CDQKL: Consensus-based Distributed Quantum Kernel Learning

Wenxuan Ma, Mengxiang Liu, and Ruilong Deng

***Abstract*—The field of quantum computing has developed rapidly in recent years due to its promising trend of surpassing traditional machine learning in terms of speed and effectiveness. Quantum kernel learning is one of the paradigms of quantum machine learning, but the training of quantum kernel is time consuming. Therefore, this work makes the first attempt to introduce a consensus-based distributed approach to quantum kernel learning - named CDQKL - that only requires to exchange model parameter information between adjacent nodes while avoiding the need of sharing local training data. Through comparative experimental studies, the advantages of CDQKL in classification accuracy and convergence speed are verified. Considering the popularization of quantum computing cloud service and miniaturization of quantum terminals, the CDQKL adapting to this trend is able to play a vital role in data security, which implies the far-reaching significance of this work. Our code is available at https://github.com/Leisurivan/CDQKL.**

***Index Terms*—Quantum kernel learning, distributed approach, consensus, classification accuracy, convergence speed**

I. Introduction

Using the properties of quantum mechanics such as entanglement and superposition, quantum computing can speedup the solution of certain problems. Through quantum computing, we can sometimes achieve exponential speedup, which has already been confirmed in the recent study. Leveraging optimization and machine learning techniques, variational quantum algorithms have become the primary strategy for demonstrating the quantum advantage at present [1]. Although machine learning has illustrated great successes in numerous scenarios like computer vision, with the continuous expansion of data sets and the end of Moore's Law, the development of machine learning is rapidly approaching the limits of classical computing models [2]. Quantum computing will be a promising solution to this bottleneck problem. Additionally, with the provable separation between classical and quantum learnability, the quantum-based computational paradigms can hopefully provide unprecedented advancements for machine learning [2].

Kernel learning is a fundamental class of algorithms for pattern analysis that involve using linear classifiers to solve nonlinear problems [3]. Different from deep neural networks, the model generated from kernel learning has solid theoretical foundations for both performance and interpretability. Kernel learning also has a corresponding quantum version and has become one of the paradigms of quantum machine learning [4]. Compared with traditional machine learning, quantum kernel learning (QKL) has demonstrated the quantum advantages in speed [5] and expression capability [6] by mapping original features to the quantum state Hilbert space. However, on the other hand, the training of quantum kernel is time-consuming. Therefore, in this paper, we make the first attempt to introduce a consensus-based distributed approach to quantum kernel learning - named CDQKL - that only needs to exchange model parameter information between adjacent nodes while not requiring to share local training data. The motivations are pictured in Fig. 1 and explained in subsequent parts.

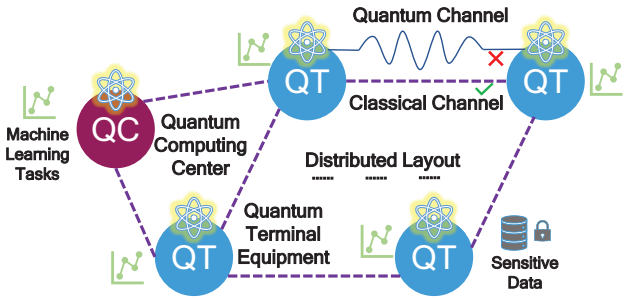

Fig. 1. A brief schematic diagram of the motivations for this paper. The motivation comes from many aspects, taking into account both current limitations and future development trends.

Time-consuming training for centralized QKL. Quantum kernel alignment, derived from the traditional kernel alignment, is an effective variational parameter optimization method to improve the model prediction accuracy on a given dataset [7]. However, in each quantum kernel alignment during the training stage, a kernel matrix with a size quadratic in the scale of the dataset needs to be constructed, which will result in time consuming and may be unacceptable for large-scale tasks [8]. Based on established experiences in traditional machine learning, if the training tasks are distributed, the training process will be significantly accelerated through the distributed computation mechanism [9].

Development trends of ubiquitous quantum equipment. Currently, quantum computing is offered as a cloud-based service from providers such as IBM Quantum, Amazon Bracket, Microsoft Azure [10] and etc. In recent times, quantum computers have also been miniaturized, and there are even

This work was supported in part by the National Natural Science Foundation of China under Grant 62073285, 62293503, 62293502, 62293500, in part by the Natural Science Foundation of Zhejiang Province under Grant LR23F030001, in part by the Xiaomi Foundation, in part by the Fundamental Research Funds for the Central Universities (Zhejiang University NGICS Platform), and in part by the Key Lab of CS&AUS of Zhejiang Province.

W. Ma and R. Deng are with the State Key Laboratory of Industrial Control Technology and the College of Control Science and Engineering, Zhejiang University, Hangzhou 310027, China (emails: {suremarvin, dengruilong}@zju.edu.cn).

M. Liu is with the Department of Automatic Control and Systems Engineering, University of Sheffield, Sheffield, UK (email: mengxiang.liu@sheffield.ac.uk).

979-8-3503-6677-8/24/$31.00 ©2024 IEEE
DOI 10.1109/QCNC62729.2024.00015

mobile versions of quantum computers entering the market[1]. In the foreseeable future, quantum equipment will also present a similar pattern to the current classical one. Therefore, when conducting quantum machine learning in the future, we will naturally face a scenario where data is distributed in quantum terminals and the aggregation or sharing of data is challenging. There will also be scenarios where the amount of data is too large to be processed on a single quantum terminal. At that time, the centralized quantum machine learning will not be able to meet the performance requirements and thus call for the development of distributed learning approaches.

Data security benefit from distributed approach. In addition to adding parallelism to improve the training speed of quantum machine learning, distributed quantum machine learning can also avoid the privacy or sensitivity issues without sharing training data [11]. Especially considering that information such as facial images, location-based services, health or personal financial status are related to personal or organizational data security, the value of distributed quantum machine learning is even more prominent [12].

The contributions of this paper are summarised as follows.

- To the best of our knowledge, we are the first to propose CDQKL — a consensus-based distributed quantum kernel learning approach that only needs to exchange model parameter information between adjacent nodes while not requiring to share local training data.
- Through extensive experiments, the superiority of our proposed CDQKL compared with the centralized QKL, local QKL, as well as traditional kernel classifier in terms of classification accuracy and convergence speed is well demonstrated.
- To handle the real-world vision dataset, a quantum-classical hybrid architecture based on autoencoder is established and the experiment results demonstrate not only the training speedup of CDQKL over centralized QKL but also its potential advantage in helping pass through the barren plateau.

