Quantum algorithms for anomaly detection using amplitude estimation

Ming-Chao Guo,$^{1,2}$ Hai-Ling Liu,$^1$ Yong-Mei Li,$^1$ Wen-Min Li,$^1$† Su-Juan Qiu,$^1$ Qiao-Yan Wen,$^1$ and Fei Gao$^1$†

$^1$State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing, 100876, China
$^2$State Key Laboratory of Cryptology, P.O. Box 5159, Beijing, 100878, China

Abstract

Anomaly detection plays a critical role in fraud detection, health care, intrusion detection, military surveillance, etc. Anomaly detection algorithm based on density estimation (called ADDE algorithm) is one of widely used algorithms. Liang et al. proposed a quantum version of the ADDE algorithm [Phys. Rev. A 99, 052310 (2019)] and it is believed that the algorithm has exponential speedups on both the number and the dimension of training data point over the classical algorithm. In this paper, we find that Liang et al.’s algorithm doesn’t actually execute. Then we propose a new quantum ADDE algorithm based on amplitude estimation. It is shown that our algorithm can achieves exponential speedup on the number $M$ of training data points compared with the classical counterpart. Besides, the idea of our algorithm can be applied to optimize the anomaly detection algorithm based on kernel principal component analysis (called ADKPCA algorithm). Different from the quantum ADKPCA proposed by Liu et al. [Phys. Rev. A 97, 042315 (2018)], compared with the classical counterpart, which offer exponential speedup on the dimension $d$ of data points, our algorithm achieves exponential speedup on $M$.

I. INTRODUCTION

Anomaly Detection (AD) refers to the problem of finding patterns in data that do not conform to expected behavior [1]. It is regarded as an essential branch in data mining and machine learning, and extensively used in a wide variety of fields, such as fraud detection for credit cards [2], fault detection of safety-critical systems [3], intrusion detection for network security [4] and health care [5]. Two common classical algorithms are AD algorithm based on Density Estimation [6] (ADDE algorithm) and AD algorithm based on Kernel Principal Component Analysis (ADKPCA algorithm) [7]. In the presence of a large amount of input data, these two classical algorithms can be very computationally intensive. It is worthwhile to explore more effective algorithms for AD.

Quantum computing has been demonstrated prominent advantages in some problems, such as factoring integers [8], searching in unstructured database [9], solving linear systems of equations [10][11], differential equation [12], quantum private query [13][15]. The combination of quantum computing and machine learning created an emerging interdisciplinary field, Quantum Machine Learning (QML) [16], which is regarded as one of the most promising research directions. QML made great strides in data classification [17][19], clustering [20], neural network [21], linear regression [22][24], association rule mining [24], dimensionality reduction [20][29], etc.

In the context of quantum computing, several works have been developed to solve AD problems. In 2018, Liu et al. proposed a quantum ADKPCA algorithm [30]. The quantum algorithm calculates the inner product of two vectors based on swap-test [31][32] to obtain the value of proximity measure with complexity $O(poly(M \log d))$, where $M$ and $d$ are the number and dimension of training data points, respectively. Subsequently, Liang et al. proposed a quantum version of the ADDE algorithm [33] and claimed that the algorithm has exponential speedups on $M$ and $d$ compared with the classical counterpart.

In this paper, we find the mistakes of Liang et al.’s algorithm, which mainly derived from the failure of controlled rotation operation in the algorithm to extract classical information. This makes the whole algorithm doesn’t actually execute. We propose a new quantum ADDE algorithm mainly utilizing amplitude estimation [34]. Our algorithm can achieve exponential speedup over the classical algorithm on $M$. In addition, the idea of our algorithm can be applied to optimize the ADKPCA algorithm. Different from Liu et al.’s algorithm [30], our algorithm shows exponential speedup on $M$ compared to the classical counterpart.

This paper is organized as follows. In Sec. II, we briefly review the classical ADDE and ADKPCA algorithms. In Sec. III, we present a new quantum ADDE algorithm and analyze its complexity in detail. In Sec. IV, we give an application of the idea of our algorithm in the ADKPCA algorithm and analyze its complexity. The conclusion is given in Sec. V. The algorithm of Liang et al. is reviewed and analyzed in Appendix A and Appendix B, respectively.

II. REVIEW OF ADDE AND ADKPCA ALGORITHMS

In this section, we introduce the classical ADDE algorithm [6] and ADKPCA algorithm [7].

Given a training data set $\{x^i\}_{i=1}^M$ consisting of $M$ normal data points and a new data point $x^0$, where

- $\{x^i\}_{i=1}^M$: Normal data points
- $x^0$: New data point
- $\{\mathcal{A}(x^i)\}_{i=1}^M$: Amplitude of $x^i$
- $\mathcal{A}(x^0)$: Amplitude of $x^0$
- $\mathcal{N}(x^i)$: Normalization of $x^i$
- $\mathcal{N}(x^0)$: Normalization of $x^0$
- $\mathcal{P}(x^i)$: Probability of $x^i$
- $\mathcal{P}(x^0)$: Probability of $x^0$
- $\mathcal{D}(x^i)$: Density of $x^i$
- $\mathcal{D}(x^0)$: Density of $x^0$
- $\mathcal{R}(x^i)$: Rank of $x^i$
- $\mathcal{R}(x^0)$: Rank of $x^0$
- $\mathcal{S}(x^i)$: Similarity of $x^i$
- $\mathcal{S}(x^0)$: Similarity of $x^0$
- $\mathcal{F}(x^i)$: Failure of $x^i$
- $\mathcal{F}(x^0)$: Failure of $x^0$
- $\mathcal{A}(x^i)$: Amplitude of $x^i$
- $\mathcal{A}(x^0)$: Amplitude of $x^0$
- $\mathcal{N}(x^i)$: Normalization of $x^i$
- $\mathcal{N}(x^0)$: Normalization of $x^0$
- $\mathcal{D}(x^i)$: Density of $x^i$
- $\mathcal{D}(x^0)$: Density of $x^0$
- $\mathcal{R}(x^i)$: Rank of $x^i$
- $\mathcal{R}(x^0)$: Rank of $x^0$
- $\mathcal{S}(x^i)$: Similarity of $x^i$
- $\mathcal{S}(x^0)$: Similarity of $x^0$
- $\mathcal{F}(x^i)$: Failure of $x^i$
- $\mathcal{F}(x^0)$: Failure of $x^0$
\( x^i = (x^i_1, \cdots, x^i_d) \in \mathcal{R}^d, i = 0, 1, \ldots, M. \) The training data set can be represented by a \( M \times d \) data matrix \( X = (x^1, \cdots, x^M)^T. \)

### A. ADDE algorithm

Assume that each feature is independent of each other and follows a Gaussian distribution \( x^j_i \sim N(\mu_j, \sigma^2_j), j = 1, \cdots, d, i = 1, \cdots, M. \) The algorithm first establishes a statistical model \( P(x) \) from the training data set. Then calculate the value of \( P(x^0) \) and compare it with a pre-determined threshold \( \delta \) to identify whether the new data point \( x^0 \) is an anomaly or not. The whole procedure is depicted as follows.

