 \ (Test)$
BCE	0.883 ± 0.034	0.871 ± 0.071	0.924 ± 0.042	0.799 ± 0.061
MSE	0.941 ± 0.017	0.938 ± 0.008	0.929 ± 0.034	0.917 ± 0.011
BQM	0.977 ± 0.011	0.971 ± 0.007	0.951 ± 0.027	0.949 ± 0.010

4.4. Complex Comparsion Analysis

After receiving the encoded data in the variable component subcircuit of the quantum classifier, the measurement operator defines a query as one measurement. The iterative process of this algorithm is shown in Figure 8. According to the quantum classifier's training mechanism, calculating the total number of measurements for variable component subcircuits is comparable to the query complexity of obtaining the gradient in a time frame.
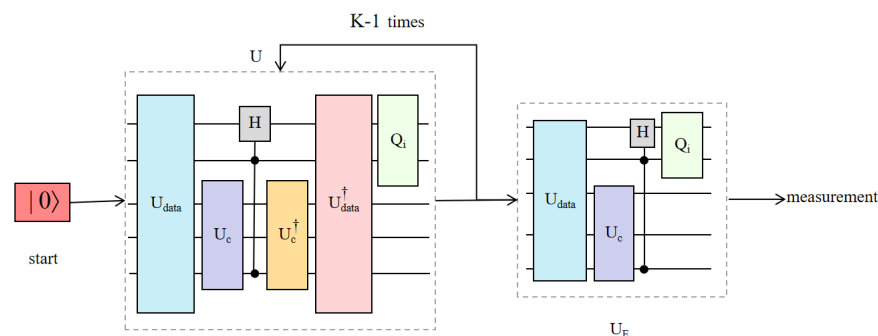
Figure 8. The iterated process of BQM algorithm. After receiving the encoded data in the variable component subcircuit of the quantum classifier, the measurement operator defines a query as one measurement.

The next step is to derive the number of measurements required for a quantum kernel classifier with BCE loss in one period. For the dataset *T*, the BCE loss is generated

$$L_{\text{BCE}} = -\frac{1}{N} \sum_{i=0}^{N-1} y_i \log(P(y_i)) + (1 - y_i) \log(1 - P(y_i)) \tag{35}$$

where y_i is the label of the ith example, and $P(y_i)$ is the prediction probability of the label y_i; the output of its quantum circuit is

$$P(y_i) = Tr((|1\rangle\langle 1|) \otimes H \otimes \nabla \tag{36}$$

where $\nabla_\theta = |i^*\rangle\langle i^*|)U_c(\theta)|0\rangle\langle 0|U_c(\theta)^\dagger$, $U_c(\theta)$ is defined in Equation (25), and $(|1\rangle\langle 1|) \otimes H \otimes (|i^*\rangle\langle i^*|) = \Pi$ is the measurement operator. According to the parameter displacement rule, the derivative of BCE loss is satisfied

$$\frac{\partial L_{\text{BCE}}}{\partial \theta_e} = \frac{1}{N} \sum_{i=0}^{N-1} \rho \frac{\text{Tr}(\Pi\nabla_{\theta_+}) - \text{Tr}(\Pi\nabla_{\theta_-})}{2} \tag{37}$$

where $\theta_\pm$ is defined in Equation (28), $\rho = \frac{1-y_i}{1-P(y_i)} - \frac{y_i}{P(y_i)}$. To obtain the gradient of BCE loss according to the above equation, we need to give each training example to the quantum kernel classifier to estimate $P(y_i)$, and then calculate the coefficient ρ.

BQM uses the superposition property of the loss function L defined in Equation (17) to obtain the gradient $\frac{\partial L}{\partial \theta_e}$. According to Equation (23), the gradient of BQM satisfies

$$\frac{\partial L(\theta, T_k)}{\partial \theta_e} = \frac{s}{2}\left(1 - y_k(\text{Tr}\left(\Pi\nabla_{\theta_+^{(k)}}\right) - \text{Tr}\left(\Pi\nabla_{\theta_-^{(k)}}\right))\right) \tag{38}$$

where y_k refers to the label of the last pair in the training example T_k. The gradient of T_k may be calculated using $2K$ measurements, where the first K measurement aims to approach $\text{Tr}\left(\Pi\nabla_{\theta_-^{(k)}}\right)$ and the last K measurement aims to approximate $\text{Tr}\left(\Pi\nabla_{\theta_+^{(k)}}\right)$, according to the equation above.