The remainder of this paper is organized as follows. Section II reviews the related works. In Section III, we introduce the preliminaries of kernel learning and quantum computing. Section IV proposes CDQKL with theoretical analysis. The experiment results are shown in Section V. Section VI concludes this paper with future works.

II. Related Works

Recently, in order to solve the computational challenges of quantum kernel alignment, the Pegasos algorithm has been extended to the quantum case, which accelerates the process of kernel alignment through stochastic gradient descent [13]. A sub-sampling training approach has also been introduced recently, which can reduce training costs in a manner similar to batch gradient descent [14]. The ideas for these methods all come from what have been done with traditional machine learning. Distributed approaches can also be used to further address the computational challenge.

[1]https://www.spinq.cn/products

There have been some studies on distributed quantum machine learning, with a majority of efforts on leveraging the properties of quantum communication to enhance the efficiency and security of traditional machine learning. For instance, a quantum counting-based communication algorithm has been introduced to improve the efficiency of distributed communication in scenarios such as distributed least squares fitting and softmax regression [15]; A distributed secure quantum machine learning protocol has been proposed to safeguard private data [16]; Shared entangled states were utilized to facilitate secure distributed quantum computing and quantum machine learning in collaborative multi-party settings [17]. However, quantum interconnection technologies face significant challenges due to the fact that quantum information is extremely fragile [18]. Nevertheless, QKL enables end-to-end learning through embedding and measurement. This provides us with an opportunity to interconnect quantum terminals using classic connection methods, as shown in Fig. 1, to obtain quantum advantage in the sooner future [19].

Additionally, there are also endeavors that integrated quantum neural networks (QNN) into federated learning, establishing a dynamic quantum federated learning framework by exploiting the inherent properties of QNN [20]. Another work combined QNN with pre-trained convolutional neural networks, constructing a hybrid quantum-classical federated learning model [21]. These efforts focus more on the protection of data privacy. In fact, distributed optimization methods for solving large-scale machine learning have been extensively studied in the past decades [22]. Consensus-based distributed stochastic gradient descent has emerged as a simple, effective, and more general approach [23]. In contrast to the potential issues of communication bottlenecks on the manager node that may lead to the failure of all connected machines in the case of federated learning with a manager-worker architecture, fully distributed learning network could circumvent potential failures and communication delays [24].

III. Preliminaries

A. Kernel Learning

Kernel learning has been widely used in the training phase of support vector machine (SVM). For linearly separable data points $\boldsymbol{x}$ and their labels y, where $\boldsymbol{x} \in D \subset R^n$, $y \in \{1, -1\}$, the task of SVM is to find a hyperplane to separate the data points $\boldsymbol{x}$ corresponding to different labels y. That is determining $\boldsymbol{w} \in R^n$, $b \in R$, and then obtaining the plane,

$$f(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{x} + b. \tag{1}$$

Therefore, the category to which the data point $\boldsymbol{x}$ belongs can be determined based on $\text{sgn}(f(\boldsymbol{x}))$.

When the data points are difficult to distinguish using planes in the existing space, we can define a mapping function $\varphi : R^n \to S$ where S is a new space. The data points x can be mapped to the data points $\varphi(\boldsymbol{x})$, so that in the new space we can find a separable hyperplane. That is determining $\boldsymbol{w}_S$ and b_S, and then obtaining the plane in S,

$$f_S(\boldsymbol{x}) = \boldsymbol{w}_S^T \varphi(\boldsymbol{x}) + b_S. \tag{2}$$

Therefore, the category to which the data point $\boldsymbol{x}$ belongs can be determined based on $\text{sgn}(f_S(\boldsymbol{x}))$.

In the process of solving Eq.(2) using the dual method involves calculating $\varphi^T(\boldsymbol{x}_i)\varphi(\boldsymbol{x}_j)$. Since the dimensions of S space may be very large, it is difficult to calculate them directly. In order to avoid this obstacle, when S is a linear space, we can define the kernel function $k : R^n \times R^n \to R$,

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \langle \varphi(\boldsymbol{x}_i), \varphi(\boldsymbol{x}_j) \rangle = \varphi^T(\boldsymbol{x}_i)\varphi(\boldsymbol{x}_j). \tag{3}$$

Thus we can calculate the inner product between data points $\varphi(\boldsymbol{x})$ through the function $k(\cdot, \cdot)$ without having to calculate it directly in linear space S.

According to the representer theorem [25], $\boldsymbol{w}_S$ can be expressed as a linear combination of data points,

$$\boldsymbol{w}_S = \sum_i c_i \varphi(\boldsymbol{x}_i). \tag{4}$$

So the decision plane Eq.(2) can be written as

$$f_S(\boldsymbol{x}) = \sum_i c_i \varphi^T(\boldsymbol{x}_i)\varphi(\boldsymbol{x}) + b_S = \sum_i c_i k(\boldsymbol{x}_i, \boldsymbol{x}) + b_S. \tag{5}$$

In other words, in addition to kernel learning that can be used in the training phase of SVM, the prediction in the application phase can still circumvent the calculation of the inner product in space S. This is called kernel trick. In fact, given any mapping function φ, a kernel function k can be determined. In turn, as long as the kernel matrix $K = [k(\boldsymbol{x}_i, \boldsymbol{x}_j)]_{m \times m}$,where m represents the size of the sample space, associated with each kernel function $k(\cdot, \cdot)$, is positive semi-definite, it corresponds to a mapping function φ [25]. There are many commonly used kernel functions. For example, the Gaussian kernel can be expressed as

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = e^{-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{2\sigma^2}}. \tag{6}$$

B. Quantum Computing

In contrast to a classical bit, a qubit in quantum computing can represent not only the state $|0\rangle$ or $|1\rangle$ but also superposition states like

$$|\varphi\rangle = \alpha|0\rangle + \beta|1\rangle. \tag{7}$$

Here, α and β carry physical meaning, and $|\alpha|^2$ and $|\beta|^2$ respectively represent the probabilities of obtaining states $|0\rangle$ and $|1\rangle$ when measured along the z-axis.