1. Fit the parameters mean \( \mu_j \) and variance \( \sigma^2_j \) by the training data set \((j = 1, \cdots, d)\):

\[
\mu_j = \frac{1}{M} \sum_{i=1}^{M} x^i_j, \quad \sigma^2_j = \frac{1}{M} \sum_{i=1}^{M} (x^i_j - \mu_j)^2. \quad (1)
\]

2. Establish the statistical model \( P(x) \) (the joint density function):

\[
P(x) = \prod_{j=1}^{d} P(x_j; \mu_j, \sigma^2_j) = \prod_{j=1}^{d} \frac{1}{\sqrt{2\pi\sigma_j}} e^{-\frac{(x_j - \mu_j)^2}{2\sigma^2_j}}. \quad (2)
\]

3. Given a new data point \( x^0 \), calculate \( P(x^0) \):

\[
P(x^0) = \prod_{j=1}^{d} P(x^0_j; \mu_j, \sigma^2_j) = \prod_{j=1}^{d} \frac{1}{\sqrt{2\pi\sigma_j}} e^{-\frac{(x^0_j - \mu_j)^2}{2\sigma^2_j}}. \quad (3)
\]

Simplify the Eq.(3):

\[
\ln P(x^0) = -\frac{d}{2} \ln 2\pi - \sum_{j=1}^{d} \ln \sigma_j - \sum_{j=1}^{d} \frac{(x^0_j - \mu_j)^2}{2\sigma^2_j}. \quad (4)
\]

Set a threshold \( \delta \) in advance, if \( \ln P(x^0) < \ln \delta \), one can flag the new data as anomaly, otherwise, it is judged as normal. The complexity of ADDE algorithm is \( O(Md) \). Therefore, when the size of the training data set is large, the ADDE algorithm may take a significant amount of time to execute.

### B. ADKPCA algorithm

First, the mean vector \( \mu \) and covariance matrix \( \Sigma \) of the training data set are defined as follows:

\[
\mu = \frac{1}{M} \sum_{i=1}^{M} x^i, \quad \Sigma = \frac{1}{M - 1} \sum_{i=1}^{M} (x^i - \mu)(x^i - \mu)^T. \quad (5)
\]

The main purpose of the algorithm is to detect the difference in the distance between the new data point \( x^0 \) and mean vector and the variance of the training data set along the direction \( x^0 - \mu \). That is, to calculate the proximity measure:

\[
f(x^0) = |x^0 - \mu|^2 - (x^0 - \mu)^T \Sigma (x^0 - \mu). \quad (6)
\]

It can quantify how anomalous the point \( x^0 \) is compared to the training data. A larger the value of \( f(x^0) \) implies a more anomalous data point than a smaller \( f(x^0) \).

This method also allows us to classify anomalies in non-linear feature spaces. The inner products are performed in an abstract linear feature space. The inner product can be represented by a kernel function \( k(x_i, x_j) \), which can be regarded as a non-linear function. For brevity, we only consider the linear kernel.

### III. A NEW QUANTUM ADDE ALGORITHM

Liang et al.’s algorithm \[33\] is reviewed in Appendix A. And we find the mistakes of this algorithm, which mainly derived from the failure of controlled rotation operation in the algorithm to extract classical information. This makes the whole algorithm doesn’t actually execute. See Appendix B for a detailed analysis.

In this section, we present a new quantum ADDE algorithm in Sec. III A, and analyze its complexity in Sec. III B.

#### A. Algorithm

We assume that there are already quantum oracles that can efficiently access the elements in the data matrix \( X \) and new data point \( x^0 \) with time \( O(\log Md) \) and \( O(\log d) \), respectively. (for all \( i = 1, \cdots, M, j = 1, \cdots, d \))

\[
O_X : |i\rangle|j\rangle|0\rangle \Rightarrow |i\rangle|j\rangle|x^i_j\rangle,
\]

\[
O_S : |j\rangle|0\rangle \Rightarrow |j\rangle|x^0_j\rangle
\]

This assumption can be made naturally when each element in \( X \) and \( x^0 \) is stored in quantum random access memory (QRAM)\[35\].

Our algorithm consists of three steps: in step 1, we prepare the state \( \frac{1}{\sqrt{d}} \sum_{j=1}^{d} |j\rangle|\mu_j\rangle \) in the quantum state \( \frac{1}{\sqrt{d}} \sum_{j=1}^{d} |j\rangle|\mu_j\rangle|\sigma^2_j\rangle|x^0_j\rangle \) can be generated; in step 2, we perform amplitude estimation \[33\] to obtain the classic information of \( \ln P(x^0) \). The entire algorithm process is shown in Fig. 1.

In order to implement steps 1 and 2, we found that the parameter values of Eq. (1) can be regarded as the inner product of two vectors, \((j = 1, 2, \cdots, d)\)

\[
\mu_j = \frac{1}{M}(x^1_j, \cdots, x^M_j) \cdot (1, \cdots, 1)^T,
\]

\[
\sigma^2_j = \frac{1}{M}(x^1_j - \mu_j, \cdots, x^M_j - \mu_j) \cdot (x^1_j - \mu_j, \cdots, x^M_j - \mu_j)^T. \quad (8)
\]
These can be calculated in parallel by amplitude estimation and controlled rotation.

**Step 1.** Prepare the state: \(\frac{1}{\sqrt{d}} \sum_{j=1}^{d} |j\rangle |\mu_j\rangle\).

(1.1) Initialize quantum state

\[
|0^{\otimes \log d} \rangle_1 |0 \rangle_2 |0^{\otimes \log M} \rangle_3 |0 \rangle_4 |0 \rangle_5.
\]

(9)

(1.2) Perform \(H^{\otimes \log d} \otimes H \) and \(H^{\otimes \log M} \) on the first, second and third register, respectively

\[
\frac{1}{\sqrt{d}} \sum_{j=1}^{d} \frac{1}{\sqrt{2}}(|j\rangle + |1\rangle)_{2} \frac{1}{\sqrt{M}} \sum_{l=1}^{M} |l\rangle_{4} |0 \rangle_{5}.
\]

(10)

(1.3) Run the oracle \(O_X\) on the step (1.2) to prepare

\[
\frac{1}{\sqrt{d}} \sum_{j=1}^{d} \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)_{2} \frac{1}{\sqrt{M}} \sum_{l=1}^{M} |l\rangle_{3} |x_j^2\rangle_{4} |0 \rangle_{5}.
\]

(11)

(1.4) Perform \(I_1 \otimes (|0\rangle_2 \otimes |0\rangle_3 \otimes U_{45} + |1\rangle_2 |1\rangle_3 \otimes I_3 \otimes I_{45}\), where \(U\) is a controlled rotation \([10][36]\). Discard the fourth register, the state will becomes

\[
\frac{1}{\sqrt{d}} \sum_{j=1}^{d} \frac{1}{\sqrt{2}} \left[ |0\rangle + \frac{1}{M} \sum_{l=1}^{M} |l\rangle + \sqrt{1 - \left(\frac{x_j^2}{C}\right)^2} - |1\rangle \right]
\]

\[
|0\rangle_{235} := \frac{1}{\sqrt{d}} \sum_{j=1}^{d} \frac{1}{\sqrt{2}} (|0\rangle_{2} |\phi_j\rangle + |1\rangle |h\rangle)_{235}.
\]