Complex comparison analysis determines the effect of the dataset size on the binary QNN model in this paper. The standard Grover search algorithm's search complexity is $O(\sqrt{Kl})$ for information data entries of size K and the total number of trainable parameters l. The optimal algorithm classification complexity value is $O(\sqrt{\frac{Kl}{MT^2}})$, as can be shown in the following Table 2. The reduced query complexity of BQM implies a potential quantum advantage in completing the classification task.

Table 2. Query complexity in several algorithms. The notations T, K, M, and l refer to the batch size range, the wide variety of measurements used to estimate quantum expectation value, the complete variety of education examples, and the total number of trainable parameters.

Algorithm's Name	Query Complexity
Grover	$O(\sqrt{Kl})$
Younes' algorithm	$O(\sqrt{\frac{Kl}{M}})$
BCE	$O(KMl)$
MSE	$O(KMl)$
BQM	$O(\sqrt{\frac{Kl}{MT^2}})$

5. Conclusions

As an essential source of information for value-added social media websites, user behavior patterns are the key to fast and accurate identification through session division to realize extensive data analysis. In this paper, based on the trial-and-error method of the Grover search algorithm, combined with the binary classification task of QNN supervised learning, the advantages of the Grover quantum algorithm are brought into play in the quantum classifier of the quantum neural network. The data are preprocessed by analyzing the network search data to realize the construction of the BQM algorithm. They lay the foundation for the development of user behavior pattern prediction. The experimental data show that the application effect of this algorithm has a more apparent accurate recognition rate than the other two classifiers, and it still has a prominent effect in the depolarized noise environment. It can play a supervisory role in the security detection of future users' network search behaviors.

References

1. Bao, Z.; Zhou, J.; Tay, Y.C. sonSQL: An Extensible Relational DBMS for Social Network Start-Ups. In Proceedings of the 32nd International Conference on Conceptual Modeling (ER 2013), Hong Kong, China, 11–13 November 2013; Volume 8217, pp. 495–498.
2. Rosen, D.; Kim, B. Social networks and online environments: When science and practice co-evolve. *Soc. Netw. Anal. Min.* **2011**, *1*, 27–42. [CrossRef]
3. Zhang, Y.; Ling, W. A comparative study of information diffusion in weblogs and microblogs based on social network analysis. *Chin. J. Libr. Inf. Sci.* **2012**, *5*, 51–66.
4. Xu, L.; Jiang, C.; Wang, J.; Yuan, J.; Ren, Y. Information Security in Big Data: Privacy and Data Mining. *Chin. J. Libr. Inf. Sci.* **2014**, *2*, 2169–3536.
5. Witten, I.; EibeFrank. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*, 1st ed.; China Machine Press: Beijing, China, 2005.
6. Nasraoui, O. Web data mining: Exploring hyperlinks, contents, and usage data. *SIGKDD Explor. Newsl.* **2008**, *10*, 23–25. [CrossRef]
7. Tang, H.S.; Yao, Y.W. Research on Decision Tree in Data Mining. *Appl. Res. Comput.* **2001**, *8*, 18.