In general, any n-qubit quantum state can be represented by a vector in the Hilbert space as

$$|\varphi\rangle = \sum_{k=1}^{2^n} \alpha_k |k\rangle. \tag{8}$$

Here, $|k\rangle$ represents the k^{th} computational basis, and α_k satisfies the probability condition, i.e., $\sum_{k=1}^{2^n} |\alpha_k|^2 = 1$.

Similar to logical circuits for classical bits, the fundamental computation routine in quantum computing is quantum circuits such as those in Fig. 2. However, unlike classical gates applied to classical bits, operations on qubits are general unitary gates. Variational quantum circuits are ordered sequences of parameterized quantum gates on quantum states. These parameters can be data points to be embedded or optimized parameters to achieve specific optimization goals [26].

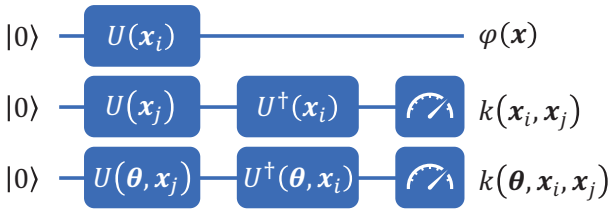

Fig. 2. Quantum circuits to obtain $\varphi(\boldsymbol{x})$, $k(\boldsymbol{x}_i, \boldsymbol{x}_j)$ and $k(\boldsymbol{\theta}, \boldsymbol{x}_i, \boldsymbol{x}_j)$.

IV. Methodology

A. Quantum Kernel Learning

The definition of QKL is analogous to the classical one. Let $U(\boldsymbol{x})$ be an unitary operation on the qubits. Then, a classical data point x can be transformed into the corresponding quantum state $\varphi(\boldsymbol{x})$ through the quantum feature mapping as illustrated in the Fig. 2. Here $\varphi(\boldsymbol{x})$ can be written as

$$\varphi(\boldsymbol{x}) = U(\boldsymbol{x})|0\rangle. \tag{9}$$

The inner product of two quantum states can be used to define the quantum kernel function $k(\cdot, \cdot)$. The overlap of the relevant quantum states can be calculated through the quantum circuit as shown in the Fig. 2. So $k(\cdot, \cdot)$ can be written as

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = |\langle \varphi(\boldsymbol{x}_i) | \varphi(\boldsymbol{x}_j) \rangle|^2 = |\langle 0|U^\dagger(\boldsymbol{x}_i)U(\boldsymbol{x}_j)|0\rangle|^2, \tag{10}$$

and it can be approximated by the frequency of the ground state through multiple measurements.

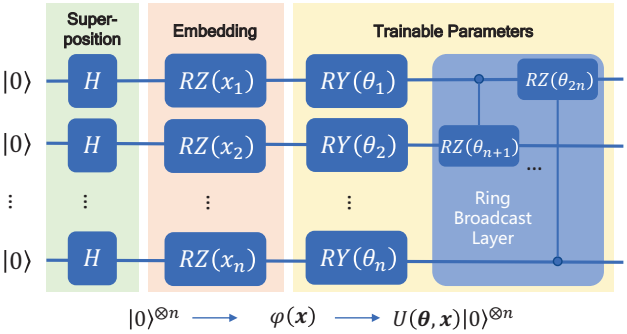

Fig. 3. A quantum feature mapping circuit with variational parameters.

By introducing variational parameters into the quantum feature mapping, we can optimize the kernel function on a specific dataset to enhance its performance. An example unitary layer with variational parameters is shown as Fig. 3. The circuit can be sequentially divided into a superposition layer, an embedding layer and a trainable parameter layer. The superposition layer applies Hardman gate to each qubit to build a superposition state. The embedding layer applies a z-axis rotation with an angle of x_i to the i^{th} qubit thereby embedding the information of $\boldsymbol{x}$ into the working system. The trainable parameter layer includes the y-axis rotation applied to each qubit and a ring broadcast layer in order to create entanglement and broadcast information. The kernel function with variational parameters $\boldsymbol{\theta} \in R^n$ can be expressed as

$$k(\boldsymbol{\theta}, \boldsymbol{x}_i, \boldsymbol{x}_j) = |\langle 0|U^\dagger(\boldsymbol{\theta}, \boldsymbol{x}_i)U(\boldsymbol{\theta}, \boldsymbol{x}_j)|0\rangle|^2. \tag{11}$$

An ideal kernel function on a given dataset can be written as

$$k^*(\boldsymbol{\theta}^*, \boldsymbol{x}_i, \boldsymbol{x}_j) = y_i y_j. \tag{12}$$

This implies that the output is 1 when $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ belong to the same class and -1 when they belong to different classes. Since the kernel matrix is used during the actual training phase, the ideal kernel matrix can be represented as

$$K^* = [k^*(\boldsymbol{\theta}^*, \boldsymbol{x}_i, \boldsymbol{x}_j)]_{m\times m} = \boldsymbol{y}\boldsymbol{y}^T. \tag{13}$$

To measure the similarity between $K(\boldsymbol{\theta})$ and K^*, their alignment value can be calculated as

$$\begin{aligned} V(K(\boldsymbol{\theta}), K^*) &= \frac{\langle K(\boldsymbol{\theta}), K^*\rangle_F}{\sqrt{\langle K(\boldsymbol{\theta}), K(\boldsymbol{\theta})\rangle_F \langle K^*, K^*\rangle_F}} \\ &= \frac{\sum_{i,j} y_i y_j k(\boldsymbol{\theta}, \boldsymbol{x}_i, \boldsymbol{x}_j)}{m\sqrt{\sum_{i,j} k^2(\boldsymbol{\theta}, \boldsymbol{x}_i, \boldsymbol{x}_j)}}. \end{aligned} \tag{14}$$

where $\langle\cdot,\cdot\rangle_F$ represents the Frobenius inner product of two matrices. Intuitively, when the labels of data points are predicted correctly, $V(K(\boldsymbol{\theta}), K^*)$ increases, and when the labels are predicted incorrectly, $V(K(\boldsymbol{\theta}), K^*)$ decreases. Therefore, the objective function for quantum kernel alignment is

$$\min_{\boldsymbol{\theta}\in R^n} L(D, \boldsymbol{\theta}) = -V(K(\boldsymbol{\theta}), K^*). \tag{15}$$

Upon obtaining gradients through methods like parameter shift [27], one can utilize gradient descent to ascertain the optimal parameters θ^*.