(12)

(1.5) Apply \(H\) gate to the second register

\[
\frac{1}{\sqrt{d}} \sum_{j=1}^{d} |j\rangle \left| \frac{1}{2} \left[ |0\rangle_2 (|\phi_j\rangle + |h\rangle)_{35} + |1\rangle_2 (|\phi_j\rangle - |h\rangle)_{35} \right] \right.
\]

\[
:= \frac{1}{\sqrt{d}} \sum_{j=1}^{d} |j\rangle_{1} |\Phi_j\rangle_{235}.
\]

(13)

Each \(\Phi_j\) is rewritten as \(|\Phi_j\rangle = \sin \theta_j |\Phi_0^j\rangle + \cos \theta_j |\Phi_1^j\rangle\), where \(\Phi_0^j, \Phi_1^j\) are the normalized quantum states of \(|0\rangle_{2} (|\phi_j\rangle + |h\rangle)\) and \(|1\rangle (|\phi_j\rangle - |h\rangle)\), respectively. In addition, \(\sin^2 \theta_j = \frac{1}{2} + \frac{1}{2} |\langle \phi_j |h\rangle| \in [0, \frac{1}{2}]\). We can use amplitude estimation \([7]\) to estimate \(\langle \phi_j |h\rangle\). Define \(Q_j = -A_j S_j A_j^\dagger S_j\), where \(A_j : |0\rangle_{234} \rightarrow |\Phi_j\rangle, S_0 = I - 2|0\rangle_{234} |0\rangle_{234} \otimes I, S_x = (I - 2|0\rangle_2 |0\rangle_2) \otimes I\). Perform \((Q_j)^l\) on the state \(|\Phi_j\rangle\):

\[
(Q_j)^l |\Phi_j\rangle = \sin(l(2 + 1)\theta_j) |\Phi_0^j\rangle + \cos(l(2 + 1)\theta_j) |\Phi_1^j\rangle,
\]

(14)

for any \(l \in \mathbb{N}\), \(Q_j\) acts as a rotation in the two subspace \(|\Phi_0^j\rangle, |\Phi_1^j\rangle\) and it has two eigenvalues \(e^{l(2 + 1)\theta_j}\) with eigenstates \(|\Phi_0^j\rangle\) (un-normalized).

(1.6) Add a ancillary register and excute amplitude estimation to generate

\[
\frac{1}{\sqrt{d}} \sum_{j=1}^{d} |j\rangle_{1} \left( |\Phi_0^j\rangle |\frac{1}{\pi} \rangle + |\Phi_1^j\rangle |1 - \frac{1}{\pi} \rangle \right)_{2356}.
\]

(15)

(1.7) Compute \(|\mu_j\rangle = |C \cdot (2 \sin^2 \theta_j - 1)\rangle\) via the quantum multiply-adder (QMA) and sine gate \([37][38]\). Hence, the state will be

\[
\frac{1}{\sqrt{d}} \sum_{j=1}^{d} |j\rangle_{1} |\tilde{\mu}_j\rangle_{6}.
\]

(16)

where \(\tilde{\mu}_j\) represents the estimate of \(\mu_j\). To simplify the notation, we add the tilde for estimates throughout this paper; that is, we denote \(\tilde{\alpha}\) as the estimate of \(\alpha\). The complete circuit diagram of step 1 is described in Figure 2.

**Step 2.** Prepare the state: \(\frac{1}{\sqrt{d}} \sum_{j=1}^{d} |j\rangle |\mu_j\rangle |\sigma_j^2\rangle |x_j^0\rangle\).

(1.1) Prepare the quantum state

\[
\frac{1}{\sqrt{d}} \sum_{j=1}^{d} |j\rangle_{1} |\tilde{\mu}_j\rangle_{6} |0\rangle_{7} |0\rangle_{8}.
\]

(17)

(2.2) Perform \(H^{\otimes \log M} \) gates on the seventh register and the oracle \(O_X\) to obtain

\[
\frac{1}{\sqrt{d}} \sum_{j=1}^{d} |j\rangle_{1} |\tilde{\mu}_j\rangle_{6} \frac{1}{\sqrt{M}} \sum_{l=1}^{M} |i\rangle_{7} |x_j^3\rangle_{8}.
\]

(18)

(2.3) The QMA gate is performed between sixth and seventh registers to generate

\[
\frac{1}{\sqrt{d}} \sum_{j=1}^{d} |j\rangle_{1} |\tilde{\mu}_j\rangle_{6} \frac{1}{\sqrt{M}} \sum_{l=1}^{M} |i\rangle_{7} |x_j^3 - \tilde{\mu}_j\rangle_{8}.
\]

(19)

(2.4) Add one qubit and run controlled rotation, discard the eighth register to generate

\[
\frac{1}{\sqrt{d}} \sum_{j=1}^{d} |j\rangle_{1} |\tilde{\mu}_j\rangle_{6} \frac{1}{\sqrt{M}} \sum_{l=1}^{M} |i\rangle_{7} \left( |x_j^3 - \tilde{\mu}_j\rangle_D |0\rangle + \sqrt{1 - \left(\frac{x_j^3 - \tilde{\mu}_j}{D}\right)^2} |1\rangle \right)_{9} := \frac{1}{\sqrt{d}} \sum_{j=1}^{d} |j\rangle_{1} |\tilde{\mu}_j\rangle_{6} |\chi_j\rangle_{79}.
\]

(20)

where \(D = \max_{i,j} |x_i^3 - \mu_j|\).

(2.5) Similar to steps (1.5)-(1.6), we can get

\[
\frac{1}{\sqrt{d}} \sum_{j=1}^{d} |j\rangle_{1} |\tilde{\mu}_j\rangle_{6} |\sigma_j^2\rangle_{10}\rangle_{10}.
\]

(21)

(2.6) Append an ancilla register and call the oracle \(O_X\)

\[
\frac{1}{\sqrt{d}} \sum_{j=1}^{d} |j\rangle_{1} |\tilde{\mu}_j\rangle_{6} |\sigma_j^2\rangle_{10} |x_j^0\rangle_{f}.
\]

(22)
FIG. 1: The process diagram of quantum ADDE algorithm, where $\delta$ is a predetermined threshold, $C - R$ denotes controlled rotation and QMA denotes quantum multiply-adder.