8. Chen, M.S.; Park, J.S.; Yu, P.S. Efficient data mining for path traversal patterns. *IEEE Trans. Knowl. Data Eng.* **1998**, *10*, 209–221. [CrossRef]
9. Juntao, H.U.; Defeng, W.U.; Li, G.; Gan, Y. Architecture and Technique of Multimedia Data Mining. *Comput. Eng.* **2003**, *29*, 149–151.
10. Shou, Y.J. Data Mining Technique. *Autom.-Petro-Chem. Ind.* **2000**, *6*, 38.
11. Yang, Y.; Adelstein, S.J.; Kassis, A.I. Target discovery from data mining approaches. *Drug Discov. Today* **2009**, *14*, 147–154. [CrossRef]
12. Mahmood, N.; Hafeez, Y.; Iqbal, K.; Hussain, S.; Aqib, M.; Jamal, M.; Song, O.Y. Mining Software Repository for Cleaning Bugs Using Data Mining Technique. *Comput. Mater. Contin.* **2021**, *69*, 873–893. [CrossRef]
13. Helma, C.; Kramer, S. Machine Learning and Data Mining. *Predict. Toxicol.* **2005**, *42*, 223–254.
14. Wang, Y.; Li, G.; Xu, Y.; Hu, J. An Algorithm for Mining of Association Rules for the Information Communication Network Alarms Based on Swarm Intelligence. *Math. Probl. Eng.* **2014**, *2014*, 894205. [CrossRef]
15. Chen, Z.Y.; Qiu, T.H.; Zhang, W.B.; Ma, H.Y. Effects of initial states on the quantum correlations in the generalized Grover search algorithm. *Chin. Phys. B* **2021**, *30*, 080303. [CrossRef]
16. Xu, Z.; Ying, M.; Valiron, B. Reasoning about Recursive Quantum Programs. *arXiv* **2021**, arXiv:2107.11679.
17. Xue, Y.J.; Wang, H.W.; Tian, Y.B.; Wang, Y.N.; Wang, Y.X.; Wang, S.M. Quantum Information Protection Scheme Based on Reinforcement Learning for Periodic Surface Codes. *Quantum Eng.* **2022**, *2022*, 7643871. [CrossRef]
18. Galindo, A.; Martín-Delgado, M.A. Family of Grover's quantum-searching algorithms. *Phys. Rev. A* **2000**, *62*, 062303. [CrossRef]
19. Ma, H.; Ma, Y.; Zhang, W.; Zhao, X.; Chu, P. Development of Video Encryption Scheme Based on Quantum Controlled Dense Coding Using GHZ State for Smart Home Scenario. *Wirel. Pers. Commun.* **2022**, *123*, 295–309. [CrossRef]
20. Zhu, D.; Linke, N.M.; Benedetti, M.; Landsman, K.A.; Nguyen, N.H.; Alderete, C.H.; Perdomo-Ortiz, A.; Korda, N.; Garfoot, A.; Brecque, C.; et al. Training of quantum circuits on a hybrid quantum computer. *Sci. Adv.* **2019**, *5*, aaw9918. [CrossRef] [PubMed]
21. Wang, H.W.; Xue, Y.J.; Ma, Y.L.; Hua, N.; Ma, H.Y. Determination of quantum toric error correction code threshold using convolutional neural network decoders. *Chin. Phys. B* **2022**, *31*, 010303. [CrossRef]
22. Zheng, W.; Yin, L. Characterization inference based on joint-optimization of multi-layer semantics and deep fusion matching network. *Peerj. Comput. Sci.* **2022**, *8*, e908. [CrossRef]
23. Singh, M.P.; Radhey, K.; Rajput, B.S. Pattern Classifications Using Grover's and Ventura's Algorithms in a Two-qubits System. *Int. J. Theor. Phys.* **2018**, *57*, 692–705. [CrossRef]
24. Huang, C.Q.; Jiang, F.; Huang, Q.H.; Wang, X.Z.; Han, Z.M.; Huang, W.Y. Dual-Graph Attention Convolution Network for 3-D Point Cloud Classification. *IEEE Trans. Neural Networks Learn. Syst.* **2022**, 1–13. [CrossRef] [PubMed]