B. Consensus-based Distributed Quantum Kernel Learning

In the considered scenario, we have N quantum computing units connected through classical channels, described by the graph $G(V, E)$, where $V = \{1, \ldots, N\}$ represents the quantum computing units in the network, and $E \subseteq V \times V$ denotes the communication links between quantum computing units. If $(i, j) \in E$, then quantum computing unit j is referred to as the neighbor of quantum computing unit i, and $j \in N_i := \{j \,|\, (i, j) \in E, \forall j \in V\} \cup \{i\}$. Additionally, we define the consensus matrix $W = [w_{ij}]_{N\times N}$, where $w_{ij} > 0$ if $(i, j) \in E$, otherwise $w_{ij} = 0$.

In this paper, we consider a connected graph G, and a doubly stochastic matrix W, which satisfies $W1_N = 1_N$ and $1_N^T W = 1_N^T$. Doubly stochastic matrices have many desirable properties, such as a spectral radius of 1, and more detailed information can be found in Ref. [28].

In the considered scenario, each quantum computing unit has a local dataset. The goal of the network is to minimize the loss function over the entire dataset, given by

$$\begin{aligned} \min_{\theta\in\mathbb{R}^n} L(D, \theta) &= \sum_{i=1}^{N} L_i(D_i, \theta) \\ &= -\sum_{i=1}^{N} V_i(K_i(\theta), K_i^*). \end{aligned} \tag{16}$$

where D and D_i represent the entire dataset and the sub-dataset residing in the i-th quantum computing unit, respectively. Eq.(16) represents the distributed formulation of QKL.

For each quantum computing unit $i \in V$, a gradient-based distributed algorithm can be designed as

$$\lambda_i^k = \sum_{j\in N_i} w_{ij}\theta_j^{k-1}, \tag{17}$$

$$\theta_i^k = \lambda_i^k - \eta_i \nabla_\theta L_i(D_i, \lambda_i^k). \tag{18}$$

where λ_i^k is an intermediate auxiliary variable, and η_i is the step size for each iteration. Intuitively, λ_i^k comes from the aggregation of gradient estimates from multiple nodes, which is the so-called consensus.

The gradient of the loss function for the i-th node, denoted as $\nabla_\theta L_i(D_i, \lambda_i^k)$, can be represented using m_i data points s_i^p from dataset D_i as

$$\nabla_\theta L_i(D_i, \lambda_i^k) = \frac{1}{m_i}\sum_{p=1}^{m_i} \nabla_\theta L_p(s_i^p, \lambda_i^k). \tag{19}$$

To further speed up training, a sub-sampling approach can be employed, wherein the stochastic gradient of the loss function on a random subset of q_i samples from D_i is used to approximate the gradient of the loss function on dataset D_i. The stochastic gradient of the loss function can be calculated as

$$\nabla_\theta \widetilde{L}_i(D_i, \lambda_i^k) = \frac{1}{q_i}\sum_{p}^{q_i} \nabla_\theta L_p(s_i^p, \lambda_i^k). \tag{20}$$

So for each quantum computing unit $i \in V$, a distributed algorithm based on stochastic gradients can be designed as

$$\lambda_i^k = \sum_{j\in N_i} w_{ij}\theta_j^{k-1}, \tag{21}$$

$$\theta_i^k = \lambda_i^k - \eta_i \nabla_\theta \widetilde{L}_i(D_i, \lambda_i^k). \tag{22}$$

Now, we summarize the implementation structure of our proposed consensus-based distributed approach to the scenario. A detailed instruction of CDQKL algorithm is provided in **Algorithm 1**. In subsequent experiments, we will also see that the CDQKL algorithm can not only be used in distributed quantum computing scenarios, but can also be specifically used to accelerate QKL.

C. Theoretical Analysis

It is worth noting that Steps 9 to 12 in **Algorithm 1** are parallel, so compared to centralized QKL, more quantum computing units will be utilized and efficiency can be improved. Furthermore, since each quantum computing unit only utilizes a subset of the entire dataset for training, it also serves as a form of sub-sampling technique, which can reduce the number of quantum circuits to be constructed for training [14].

Regarding convergence, a possible theoretical explanation comes from [23], indicating that if the training parameters are chosen appropriately, **Algorithm 1** will converge. The specific proof involves regarding the stochastic gradient as the true gradient with random noise.

An additional practical challenge arises from finite sampling. This is due to the fact that $k(\theta, x_i, x_j)$ represents the expectation of measurement values, whereas in practice, the approximate value obtained by repeating measurements of the

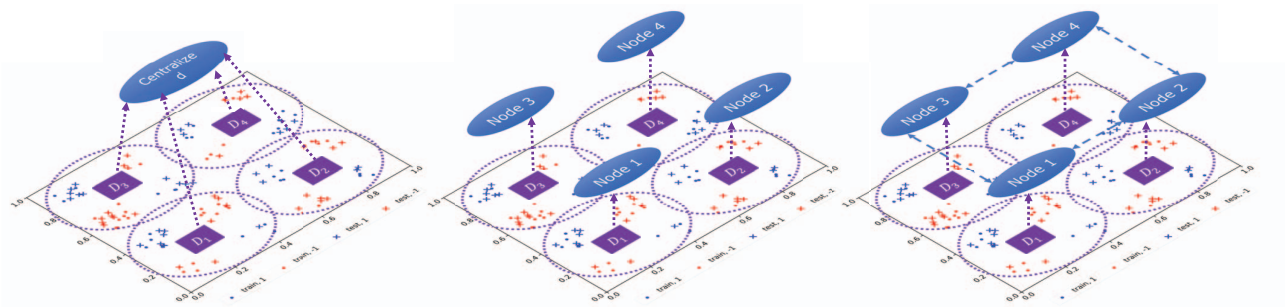

Fig. 4. Experiment settings in Section V-A. From left to right, they are the centralized, local, and consensus-based distributed training settings, respectively. For the first two cases, both quantum kernel and Gaussian kernel are employed for comparison.