(3.3) According to step 2, we can prepare the state

$$\frac{1}{\sqrt{d}} \sum_{j=1}^{d} |j\rangle |\tilde{\sigma}^2_j\rangle |0\rangle_g = \frac{1}{\sqrt{d}} \sum_{j=1}^{d} |j\rangle |\hat{\sigma}^2_j\rangle |0\rangle_g$$  \hspace{1cm} (25)

(3.4) Perform unitary operation $|0\rangle_c |0\rangle_c \otimes I_1 \otimes U_{1g} + |1\rangle_c |1\rangle_c \otimes I_{16g}$. Uncompute the sixth register:

$$\frac{1}{\sqrt{d}} \sum_{j=1}^{d} |j\rangle |1\rangle_g := \frac{1}{\sqrt{d}} \sum_{j=1}^{d} |j\rangle |\phi_j\rangle + |1\rangle_h^g$$  \hspace{1cm} (26)

where $|\phi_j\rangle := \frac{1}{\sqrt{d}} \sum_{j=1}^{d} |j\rangle \left( \ln \frac{\hat{\sigma}^2_j}{E} |0\rangle + \sqrt{1 - \left( \frac{\ln \hat{\sigma}^2_j}{E} \right)^2} |1\rangle \right) + |1\rangle_c |1\rangle_g$,

(3.5) Execute the $H$ gate to obtain

$$\frac{1}{\sqrt{d}} \left[ |0\rangle_c |\phi_j\rangle + |1\rangle_h^g \right]$$

(3.6) Run amplitude estimation to obtain

$$\hat{q} = \frac{1}{d} \sum_{j=1}^{d} \frac{\ln \hat{\sigma}^2_j}{L}$$  \hspace{1cm} (28)
Hence, we can estimate the value of $\ln P(x^0)$, that is,
\[
\ln P(x^0) = -\frac{d}{2} \ln 2\pi - \frac{1}{2} dL \cdot \hat{q} - \frac{1}{2} dT^2 \cdot \hat{p}.
\] (29)

B. Complexity Analysis

(i). The complexity of step 1 is $O((\log M d)/\varepsilon')$, which is mainly derived from the oracle $O_X$ and amplitude estimation. Specific analysis is as follows:

In steps (1.2)—(1.4), it takes $H$ gates, controlled rotation and oracle $O_X$ with complexity $O(\log M d)$. In step (1.6), the amplitude estimation block needs $O(1/\varepsilon')$ applications of $Q_j = -A_j S_0 A_j^\dagger S_X$ to achieve error $\varepsilon'$. The complexity of performing unitary operator $Q_j$ is $O(\log M d)$. In step (1.7), it takes QMA and sine gates with complexity $O(\text{polylog } 1/\varepsilon')$, which is smaller than $O(1/\varepsilon')$, the complexity of these gates can be omitted.

Now we analyze the error of step 1, which is mainly caused by the estimation of $\theta_j$, ($|\theta_j - \hat{\theta}_j| \leq \varepsilon'$)

\[
|\mu_j - \hat{\mu}_j| = C(2\sin^2 \theta_j - 1 - (2\sin^2 \hat{\theta}_j - 1)) \leq 2C\varepsilon'.
\] (30)

In a word, the complexity of step 1 is $O((\log M d)/\varepsilon')$ and the error is $2C\varepsilon'$ to get \( \sum_{j=1}^{d} |j\rangle|\hat{\mu}_j\rangle \).

(ii). The time complexity of step 2 is $O((\log M d)/\varepsilon''\varepsilon'')$, which is mainly stem from the amplitude estimation of step (2.5). The detailed analysis is as follows:

In step (2.1), perform the operations of step 1 with complexity $O((\log M d)/\varepsilon')$. In step (2.2), it takes the oracle $O_X$ with complexity $O(\log M d)$. In step (2.3), the QMA gate is performed and its complexity can be ignored. In step (2.5), the amplitude estimation block needs $O(1/\varepsilon'')$ applications of the operation of step (2.4) to achieve error $\varepsilon''$. It implies the time complexity of step 2 is $O((\log M d)/\varepsilon''\varepsilon'')$. In step (2.6), it takes the oracle $O_x$ with complexity $O(\log d)$. The complexity of step 1 and step 2 are illustrated in the TABLE I.

| steps | complexity | steps | complexity |
|-------|------------|-------|------------|
| (1.2)—(1.4) | $O(\log M d)$ | (2.1) | $O(\log M d)$ |
| — | — | (2.2) | $O(\log M + \log M d)$ |
| — | — | (2.3)-(2.4) | $O(\text{polylog } \frac{1}{\varepsilon''})$ |
| (1.6) | $O(\log M d)$ | (2.5) | $O(\frac{\log M d}{\varepsilon''})$ |
| (1.7) | $O(\text{polylog } \frac{1}{\varepsilon''})$ | (2.6) | $O(\log d)$ |
| total | $O(\log M d)$ | total | $O(\log M d)$ |

Then we analyze the error of $\sigma_j^2$:

\[
|\hat{\sigma}_j^2 - \sigma_j^2| = |\hat{\sigma}_j^2 - \frac{1}{M} \sum_{i=1}^{M} (x_j^i - \mu_j)^2|
\leq 2D \cdot \varepsilon'' + \left| \frac{1}{M} \sum_{i=1}^{M} (x_j^i - \hat{\mu}_j)^2 - \frac{1}{M} \sum_{i=1}^{M} (x_j^i - \mu_j)^2 \right|
\leq 2D \cdot \varepsilon'' + 8C^2\varepsilon',
\] (31)

where $2D \cdot \varepsilon''$ is derived from the amplitude estimation in step (2.5).

(iii). The complexity of step 3 is mainly came from amplitude estimation of step (3.2) and step (3.6), and assume its error is $\varepsilon'''$. The complexity of estimating the values of $\sum_{j=1}^{d} (x_j^0 - x_j^0)^2$ and $\sum_{j=1}^{d} \ln \sigma_j$ are $O(\frac{\log M d}{\varepsilon'''\varepsilon''''})$, and their errors are as shown:

\[
|dE\hat{q} - \sum_{j=1}^{M} \ln \sigma_j^2| = |dE\hat{q} - \sum_{j=1}^{d} \frac{\ln \hat{\sigma}_j^2}{E} + \frac{\ln \hat{\sigma}_j^2}{E} - E| \\
\leq E(d\varepsilon''' + \left| \frac{1}{M} \sum_{j=1}^{M} \ln \hat{\sigma}_j^2 - \ln \sigma_j^2 \right|) \\
\leq E\varepsilon''' + \left| \ln (\frac{8C^2\varepsilon' + 2D\varepsilon''}{\min \sigma_j}) \right| \leq \frac{(E + 3 \min \sigma_j)\varepsilon}{3T^2}.
\] (32)

\[
|dT^2 \hat{p} - \sum_{j=1}^{d} \frac{(x_j^0 - \mu_j)^2}{\sigma_j^2}| = \\
= |dT^2 \hat{p} - \sum_{j=1}^{d} \frac{(x_j^0 - \hat{\mu}_j)^2}{dT^2 \hat{\sigma}_j^2} + \sum_{j=1}^{d} \frac{(x_j^0 - \hat{\mu}_j)^2}{dT^2 \hat{\sigma}_j^2} - \frac{(x_j^0 - \mu_j)^2}{dT^2 \sigma_j^2}| \\
\leq |dT^2 \varepsilon''' + \sum_{j=1}^{d} \frac{(x_j^0 - \hat{\mu}_j)^2}{\sigma_j^2} - \frac{(x_j^0 - \mu_j)^2}{\sigma_j^2} | \\
\leq |dT^2 \varepsilon''' + dT \frac{8C^2\varepsilon' + 2D\varepsilon''}{\min_j \sigma_j^2} + d \frac{8C^2\varepsilon'}{\min_j \sigma_j^2} |.
\] (33)

