Quantum Machine Intelligence (2024) 6:26
https://doi.org/10.1007/s42484-024-00163-2

RESEARCH ARTICLE

Quantum deep learning-based anomaly detection for enhanced network security

Moe Hdaib[1] · **Sutharshan Rajasegarar**[1] · **Lei Pan**[1]

Received: 30 October 2023 / Accepted: 11 April 2024

Abstract
Identifying and mitigating aberrant activities within the network traffic is important to prevent adverse consequences caused by cyber security incidents, which have been increasing significantly in recent times. Existing research mainly focuses on classical machine learning and deep learning-based approaches for detecting such attacks. However, exploiting the power of quantum deep learning to process complex correlation of features for anomaly detection is not well explored. Hence, in this paper, we investigate quantum machine learning and quantum deep learning-based anomaly detection methodologies to accurately detect network attacks. In particular, we propose three novel quantum auto-encoder-based anomaly detection frameworks. Our primary aim is to create hybrid models that leverage the strengths of both quantum and deep learning methodologies for efficient anomaly recognition. The three frameworks are formed by integrating the quantum autoencoder with a quantum one-class support vector machine, a quantum random forest, and a quantum k-nearest neighbor approach. The anomaly detection capability of the frameworks is evaluated using benchmark datasets comprising computer and Internet of Things network flows. Our evaluation demonstrates that all three frameworks have a high potential to detect the network traffic anomalies accurately, while the framework that integrates the quantum autoencoder with the quantum k-nearest neighbor yields the highest accuracy. This demonstrates the promising potential for the development of quantum frameworks for anomaly detection, underscoring their relevance for future advancements in network security.

Keywords Quantum machine learning · Quantum autoencoder · Quantum anomaly detection · Quantum computing

1 Introduction

Quantum computing possesses unique attributes, especially in terms of parallelism, featuring the trend of future computing. Different from the binary digits used in classical computing, quantum bits (qubits) used in quantum computing leverage the phenomenon of superposition to allow the concurrent representation of 2^n classical bits, where n is the dimension. This parallelism allows for thorough investigations of the varied possibilities, raising the level of the computation (Alchieri et al. 2021; Kulkarni et al. 2021). Combining quantum computing with machine learning provides high potential for many applications, such as image recognition (Hashemzehi et al. 2020), protein folding (Lu and Li 2019), and fraud detection (Rizvi et al. 2020; Kyrienko and Magnusson 2022).

Currently, the emerging field of quantum machine learning (QML) has seen many related works on accelerating processing speed and resolving the issue of data dimensionality (Alchieri et al. 2021; Kulkarni et al. 2021). The application of quantum autoencoders (QAEs) is not limited to simulating the complex quantum systems (Pu et al. 2016; Bartůšková et al. 2006), but they are also applied in other areas, such as communication and distributed computation (Steinbrecher et al. 2019; Lamata et al. 2018; Aspuru-Guzik et al. 2005).

QML has been rapidly evolving in pattern recognition areas (Trugenberger 2002). However, the QML's ability to enhance the anomaly detection capability for network secu-

Sutharshan Rajasegarar and Lei Pan contributed equally to this work.

✉ Moe Hdaib
mhdaib@deakin.edu.au

Sutharshan Rajasegarar
srajas@deakin.edu.au

Lei Pan
l.pan@deakin.edu.au

[1] School of Information Technology, Faculty of Science, Engineering and Built Environment, Deakin University, Geelong, VIC 3220, Australia

Published online: 02 May 2024

rity is not well studied. Challenges still exist in devising practical solutions for cyber security challenges because of the growing data complexity and their evolving nature. By exploiting the synergy between the quantum and classical technologies, we aim to achieve this by devising novel hybrid models. Specifically, we propose three quantum auto-encoder-based frameworks for anomaly detection. Our proposed methodologies encompass a hybrid architecture of parameterized quantum circuits merged with deep neural networks, thus uniting quantum- and classical computing principles to identify anomalies. In the past, quantum encoding autoencoders (QAEs) were used to compress quantum states (Ding et al. 2019; Pepper et al. 2019; Huang et al. 2020). In this work, this is exploited and integrated with three components, namely quantum one-class support vector machine, quantum random forest, and quantum k-nearest neighbor, to enable effective anomaly detection.

Moreover, QAEs encode features in the shape of single-qubit rotation gates (Bravo-Prieto 2021). QAEs' efficiency provides a clear advantage over other architectures by simplifying data representation and shortening the specifications of quantum communication channels (Steinbrecher et al. 2019), and the complexity of quantum gates (Lamata et al. 2018; Ding et al. 2019). In addition to data compression, QAE has been used successfully for error reduction (Zhang et al. 2021), order detection (Srikumar et al. 2021), and quantum state compression (Ding et al. 2019) tasks. Through the integration of QAEs into our methodology, we take QAEs' advantages for anomaly detection to improve the efficiency and accuracy of network attack detection.