Algorithm 1 CDQKL: Consensus-based Distributed Quantum Kernel Learning

Input: sub-dataset D_i for i-th quantum computing unit
for initialize $\forall i \in V$ **do**
  Establish connections with the neighbor quantum computing units in N_i
  Initialize the weight matrix W and determine the weight $w_{ij}, \forall j \in N_i$
  Initialize local variational quantum circuit parameters θ_i^0 and training step size η_i
end for
repeat
  Set $k := k + 1$
  for $\forall i \in V$ **do**
    Update intermediate variable λ_i^k to λ_i^{k+1} according to Eq.(21) by communicating with neighbors N_i
    Update variational quantum circuit parameters θ_i^k to θ_i^{k+1} according to Eq.(22)
  end for
until Termination condition is satisfied such as the norm of convergence gradient $\|g\| < g_{thresh}$ or iteration budget is approached and obtain $\boldsymbol{\theta}_i^e$ for $\forall i \in V$
for $\forall i \in V$ **do**
  Calculate the kernel matrix $K = [k(\boldsymbol{\theta}_i^e, \boldsymbol{x}_i, \boldsymbol{x}_j)]_{m\times m}$ according to Eq.(11)
  Solve the SVM problem using the dual method and obtain the decision plane Eq.(5)
end for

circuit depicted in Fig. 2 is used. However, by increasing the number of measurements, one can rapidly approximate the true expectation [8]. The remaining error may still be considered as random noise added to the true gradient, thereby not affecting the convergence of the training process.

V. Experiments

In this section, we will evaluate the performance of our proposed CDQKL algorithm through experiments on both artificial dataset [7] and real-world dataset. Experiments involve manual partitioning and random partitioning the dataset, revealing favorable performance of the algorithm in both cases. On the artificial dataset, we conduct a detailed comparison among our proposed CDQKL, the centralized QKL, local QKL, as well as traditional kernel classifier. For the real-world dataset, we employ a quantum-classical hybrid architecture, which first extracts features through an autoencoder and then trains the kernel using CDQKL, ultimately achieving satisfactory results. The specific implementation of the experiments are based on Pennylane framework [29]. The backend using to simulate the quantum computer is the Lightning of Pennylane and the method of differentiation to use is backprop.

A. Artificial Dataset

TABLE I
Experiment Results

Dataset	Experiment Setting	Score
Artificial (Checkerboard)	Centralized w/ Gaussian kernel	51.66%
	Centralized QKL	100%
	Local w/ Gaussian kernel	52.50% (avg.)
	Local QKL	75.42% (avg.)
	CDQKL	92.91% (avg.)
	CDQKL (whole data)	98.75% (avg.)

The experiment utilizes randomly generated checkerboard data within 1x1-sized regions, divided into a total of 16 checkerboard cells , each of which is a square of 0.25×0.25. Each cell contains Gaussian random data centered around the respective checkerboard. As illustrated in Fig. 4, the 16 checkerboard cells are manually grouped into four parts, each allocated to a quantum processing unit for training. In our proposed CDQKL, communication is allowed between adjacent nodes. In experiments, the consensus matrix W in **Algorithm 1** is set as

$$W = \begin{bmatrix} 1/3 & 1/3 & 1/3 & 0 \\ 0 & 1/3 & 1/3 & 1/3 \\ 1/3 & 0 & 1/3 & 1/3 \\ 1/3 & 1/3 & 0 & 1/3 \end{bmatrix}, \tag{23}$$

and the sample number in each iteration is 4.

Additionally, experiments are also conducted with the centralized QKL based on all data, local QKL without communication, and a comparison with a traditional classifier using Gaussian kernel. For QKL, the feature mapping circuit is a stack of 8 layers of the circuit as shown in the Fig. 3 and illustrated in Section IV. The number of the working qubits n is 5 and the embedding layer alternately embeds the binary features. That