If $C, D, T, E, \min_j \sigma_j^2 = O(1)$, and $\varepsilon''' = \frac{\varepsilon'}{\min_j \sigma_j^2}$, $\varepsilon'' = \frac{\varepsilon''}{\min_j \sigma_j^2}$, $\varepsilon'''' = \frac{\varepsilon''''}{\min_j \sigma_j^2}$. The overall runtime will be

\[
O(d^3 \log(M d)\varepsilon^{-3}).
\]

That is, we can get an $\varepsilon$-approximate of $\ln P(x^0)$ in complexity $O(d^3 \log(M d)\varepsilon^{-3})$. This means that our quantum algorithm is exponentially faster than the corresponding classical algorithm when $1/\varepsilon, d = O(\log M)$.

IV. APPLICATION IN ADKPCA ALGORITHM

In this section, we propose an application of the idea of our algorithm in the ADKPCA algorithm, and assume that there are already quantum oracles $O_X$ and $O_x$, which are the same as the quantum ADDE algorithm.

A. Algorithm

In order to calculate the proximity measure $f(x^0)$, our quantum algorithm consists of the following two steps:

**Step 1.** Computing $|x^0 - \mu|^2$.

(1.1) According to step 1 in Sec. III A, we can prepare quantum state $\frac{1}{\sqrt{d}} \sum_{j=1}^{d} |j\rangle|0\rangle|\hat{\mu}_j\rangle$. 

TABLE I: The time complexity of step 1 and step 2
(1.2) Perform the oracle $O_{\mathbf{z}}$ on step (1.1):

$$\frac{1}{\sqrt{d}} \sum_{j=1}^{d} |j\rangle|\hat{\mu}_j\rangle \xrightarrow{O_{\mathbf{z}}} \frac{1}{\sqrt{d}} \sum_{j=1}^{d} |j\rangle|x_j^0\rangle|\hat{\mu}_j\rangle.$$ (34)

(1.3) Execute QMA gate and controlled rotation to generate:

$$\frac{1}{\sqrt{d}} \sum_{j=1}^{d} |j\rangle|x_j^0 - \hat{\mu}_j\rangle(x_j^0 - \hat{\mu}_j|0\rangle + \sqrt{1 - (x_j^0 - \hat{\mu}_j/C)^2}|1\rangle)$$

$$:= |\psi_0\rangle + |\psi_1\rangle,$$ (35)

where $C' = \max_j |x_j^0 - \mu_j|.$

(1.4) Run amplitude estimation, we can get:

$$\hat{a} = \frac{1}{d} \sum_{j=1}^{d} (x_j^0 - \hat{\mu}_j)^2.$$ (36)

Therefore, $|z^0|^2 = \sum_{j=1}^{d} (x_j^0 - \mu_j)^2 = d(C')^2 \cdot \hat{a}.$

**Step 2. Computing** $(x^0 - \mu)^T \Sigma(x^0 - \mu).$

(2.1) Prepare the state $|0\rangle|0\rangle \frac{1}{\sqrt{d}} \sum_{j=1}^{d} |j\rangle|x_j^0\rangle|\hat{\mu}_j\rangle.$

(2.2) Perform $H$ gates and the oracle $O_{\mathbf{X}}$ to obtain:

$$\frac{1}{\sqrt{M}} \sum_{i=1}^{M} |i\rangle|x_i^0\rangle \xrightarrow{H \otimes O_{\mathbf{X}}} \frac{1}{\sqrt{d}} \sum_{j=1}^{d} |j\rangle|x_j^0\rangle|\hat{\mu}_j\rangle$$ (37)

(2.3) Perform QMA gate and control rotation operation, uncompute the last three registers, obtain

$$\frac{1}{\sqrt{M}} \sum_{i=1}^{M} |i\rangle \sqrt{d} \sum_{j=1}^{d} |j\rangle\left((x_j^0 - \hat{\mu}_j)(x_j^0 - \hat{\mu}_j)\right)|0\rangle$$

$$+ \sqrt{1 - (x_j^0 - \hat{\mu}_j/C)^2}|1\rangle) := \frac{1}{\sqrt{M}} \sum_{i=1}^{M} |i\rangle|\Psi_i\rangle,$$ (38)

where $C'' = \max_j (|x_j^0 - \mu_j|)(x_j^0 - \mu_j).$

(2.4) Similar to step (1.3) in Sec. III A, we can prepare

$$\frac{1}{\sqrt{M}} \sum_{i=1}^{M} \frac{1}{2} |0\rangle(|\Psi_i\rangle + |h'\rangle) + |1\rangle(|\Psi_i\rangle - |h'\rangle|$$

$$:= \frac{1}{\sqrt{M}} \sum_{i=1}^{M} |i\rangle|\varphi_i\rangle,$$ (39)

where $|h'\rangle = \frac{1}{2} \sum_{j=1}^{d} |j\rangle|0\rangle.$

(2.5) Excute amplitude estimation, QMA and sine gate (similar to the steps (1.6)-(1.7) in Sec. III A), generate

$$\frac{1}{\sqrt{M}} \sum_{i=1}^{M} |i\rangle|\bar{\omega}_i\rangle,$$ (40)

where $\omega_i := \langle \Psi_i|h'\rangle = \frac{1}{d} \sum_{j=1}^{d} (x_j^0 - \hat{\mu}_j)(x_j^0 - \hat{\mu}_j)^2/C''.$

(2.6) Add an ancilla register and perform controlled rotation to obtain the quantum state:

$$\frac{1}{\sqrt{M}} \sum_{i=1}^{M} |i\rangle|\bar{\omega}_i\rangle(\bar{\omega}_i|0\rangle + \sqrt{1 - \bar{\omega}_i^2}|1\rangle).$$ (41)

(2.7) Run amplitude estimation to obtain

$$\hat{b} = \frac{1}{M} \sum_{i=1}^{M} (\bar{\omega}_i)^2.$$ (42)

The value of Eq. (6) can be approximated as

$$f(x^0) = d(C'')^2 \cdot \hat{a} - \frac{M}{M - 1} (dC'')^2 \cdot \hat{b}.$$ (43)

**B. Complexity analysis**

(i). The complexity of step 1 is $O(\frac{\log Md}{\epsilon})$, which is mainly derived from the amplitude estimation. The detailed analysis is shown below:

According to step 1 performed in Sec. III A, the time complexity of step (1.1) is $O(\log Md)$, and error is $2C^\epsilon'$. In step (1.2), the oracle $O_{\mathbf{z}}$ is performed with complexity $O(\log d)$. In step (1.3), it takes QMA and controlled rotation with complexity $O(\text{polylog}(1/\epsilon'))$, the complexity of these gates can be omitted. In step (1.4), the amplitude estimation is performed with complexity $O(\frac{\log Md}{\epsilon})$ to achieve error $\epsilon''$.

$$|\hat{a} - \frac{1}{d} \sum_{j=1}^{d} (x_j^0 - \hat{\mu}_j)^2| \leq \epsilon''.$$ (44)

Therefore, the error of step 1 is shown below

$$|d(C'')^2 \cdot \hat{a} - \sum_{j=1}^{d} (x_j^0 - \mu_j)^2|$$

$$\leq d(C'')^2 \epsilon'' + \left|\sum_{j=1}^{d} (x_j^0 - \hat{\mu}_j)^2 - \sum_{j=1}^{d} (x_j^0 - \mu_j)^2\right|$$

$$\leq d(C'')^2 \epsilon'' + 4dC'C\epsilon',$$ (45)

where $|\mu_j - \hat{\mu}_j| \leq 2C\epsilon'$ and $C' = \max_{x_j^0} |x_j^0|$, which is the same as the error analysis of Step 1 in Sec. III A. When $\epsilon' = \frac{4M}{4M+1}$ and $\epsilon'' = \frac{\epsilon}{dC'C\epsilon'},$ $|d(C'')^2 \cdot a - \sum_{j=1}^{d} (x_j^0 - \mu_j)^2| \leq \epsilon$. Then, step 1 returns an $\epsilon$-approximate of $|z^0 - \mu|^2$ in complexity $O(\frac{d^3 \log Md}{\epsilon^3})$.

(ii). The complexity of step 2 is $O(\frac{\log Md}{\epsilon^3})$. The specific analysis is as follows:

In step (2.1), perform the Oracle $O_{\mathbf{X}}$ with complexity of $O(\log Md)$. In steps (2.2)-(2.3), the $H$ gate, QMA and controlled rotation are performed. In step (2.4), we still need the information of $\omega_i$ to do further operations, and thus we store it in an ancilla register as Eq. (40). But, in step (2.6), we need an output about the estimate of
\[
\frac{1}{M} \sum_{i=1}^{M} \omega_i^2. \text{ Hence, after the amplitude estimation, we need to perform a measurement on the auxiliary register. It is assumed that the errors of amplitude estimation of step (2.4) and step (2.6) are } \varepsilon''\text{ and } \varepsilon''', \text{ respectively.}
\]

\[
\begin{align*}
\left| \hat{b} - b + b - \frac{1}{Md^2} \sum_{i=1}^{M} \left[ \sum_{j=1}^{d} \frac{(x_j^0 - \mu_j)(x_j^i - \mu_j)}{C''} \right]^2 + \left[ \sum_{j=1}^{d} \frac{(x_j^0 - \mu_j)(x_j^i - \mu_j)}{C''} \right]^2 - \left[ \sum_{j=1}^{d} \frac{(x_j^0 - \mu_j)(x_j^i - \mu_j)}{C''} \right]^2 \right|.
\end{align*}
\]

where the difference of the first two items is the error of the amplitude estimation of step (2.6), and the difference of the third and fourth items is the error of the amplitude estimation of step (2.4). That is,

\[
\left| \frac{1}{M} \sum_{i=1}^{M} \left[ \sum_{j=1}^{d} \frac{(x_j^0 - \mu_j)(x_j^i - \mu_j)}{dC''} \right]^2 \right| \leq \varepsilon'' + \varepsilon'' + 16C^2 \varepsilon'.
\]

When \( \varepsilon''' = \frac{dC''}{\varepsilon} \), \( \varepsilon'''' = \frac{dC''}{\varepsilon} \), and \( \varepsilon' = \frac{dC''}{\varepsilon} \), \( \varepsilon'' = \frac{dC''}{\varepsilon} \), and \( \varepsilon'''' = \frac{dC''}{\varepsilon} \). That is, the algorithm returns a 2-\( \varepsilon \)-approximate of \( f(\mathbf{x}) \) in complexity \( O(d^6 \log(Md)\varepsilon^{-3}) \).

Different from Liu et al.’s algorithm [30] calculates inner product of two vectors based on swap-test, our algorithm calculates \( M \) inner product values in parallel based on the amplitude estimation. This means that our algorithm achieves exponential speedup on \( M \) compared with the classical counterpart.

V. CONCLUSION

In the present study, we found the mistakes in the Liang et al.’s algorithm and proposed a new quantum ADDE algorithm mainly based on amplitude estimation. Our algorithm can achieve exponential speedup on the number of training data points \( M \) compared with the classical counterpart. Moreover, our algorithm can be applied to optimize the ADKPCA algorithm. This quantum algorithm has exponential speed on \( M \) compared with the classical algorithm, different from Liu et al.’s algorithm. We hope our algorithms and particularly the key technique used in our algorithms, amplitude estimation, can inspire more efficient quantum machine learning algorithms.

ACKNOWLEDGEMENTS

We thank Shijie Pan and Linchun Wan for useful discussions on the subject. This work is supported by the Fundamental Research Funds for the Central Universities (Grant No.2019XD-A01) and NSFC (Grants No.61976024, No.61972048).

The complexity of solving the value of \( (\mathbf{z}^0)^T \Sigma(\mathbf{z}^0) \) is \( O\left( \frac{\log Md}{\varepsilon \epsilon'''} \right) \). The error is shown below:

\[
\text{Appendix A: Liang et al.’s algorithm}
\]

Given a set \( \{ \mathbf{x}^i \}_{i=1}^M \) composed of \( M \) normal quantum training states, where \( \mathbf{x}^i = \sum_j x_j^i \), and \( \mathbf{x}^i = (x_1^i, \ldots, x_M^i)^T \), \( i = 1, \ldots, M \) is a normalized vector. The objective is to detect how anomalous the new quantum state \( \mathbf{x}^0 \) is compared to the normal states. The algorithm is as follows:

Given the control unitaries to create the following superpositions of training states

\[
|0\rangle \sum_{i=1}^{M} |\mathbf{x}^i\rangle |i\rangle + \frac{|1\rangle |\mu\rangle}{\sqrt{2}} \sum_{i=1}^{M} |i\rangle,
\]

where \( |\mu\rangle = \frac{1}{N_\mu} \sum_j \mu_j |j\rangle \), \( \mu_j = \sum_{i=1}^{d} x_j^i \), \( N_\mu \) is normalized coefficient.