25. Ding, L.; Wang, H.; Wang, Y.; Wang, S. Based on Quantum Topological Stabilizer Color Code Morphism Neural Network Decoder. *Quantum Eng.* **2022**,*2022*. [CrossRef]
26. Zhou, G.; Zhang, R.; Huang, S. Generalized Buffering Algorithm. *IEEE Access* **2021**, *99*, 1. [CrossRef]
27. Xu, Y.; Zhang, X.; Gai, H. Quantum Neural Networks for Face Recognition Classifier. *Procedia Eng.* **2011**, *15*, 1319–1323. [CrossRef]
28. Zhang, Y.C.; Bao, W.S.; Xiang, W.; Fu, X.Q. Decoherence in optimized quantum random-walk search algorithm. *Chin. Phys.* **2015**, *24*, 197–202. [CrossRef]
29. Tseng, H.-Y.; Tsai, C.-W.; Hwang, T.; Li, C.-M. Quantum Secret Sharing Based on Quantum Search Algorithm. *Int. J. Theor. Phys.* **2012**, *51*, 3101–3108. [CrossRef]
30. Chen, J.; Liu, L.; Liu, Y.; Zeng, X. A Learning Framework for n-Bit Quantized Neural Networks toward FPGAs. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *32*, 1067–1081. [CrossRef]
31. Xiao, Y.; Huang, W.; Oh, S.K.; Zhu, L. A polynomial kernel neural network classifier based on random sampling and information gain. *Appl. Intell.* **2021**, *52*, 6398–6412. [CrossRef]
32. He, Z.; Zhang, H.; Zhao, J.; Qian, Q. Classification of power quality disturbances using quantum neural network and DS evidence fusion. *Eur. Trans. Electr. Power* **2012**, *22*, 533–547. [CrossRef]
33. Du, Y.; Hsieh, M.H.; Liu, T.; Tao, D. A Grover-search Based Quantum Learning Scheme for Classification. *New J. Phys.* **2021**, *23*, 20. [CrossRef]
34. Younes, A.; Rowe, J.; Miller, J. Enhanced quantum searching via entanglement and partial diffusion. *Phys. Nonlinear Phenom.* **2008**, *237*, 1074–1078. [CrossRef]
35. Liang, X.; Luo, L.; Hu, S.; Li, Y. Mapping the knowledge frontiers and evolution of decision making based on agent-based modeling. *Knowl. Based Syst.* **2022**, *250*, 108982. [CrossRef]
36. Liu, Z.; Chen, H. Analyzing and Improving the Secure Quantum Dialogue Protocol Based on Four-Qubit Cluster State. *Int. J. Theor. Phys.* **2020**, *59*, 2120–2126. [CrossRef]
37. Chakrabarty, I.; Khan, S.; Singh, V. Dynamic Grover search: Applications in recommendation systems and optimization problems. *Quantum Inf. Process.* **2017**, *16*, 153. [CrossRef]
38. Song, J.; Ke, Z.; Zhang, W.; Ma, Y.; Ma, H. Quantum Confidentiality Query Protocol Based on Bell State Identity. *Int. J. Theor. Phys.* **2022**,*61*, 52. [CrossRef]
39. Panella, M.; Martinelli, G. Neural networks with quantum architecture and quantum learning. *Int. J. Circuit Theory Appl.* **2011**, *39*, 61–77. [CrossRef]
40. Carlo, C.; Stefano, M. On Grover's search algorithm from a quantum information geometry viewpoint. *Phys. Stat. Mech. Its Appl.* **2012**, *391*, 1610–1625.
41. Ashraf, I.; Mahmood, T.; Lakshminarayanan, V. A Modification of Grover's Quantum Search Algorithm. *Photonics Optoelectron* **2012**, *1*, 20–24.
42. Long, G.; Liu, Y.; Loss, D. Search an unsorted database with quantum mechanics. *Front. Comput. Sci. China* **2007**, *1*, 247–271. [CrossRef]
43. Bin, C.; Zhang, W.; Wang, X.; Zhao, J.; Gu, Y.; Zhang, Y. A Memetic Algorithm Based on Two_Arch2 for Multi-depot Heterogeneous-vehicle Capacitated Arc Routing Problem. *Swarm Evol. Comput.* **2021**, *63*, 100864.
44. Ma, Z.; Zheng, W.; Chen, X.; Yin, L. Joint embedding VQA model based on dynamic word vector. *Peerj Comput. Sci.* **2021**, *7*, e353. [CrossRef] [PubMed]
45. Grover, L.K. A fast quantum mechanical algorithm for database search. In Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing; Association for Computing Machinery, New York, NY, USA, 3–5 May 1996; pp. 212–219.
46. Benedetti, M.; Garcia-Pintos, D.; Perdomo, O.; Leyton-Ortega, V.; Nam, Y.; Perdomo-Ortiz, A. A generative modeling approach for benchmarking and training shallow quantum circuits. *npj Quantum Inf.* **2019**, *5*, 45. [CrossRef]
47. Mitarai, K.; Negoro, M.; Kitagawa, M.; Fujii, K. Quantum circuit learning. *Phys. Rev. A* **2018**, *98*, 2309. [CrossRef]
48. Zhang, Y.J.; Tang, J. Analysis and assessment of network security situation based on cloud model. *Int. J. Theor. Phys.* **2014**, *36*, 63–67.
49. Tian, H.; Qin, Y.; Niu, Z.; Wang, L.; Ge, S. Summer Maize Mapping by Compositing Time Series Sentinel-1A Imagery Based on Crop Growth Cycles. *J. Indian Soc. Remote. Sens.* **2021**, *49*, 2863–2874. [CrossRef]
50. Qin, Y. Early-Season Mapping of Winter Crops Using Sentinel-2 Optical Imagery. *Remote. Sens.* **2021**, *13*, 3822.
51. Zheng, W.; Xun, Y.; Wu, X.; Deng, Z.; Chen, X.; Sui, Y. A Comparative Study of Class Rebalancing Methods for Security Bug Report Classification. *IEEE Trans. Reliab.* **2021**, *70*, 4. [CrossRef]
52. Wu, X.; Zheng, W.; Xia, X.; Lo, D. Data quality matters: A case study on data label correctness for security bug report prediction. *IEEE Trans. Softw. Eng.* **2021**, *48*, 2541–2556 [CrossRef]
53. Aghayar, K.; Rezaei Fard, E. Noisy Quantum Mixed State Pattern Classification Based on the Grover's Search Algorithm. *Int. J. Nanotechnol. Appl.* **2021**, *15*, 171–180.