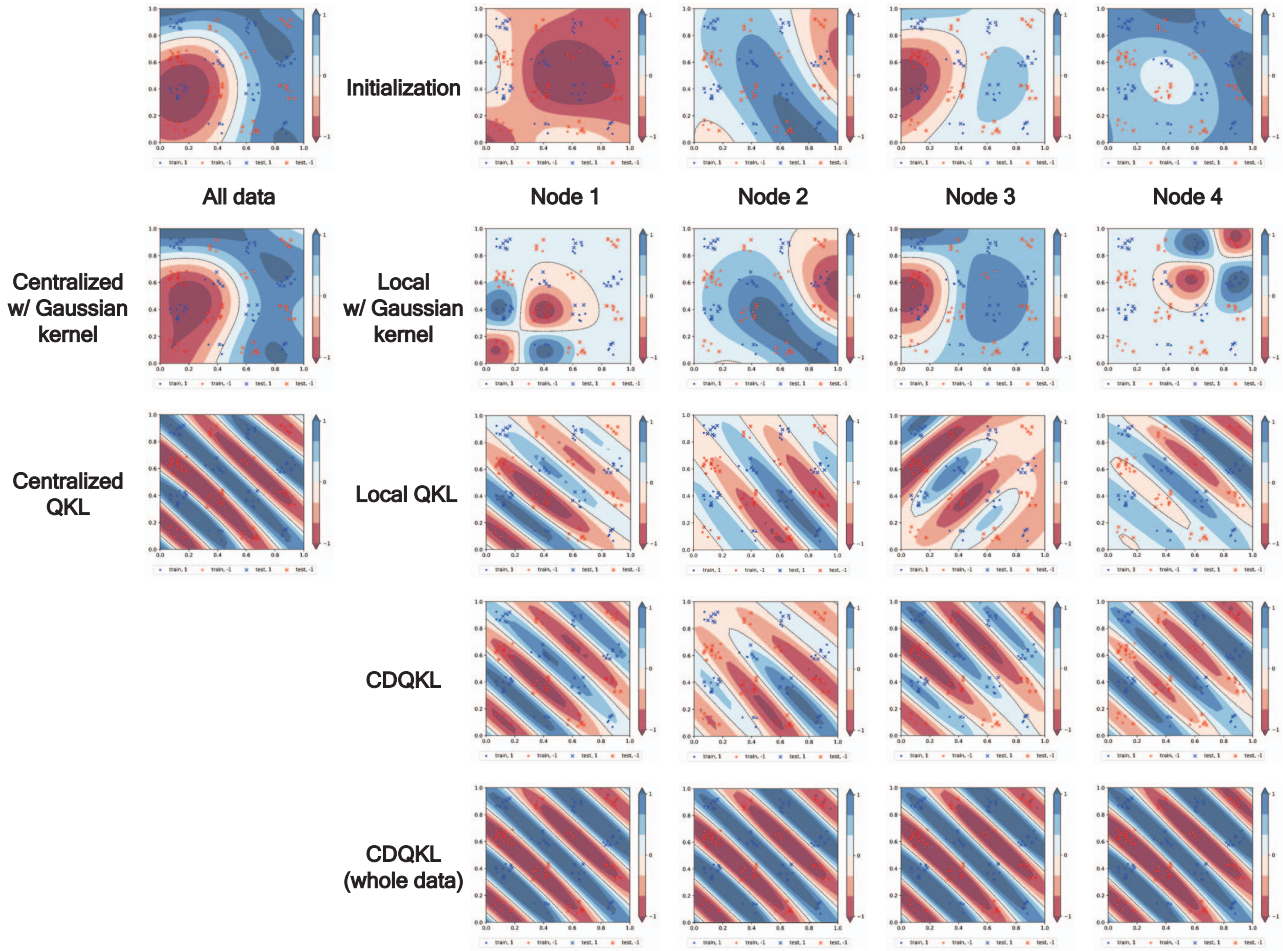

Fig. 5. Decision diagrams for different experimental settings. Every diagram is contoured and divided into different regions based on the decision results. Points in the blue area are considered to have the label of 1, while points in the red area are considered to have a label of -1, with darker colors indicating higher confidence. For each training method, a sufficient number of iterations are executed to achieve convergence. Through comparison, it can be found that QKL has good generalization to the checkerboard paradigm and CDQKL shows better performance than local QKL.

is, x_1 is embedded in the 1^{st}, 3^{rd}, and 5^{th} qubits, and x_2 is embedded in the 2^{nd} and 4^{th} qubits. For Gaussian kernel, σ in Eq. 6 is set to 0.5.

The results of different experimental settings are reflected through decision diagrams as shown in Fig. 5 and Tab. I. Firstly, we can observe the differences in expressive ability between quantum kernel and traditional kernel. Comparing classifiers obtained from the local QKL with those using traditional Gaussian kernel, it is found that the local QKL not only effectively learns its own dataset but also demonstrates excellent generalization performance for the entire dataset. This suggests that quantum kernel, by mapping data to quantum state space, can possess strong data representation capabilities, adequately capturing data features, especially in the case of checkerboard data.

In comparison of our proposed CDQKL with the local QKL, it is evident that, although nodes do not directly transmit data, they can leverage global data information. Thus, by utilizing only local data, they can still exhibit satisfactory discriminative performance on whole data. However, in terms of convergence speed (as shown in Fig. 6), our proposed CDQKL is slightly slower than the local QKL. Nevertheless, compared to the centralized QKL, our proposed CDQKL shows similar discriminative performance but with significantly reduced convergence time.

Considering that our proposed CDQKL can also be specifically used in parallel to increase the training speed, the last row in Fig. 5 also shows the image of the decision function which is from consensus-based distributed training but ultimately solving SVM using whole data. The result in Tab. I shows that the classification accuracy reaches 98.75%, which is very close to the results of centralized training.

B. Real-world Dataset

The MNIST dataset is a classic real-world dataset for handwritten digit recognition. To enable the quantum circuits in artificial dataset experiments to be effective, a quantum-classical hybrid framework based on an autoencoder is designed. As shown in Fig.7, digits 7 and 9, which are relatively challenging to distinguish, are selected. The encoder, which contains 4

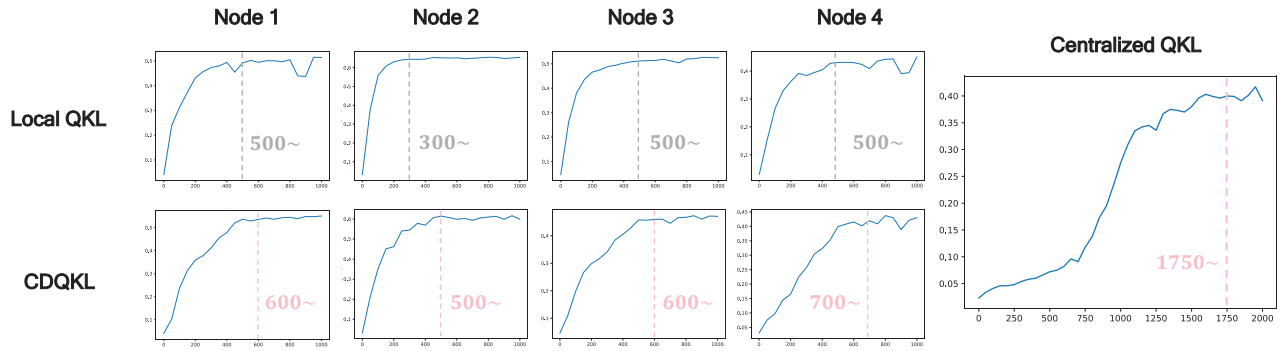

Fig. 6. Convergence curves for different experiment settings, where x-axis denotes iteration times and y-axis denotes kernel alignment value. The approximate number of convergence rounds is shown in the figure. It can be observed that our proposed CDQKL is slightly slower than the local QKL but much faster than the centralized QKL.

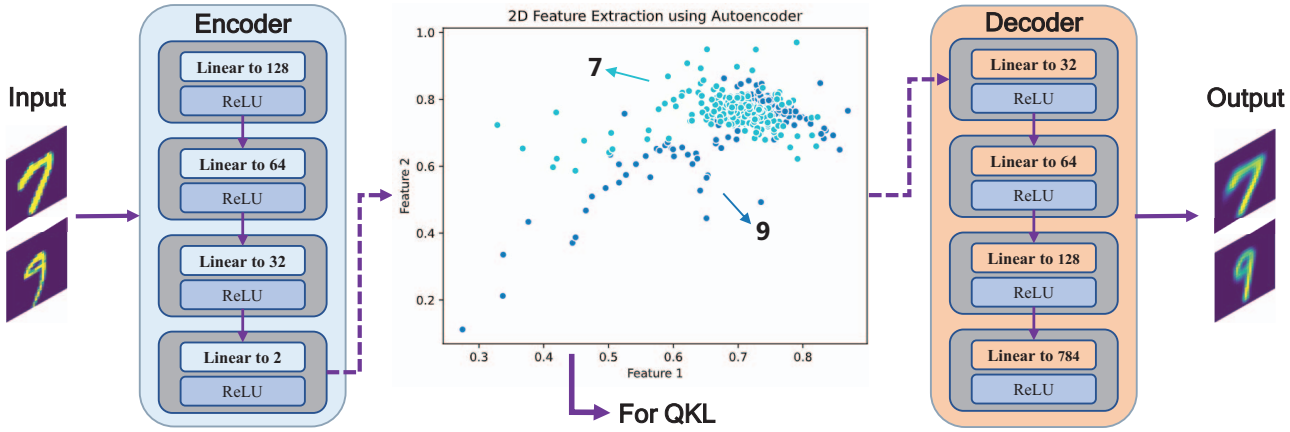

Fig. 7. Classical processing part for QKL. Digits 7 and 9, which are relatively challenging to distinguish, are selected. The encoder maps them to a low-dimensional representation, from where the decoder attempts to reconstruct the original input data.