Apply \( H \) gate on qubit of the first register to obtain

\[
\frac{1}{2} \left[ |0\rangle \sum_{j=1}^{d} \sum_{i=1}^{M} (x_j^i + \mu_j) |i\rangle |j\rangle + |1\rangle \sum_{j=1}^{d} \sum_{i=1}^{M} (x_j^i - \mu_j) |i\rangle |j\rangle \right].
\]

Measure the first register in \( |1\rangle \), the remaining qubits collapse into the state

\[
\sum_{j=1}^{d} \sum_{i=1}^{M} (x_j^i - \mu_j) |i\rangle |j\rangle = \sum_{j=1}^{d} |\chi_j^0\rangle |j\rangle.
\]

Prepare quantum states \( \sum_{j=1}^{d} \sum_{i=1}^{M} (x_j^0 - \mu_j) |i\rangle |j\rangle = \sum_{j=1}^{d} |\chi_j^0\rangle |j\rangle \) by the same method.

**Algorithm 1.** Efficiently computing \( \sum_{j=1}^{d} \frac{(x_j^0 - \mu_j)^2}{2\sigma_j^2} \).

1. Prepare the quantum state

\[
\sum_{j=1}^{d} |x_j\rangle |\chi_j^0\rangle |j\rangle,
\]

where \( |x_j\rangle = \sum_{i=1}^{d} (x_j^i - \mu_j) |i\rangle \), \( \mu_j = \frac{1}{M} \sum_{i=1}^{M} x_j^i \) and \( |\chi_j^0\rangle = \sum_{i=1}^{M} (x_j^0 - \mu_j) |i\rangle \).

2. Add an ancilla qubit \( |0\rangle \) and perform a controlled unitary operator \( R_1 \) to obtain

\[
\sum_{j=1}^{d} |x_j\rangle |\chi_j^0\rangle |j\rangle (\frac{|x_j^0\rangle}{|x_j\rangle} |0\rangle + \sqrt{1 - (\frac{|x_j^0\rangle}{|x_j\rangle})^2} |1\rangle)
\]


(1.3) Uncompute the second and third registers, the system state is

\[ \sum_{j=1}^{d} |j\rangle (\frac{x^0_j}{\chi_j}|0\rangle + \sqrt{1 - \frac{x^0_j}{\chi_j}^2}|1\rangle) \] (A6)

(1.4) Measure the quantum observation \( M_1 = I \otimes |0\rangle\langle 0| \) and the expectation \( \langle M_1 \rangle = \sum_{j=1}^{d} (\frac{x^0_j}{\chi_j})^2 \).

**Algorithm 2. Efficiently computing \( \sum_{j=1}^{d} \ln \sigma_j \)**

(2.1) Prepare the quantum state \( \sum_{j=1}^{d} |x_j\rangle |j\rangle \).

(2.2) Add an ancilla qubit \( |0\rangle \) and perform a controlled unitary operator \( R_2 \) to obtain

\[ \sum_{j=1}^{d} |x_j\rangle |j\rangle (\ln x_j |0\rangle + \sqrt{1 - 2 \ln x_j} |1\rangle) \] (A7)

(2.3) Uncompute the second register. The system state becomes

\[ \sum_{j=1}^{d} |j\rangle (\ln x_j |0\rangle + \sqrt{1 - 2 \ln x_j} |1\rangle) \] (A8)

(2.4) Measure the quantum observation \( M_2 = I \otimes |0\rangle\langle 0| \) and the expectation \( \langle M_2 \rangle = 2 \sum_{j=1}^{d} \ln x_j \).

If \( \ln P(|x^0\rangle) = -\frac{d}{2} \ln 2\pi - \frac{1}{2} (\langle M_1 \rangle + \langle M_2 \rangle) < \ln \delta \), one can flag \( |x^0\rangle \) as an anomaly quantum state. Otherwise, it is normal.

**Appendix B: Analysis of Liang et al.’s algorithm**

To facilitate understanding, we first introduce controlled rotation [11] [36] and two forms of quantum states corresponding to classical data: digital-encoded state and analog-encoded state [39], as shown below:

(a) Analog-encoded state: Let \( \alpha = (\alpha_1, \cdots, \alpha_N) \) denote a normalized vector, the quantum state \( \sum_{j=1}^{N} \alpha_j |j\rangle \) is called an analog-encoded state.

(b) Digital-encoded state: Let \( \beta = \beta_1 \cdots \beta_n \) be an \( n \)-bit real number, the quantum state \( |\beta_1 \cdots \beta_n\rangle \) is called a digital-encoded state.

(c) Controlled rotation: Let \( x \) denote an \( n \)-bit real number, there exist unitary operation \( U_x \) satisfying

\[ |x\rangle |0\rangle \xrightarrow{U_x} |x\rangle (f(x)|0\rangle + \sqrt{1 - f(x)^2}|1\rangle) \] (B1)

where \( |x\rangle \) is a digital-encoded state, \( |f(x)| \leq 1 \) and the function \( f \) can be computed efficiently on a classical computer.

Then, we analyze and point out the mistakes of Liang et al.’s algorithm, as shown below:

1. In this algorithm, it is claimed that the controlled rotation operations \( R_1 \) and \( R_2 \) can be performed:

\[ \sum_{j=1}^{d} |x_j\rangle |j\rangle (\frac{x^0_j}{\chi_j}|0\rangle + \sqrt{1 - \frac{x^0_j}{\chi_j}^2}|1\rangle) \]

\[ \sum_{j=1}^{d} |x_j\rangle |j\rangle (\ln x_j |0\rangle + \sqrt{1 - \ln x_j^2}|1\rangle) \] (B2)

where \( |x_j\rangle = \sum_{i=1}^{M} (x^i_j - \mu_j |i\rangle |i\rangle \rangle, |x^0_j\rangle = \sum_{i=1}^{M} (x^0_j - \mu_j |i\rangle |i\rangle \rangle \). However, \( |x_j\rangle \) and \( |x^0_j\rangle \) are not digital-encoded state, and \( x_j^i - \mu_j \) is unknown in advance \( (i = 0, 1, \cdots, M, j = 1, \cdots, d) \). This leads to the \( R_1 \) and \( R_2 \) cannot be effectively implemented.

2. It is claimed the quantum states \( \sum_{j=1}^{d} \sum_{i=1}^{M} (x^i_j - \mu_j |i\rangle |i\rangle \rangle \) and \( \sum_{j=1}^{d} \sum_{i=1}^{M} (x^0_j - \mu_j |i\rangle |i\rangle \rangle \) can be prepared (See Eq. (A1) — (A3) in Appendix A). However, \( |\mu\rangle = \frac{1}{N^\mu \sum_{j=1}^{d} \mu_j |j\rangle \rangle \), and performing the \( H \) gate on Eq. (A1) to generate

\[ \frac{1}{2} [ |0\rangle \sum_{j=1}^{d} \sum_{i=1}^{M} (x^i_j + \frac{\mu_j}{N^\mu} |i\rangle |i\rangle \rangle + |1\rangle \sum_{j=1}^{d} \sum_{i=1}^{M} (x^0_j - \frac{\mu_j}{N^\mu} |i\rangle |i\rangle \rangle ] \] (B3)

Then measure the first register in \( |1\rangle \), the Eq. (9) collapse into the state \( \sum_{j=1}^{d} \sum_{i=1}^{M} (x^i_j - \frac{\mu_j}{N^\mu} |i\rangle |i\rangle \rangle \), which is not consistent with the quantum state of \( |0\rangle \). (A3). Similarly, the state \( \sum_{j=1}^{d} \sum_{i=1}^{M} (x^0_j - \mu_j |i\rangle |i\rangle \rangle \) cannot be obtained.