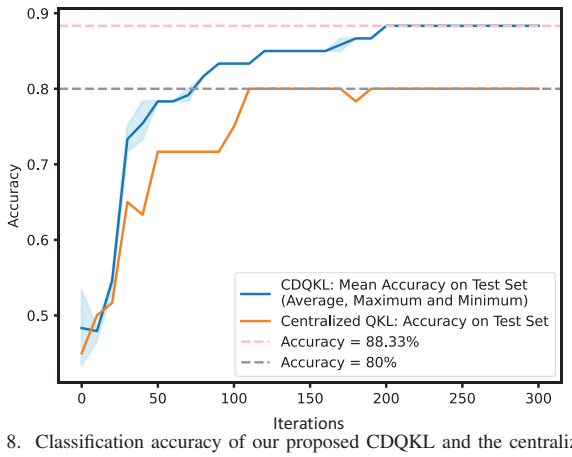

Fig. 8. Classification accuracy of our proposed CDQKL and the centralized QKL in the training process.

fully connected layers with gradually decreasing dimensions, maps them to a 2D representation, and the decoder, which contains 4 corresponding fully connected layers, attempts to reconstruct the original input data for digits 7 and 9 from this low-dimensional representation. Through self-supervised learning, low-dimensional features could be obtained, allowing us to use QKL for classification of digits 7 and 9.

In this scenario, we randomly partition the entire dataset into four parts for our proposed CDQKL. The learning settings are just same to those in artificial dataset experiments. The classification accuracy of our proposed CDQKL training results is presented in Fig. 8, which is from consensus-based distributed training but ultimately solving SVM using whole data. We can observe that our proposed CDQKL ultimately achieves significantly better classification accuracy than the centralized QKL. We believe the performance advantage is because that the distributed method is equivalent to adding noise [30], making it easier for the training process to pass through the barren plateau [31].

C. Discussion

In the experiments, CDQKL show advantages over centralized QKL. However, the specific effect shown also depends on the training details and specific data set. In fact, the learning ability of quantum kernel and how to design quantum kernel are still ongoing research. CDQKL provides a new perspective for improving the performance of QKL.

VI. Conclusion and Future Work

Conclusion. This work proposed a CDQKL approach that solely requires to exchange model parameter information between adjacent nodes. Through experiments on both artificial

and real-world datasets, we validated the performance of our proposed CDQKL in terms of classification accuracy and convergence speed. In addition to simply being used to improve the performance of QKL, CDQKL will also play an important role in privacy-preserving machine learning tasks in future distributed quantum computing scenarios due to no need of sharing local training data.

Future Work. It is noted that currently the issue such as device noise persist in quantum machine learning. Further theoretical and empirical research on real quantum computers is needed. Additionally, in our experiments, we observed that the constructed quantum kernel exhibits particular learning capabilities for specific patterns. In the future, we will further demonstrate the paradigm of our proposed CDQKL and explore its deployment in real-world scenarios like microgrids [32]–[35], state estimation [36], [37], wind turbines [38], and energy market [39].

ACKNOWLEDGMENT

The authors would like to thank Prof. Jinming Xu from Zhejiang University, Prof. Jianxi Gao from Rensselaer Polytechnic Institute and anonymous reviewers very much for their fruitful and insightful suggestions to improve this work.

REFERENCES

[1] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio, *et al.*, "Variational quantum algorithms," *Nature Reviews Physics*, vol. 3, no. 9, pp. 625–644, 2021.

[2] C. Ciliberto, M. Herbster, A. D. Ialongo, M. Pontil, A. Rocchetto, S. Severini, and L. Wossnig, "Quantum machine learning: a classical perspective," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 474, no. 2209, p. 20170551, 2018.

[3] T. Hofmann, B. Schölkopf, and A. J. Smola, "Kernel methods in machine learning," 2008.

[4] M. Schuld, F. Petruccione, M. Schuld, and F. Petruccione, "Quantum models as kernel methods," *Machine Learning with Quantum Computers*, pp. 217–245, 2021.

[5] Y. Liu, S. Arunachalam, and K. Temme, "A rigorous and robust quantum speed-up in supervised machine learning," *Nature Physics*, vol. 17, no. 9, pp. 1013–1017, 2021.

[6] H. Hsin-Yuan, B. Michael, M. Masoud, B. Ryan, B. Sergio, N. Hartmut, and R. M. Jarrod, "Power of data in quantum machine learning," *Nature Communications*, vol. 12, no. 2631, pp. 1–9, 2021.

[7] T. Hubregtsen, D. Wierichs, E. Gil-Fuster, P.-J. H. Derks, P. K. Faehrmann, and J. J. Meyer, "Training quantum embedding kernels on near-term quantum computers," *Physical Review A*, vol. 106, no. 4, p. 042431, 2022.

[8] G. Gentinetta, A. Thomsen, D. Sutter, and S. Woerner, "The complexity of quantum support vector machines," *Quantum*, vol. 8, p. 1225, 2024.

[9] J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, M. Mao, M. Ranzato, A. Senior, P. Tucker, K. Yang, *et al.*, "Large scale distributed deep networks," *Advances in neural information processing systems*, vol. 25, 2012.

[10] A. Mi, S. Deng, and J. Szefer, "Securing reset operations in nisq quantum computers," in *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, pp. 2279–2293, 2022.

[11] J. Verbraeken, M. Wolting, J. Katzy, J. Kloppenburg, T. Verbelen, and J. S. Rellermeyer, "A survey on distributed machine learning," *ACM Computing Surveys*, vol. 53, no. 2, pp. 1–33, 2020.