3. It is believed that \( \langle M_2 \rangle = 2 \sum_{j=1}^{d} \ln x_j \). In fact,

\[ \langle M_2 \rangle = \sum_{j=1}^{d} (\ln x_j)^2 \neq 2 \sum_{j=1}^{d} \ln x_j \neq \sum_{j=1}^{d} \ln x_j^2 \] (B4)

This causes the algorithm failing to reach its goal, i.e,

\[ \ln P(|x^0\rangle) = -\frac{d}{2} \ln 2\pi - \frac{1}{2} (\langle M_1 \rangle + \langle M_2 \rangle) \] (B5)
[3] R. Fujimaki, T. Yairi, K. Machida, An approach to spacecraft anomaly detection problem using kernel feature space, Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining (2005).
[4] V. Kumar, Parallel and distributed computing for cybersecurity, IEEE Distributed Systems Online 6.10 (2005).
[5] S. Clay, L. Parra, P. Sajda, Detection, synthesis and compression in mammographic image analysis with a hierarchical image probability model, Proceedings IEEE workshop on mathematical methods in biomedical image analysis (MMBIA 2001). IEEE 2001.
[6] M. Markou, S. Singh, Novelty detection: A review—Part 1: Statistical Approaches, Signal Processing, 83.12 (2003).
[7] H. Hoffmann, Kernel PCA for novelty detection, Pattern Recognition, 40.3 (2007).
[8] P W. Shor, Algorithms for quantum computation: Discrete logarithms and factoring, In proceedings of 35th Annual Symposium on Foundations of Computer Science, ieee, 124-134 (1994).
[9] L. K. Grover, A fast quantum mechanical algorithm for database search, In Proceedings of the twenty-eighth annual ACM symposium on Theory of computing, 212-219 (1996).
[10] A. W. Harrow, A. Hassidim, and S. Lloyd, Quantum algorithm for linear systems of equations, Phys. Rev. Lett. 103, 150502 (2009).
[11] L C. Wan, C H. Yu, S J. Pan, F. Gao, Q Y. Wen, S J. Qin, Asymptotic quantum algorithm for the Toeplitz systems, Phys. Rev. A 97.6, 062322 (2018).
[12] H L. Liu, Y S. Wu, L C. Wan, S J. Pan, F. Gao, S J. Qin, Q Y. Wen, Variational quantum algorithm for the Poisson equation, 104.2 022418 (2021).
[13] C Y. Wei, X Q. Cai, T Y. Wang, S J. Qin, F. Gao, Q Y. Wen, Error Tolerance Bound in QKD-Based Quantum Private Query, IEEE Journal on Selected Areas in Communications, 38, 517-527 (2020).
[14] F. Gao, S J. Qin, W. Huang, Q Y. Wen, Quantum private query: a new kind of practical quantum cryptographic protocols, Sci. China-Phys. Mech. Astron. 62, 070301 (2019).
[15] V. Giovannetti, S. Lloyd, L. Maccone, Quantum Private Queries, Phys. Rev. Lett. 100, 230502 (2008).
[16] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, S. Lloyd, Quantum machine learning, Nature 549, 195-202 (2017).
[17] S. Lloyd, M. Mohseni, and P. Rebentrost, Quantum algorithms for supervised and unsupervised machine learning, arXiv:1307.0411 (2013).
[18] P. Rebentrost, M. Mohseni, S. Lloyd, Quantum support vector machine for big data classification, Phys. Rev. Lett. 113, 130503 (2014).
[19] N. Wiebe, D. Braun, S. Lloyd, Quantum algorithm for data fitting, Phys. Rev. Lett. 109.5, 050505 (2012).
[20] S C. Morampudi, B. Hsu, S L. Sondhi, R. Moessner, Clustering in Hilbert space of a quantum optimization problem, Phys. Rev. A 96.4, 042303 (2017).
[21] P. Rebentrost, T R. Bromley, C. Weedbrook, and S. Lloyd, Quantum Hopfield neural network, Phys. Rev. A 98, 042308 (2018).
[22] G M. Wang, Quantum algorithm for linear regression, Phys. Rev. A 96.1, 012335 (2017).
[23] C H. Yu, F. Gao, Q Y. Wen, An improved quantum algorithm for ridge regression, IEEE Transactions on Knowledge and Data Engineering (2019).
[24] C H. Yu, F. Gao, C. Liu, D. Huynh, M. Reynolds, J. Wang, Quantum algorithm for visual tracking, Phys. Rev. A 99, 022301 (2019).
[25] C H. Yu, F. Gao, Q L. Wang, Q Y. Wen, Quantum algorithm for association rules mining, Phys. Rev. A 94, 042311 (2016).
[26] I. Cong, L. Duan, Quantum discriminant analysis for dimensionality reduction and classification, New J. Phys. 18, 073011 (2016).
[27] S. Lloyd, M. Mohseni, P. Rebentrost, Quantum principal component analysis, Nature Physics, 10, 631 (2014).
[28] S J. Pan, L C. Wan, H L. Liu, F. Gao, S J. Qin, Q Y. Wen, Improved quantum algorithm for A-optimal projection, Phys. Rev. A 102.5, 052402 (2020).
[29] C H. Yu, F. Gao, S. Lin, J. Wang, Quantum data compression by principal component analysis, Quantum Information Processing, 18.8, 1-20 (2019).
[30] N. Liu, P. Rebentrost, Quantum machine learning for quantum anomaly detection, Phys. Rev. A 97, 042315 (2018).
[31] H. Buhrman, R. Cleve, J. Watrous, R D. Wolf, Quantum fingerprinting, Phys. Rev. Lett. 87, 167902 (2001).
[32] M. Schuld, I. Sinayskiy, F. Petruccione, Prediction by linear regression on a quantum computer, Phys. Rev. A 94, 022342 (2016).
[33] J M. Liang, S Q. Shen, M. Li, L Li, Quantum anomaly detection with density estimation and multivariate Gaussian distribution, Phys. Rev. A 99.5, 052310 (2019).
[34] G. Brassard, P. Hoyer, M. Mosca, Quantum amplitude amplification and estimation, Contemporary Mathematics 305 (2002).
[35] V. Giovannetti, S. Lloyd, L. Maccone, Quantum Random Access Memory, Phys. Rev. Lett. 100, 160501 (2008).
[36] B J. Duan, J B. Yuan, Y. Liu, D. Li, Efficient quantum circuit for singular-value thresholding, Phys. Rev. A 98.1, 012308 (2018).
[37] S S. Zhou, T. Loke, J C. Garcia-Escartin, Quantum arithmetic with the quantum fourier transform, Quantum Information Processing 16.3, 82 (2017).
[38] L Ruiz-Perez, J C. Garcia-Escartin, Quantum algorithms for the poisson equation, 104.2 022418 (2021).