[12] J. Liu, J. Huang, Y. Zhou, X. Li, S. Ji, H. Xiong, and D. Dou, "From distributed machine learning to federated learning: A survey," *Knowledge and Information Systems*, vol. 64, no. 4, pp. 885–917, 2022.

[13] G. Gentinetta, D. Sutter, C. Zoufal, B. Fuller, and S. Woerner, "Quantum kernel alignment with stochastic gradient descent," *arXiv preprint arXiv:2304.09899*, 2023.

[14] M. E. Sahin, B. C. Symons, P. Pati, F. Minhas, D. Millar, M. Gabrani, J. L. Robertus, and S. Mensa, "Efficient parameter optimisation for quantum kernel alignment: A sub-sampling approach in variational training," *arXiv preprint arXiv:2401.02879*, 2024.

[15] H. Tang, B. Li, G. Wang, H. Xu, C. Li, A. Barr, P. Cappellaro, and J. Li, "Communication-efficient quantum algorithm for distributed machine learning," *Physical Review Letters*, vol. 130, no. 15, p. 150602, 2023.

[16] Y.-B. Sheng and L. Zhou, "Distributed secure quantum machine learning," *Science Bulletin*, vol. 62, no. 14, pp. 1025–1029, 2017.

[17] N. M. Neumann and R. S. Wezeman, "Distributed quantum machine learning," in *International Conference on Innovations for Community Services*, pp. 281–293, Springer, 2022.

[18] D. Awschalom, K. K. Berggren, H. Bernien, S. Bhave, L. D. Carr, P. Davids, S. E. Economou, D. Englund, A. Faraon, M. Fejer, *et al.*, "Development of quantum interconnects (quics) for next-generation information technologies," *PRX Quantum*, vol. 2, no. 1, p. 017002, 2021.

[19] D. Cuomo, M. Caleffi, and A. S. Cacciapuoti, "Towards a distributed quantum computing ecosystem," *IET Quantum Communication*, vol. 1, no. 1, pp. 3–8, 2020.

[20] W. J. Yun, J. P. Kim, S. Jung, J. Park, M. Bennis, and J. Kim, "Slimmable quantum federated learning," *arXiv preprint arXiv:2207.10221*, 2022.

[21] S. Y.-C. Chen and S. Yoo, "Federated quantum machine learning," *Entropy*, vol. 23, no. 4, p. 460, 2021.

[22] Y. Huang, Y. Sun, Z. Zhu, C. Yan, and J. Xu, "Tackling data heterogeneity: A new unified framework for decentralized sgd with sample-induced topology," *arXiv preprint arXiv:2207.03730*, 2022.

[23] Z. Li, B. Liu, and Z. Ding, "Consensus-based cooperative algorithms for training over distributed data sets using stochastic gradients," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 10, pp. 5579–5589, 2021.

[24] B. Liu, Z. Ding, and C. Lv, "Distributed training for multi-layer neural networks by consensus," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 5, pp. 1771–1778, 2019.

[25] B. Scholkopf and A. J. Smola, *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2018.

[26] M. A. Nielsen and I. L. Chuang, *Quantum computation and quantum information*. Cambridge university press, 2010.

[27] M. Schuld, V. Bergholm, C. Gogolin, J. Izaac, and N. Killoran, "Evaluating analytic gradients on quantum hardware," *Physical Review A*, vol. 99, no. 3, p. 032331, 2019.

[28] F. Bullo, *Lectures on network systems*. Kindle Direct Publishing Seattle, DC, USA, 2020.

[29] V. Bergholm, J. Izaac, M. Schuld, C. Gogolin, S. Ahmed, V. Ajith, M. S. Alam, G. Alonso-Linaje, B. AkashNarayanan, A. Asadi, *et al.*, "Pennylane: Automatic differentiation of hybrid quantum-classical computations," *arXiv preprint arXiv:1811.04968*, 2018.

[30] J. Liu, F. Wilde, A. A. Mele, L. Jiang, and J. Eisert, "Stochastic noise can be helpful for variational quantum algorithms," tech. rep., 2022.

[31] J. R. McClean, S. Boixo, V. N. Smelyanskiy, R. Babbush, and H. Neven, "Barren plateaus in quantum neural network training landscapes," *Nature communications*, vol. 9, no. 1, p. 4812, 2018.

[32] M. Liu, F. Teng, Z. Zhang, P. Ge, M. Sun, R. Deng, P. Cheng, and J. Chen, "Enhancing cyber-resiliency of der-based smart grid: A survey," *IEEE Transactions on Smart Grid*, pp. 1–1, 2024.

[33] M. Liu, C. Zhao, J. Xia, R. Deng, P. Cheng, and J. Chen, "Pddl: Proactive distributed detection and localization against stealthy deception attacks in dc microgrids," *IEEE Transactions on Smart Grid*, vol. 14, no. 1, pp. 714–731, 2023.

[34] M. Liu, C. Zhao, Z. Zhang, R. Deng, P. Cheng, and J. Chen, "Converter-based moving target defense against deception attacks in dc microgrids," *IEEE Transactions on Smart Grid*, vol. 13, no. 5, pp. 3984–3996, 2021.

[35] W. Yang, S. N. Sparrow, M. Ashtine, D. C. Wallom, and T. Morstyn, "Resilient by design: Preventing wildfires and blackouts with microgrids," *Applied Energy*, vol. 313, p. 118793, 2022.

[36] S. Peng, Z. Zhang, R. Deng, and P. Cheng, "Localizing false data injection attacks in smart grid: A spectrum-based neural network approach," *IEEE Transactions on Smart Grid*, 2023.

[37] Z. Zhang, R. Deng, and D. K. Yau, "Vulnerability of the load frequency control against the network parameter attack," *IEEE Transactions on Smart Grid*, 2023.

[38] S. Zhao, J. Xia, R. Deng, P. Cheng, Q. Yang, and X. Jiao, "Dual-triggered adaptive torque control strategy for variable-speed wind turbine against denial-of-service attacks," *IEEE Transactions on Smart Grid*, 2022.

[39] W. Yang, S. N. Sparrow, and D. C. H. Wallom, "Optimising multi-factor assistance in a deep learning-based electricity forecasting model with climate resilience: an australian case study," in *2023 IEEE PES Innovative Smart Grid Technologies Europe (ISGT EUROPE)*, pp. 1–5, 2023.