The contributions in this paper are as follows.

- We explore the current state of QML in the context of cyber security anomaly detection, specifically focusing on network traffic. We propose quantum algorithms to improve the detection of anomalies from network traffic information obtained from computer or IoT networks.
- We introduce three novel frameworks for anomaly detection using QML and QDL in conjunction with autoencoders. In particular, we integrate quantum autoencoders with quantum one-class support vector machine, quantum random forest, and quantum k-nearest neighbors, respectively. We distinguish ourselves by introducing innovative technical methods, with a strong emphasis on our encoding strategies. This introduction sets the stage for our exploration into the efficacy of these proposed techniques in anomaly detection.
- Using NISQ quantum computers and IBM quantum simulators, our evaluation reveals that all three proposed frameworks improve the anomaly detection performance compared to the classical counterparts on the benchmark datasets. In particular, QAE with quantum kNN performs the best among the three frameworks.

The remainder of the paper is structured as follows: The related work is presented in Sect. 2, and the anomaly detection applications of autoencoders and quantum autoencoders are introduced in Sect. 3. Section 4 proposes three quantum frameworks. Section 5 details the experiment setup, Sect. 6 describes the three datasets used in this paper, before Sect. 7 presents the results and discussion. Finally, this paper is concluded in Sect. 8.

2 Related work

Quantum neural networks (QNNs) are formed using parametrized quantum circuits and classical neural networks. Like their classical predecessors, QNNs are algorithmic models that can be taught to uncover hidden patterns in the data. These models have the ability to load classical data (inputs) into a quantum state and then process it using trainable weight-parameterized quantum gates. A generic QNN loads data using a feature map and performs processing steps using ansatz, where each data is assigned to a weight value. The weights can then be trained using backpropagation by feeding the measurement's output into a loss function. The QNNS can be trained variationally using conventional optimizers (Li and Deng 2022).

In Mangini et al. (2022), a quantum machine learning approach is used to address an industrial quality control problem. It consists of a quantum neural network (QNN) in combination with a classifier and an autoencoder. The QNN autoencoder performs dimensionality reduction and extracts features, whereas the QNN classifier is utilized to gauge the quality of industrial samples. A suitable encoding technique is required to convert the classical data into quantum states. After encoding, a parameterized quantum circuit is used to learn a lower-dimensional representation of the input data before feature extraction.

In Wang and Jiang (2022), the QNN-based method is used for reconstructing missing or erroneous data. Three components make up this QNN-based method, namely i) data encoding, which converts input data into a quantum state; ii) a quantum neural network, which explains the relationship between the available data and the corrupted or missing values; and iii) a parameterized quantum circuit. The initial quantum state, sometimes referred to as state preparation, needs to be produced in accordance with the classical characteristics prior to using a QNN on classical information. In almost all quantum algorithms, this initialization of the input qubits to a proper starting state is a crucial step (Zhang et al. 2022).

In Mete et al. (2021), using quantum autoencoders, the Hamiltonian dynamics is modeled. The procedure reduces computer resource requirements due to the utilization of the fundamental design of the physical system. It is challenging

to model the computationally efficient temporal evolution of quantum systems governed by Hamiltonians due to the exponential growth of the system's Hilbert space, which is a major obstacle in quantum computation. In their work, before being decompressed by a Hamiltonian simulator employing the inverse of the quantum encoders, the quantum state is compressed using quantum autoencoders. The main objective of the use of quantum autoencoders here is for dimensionality reduction (Romero et al. 2017).

The powerful parallel computing ability of quantum computers is leveraged in Dang et al. (2018) to boost the efficiency of image classification. A quantum k-nearest neighbor algorithm is used to classify images by computing similarity between feature vectors of images. The feature vectors are initially extracted on classical computers before being converted into a quantum superposition state. The quantum minimum search algorithm is used to speed up the search process for similarity, and the image is classified by quantum measurements. This quantum algorithm's complexity is only $\mathcal{O}((kM)^{(1/2)})$, which is superior to classical algorithms. While achieving similar accuracy scores, the quantum algorithm has significantly reduced the computation time compared to the classical algorithms.

Classical deep learning methods are popular for anomaly detection. Some of them include variational autoencoders (VAE) (An and Cho 2015; Pol et al. 2019), one-class methods, such as support vector data description (SVDD) (Tax and Duin 2004) that encircle normal data points within a hypersphere in a replicating Hilbert space, providing a detailed representation of the data (Ruff et al. 2018; Rajasegarar et al. 2010) and multiclass anomaly detectors (Shilton et al. 2020). Additionally, the LAKE approach proposed in Lv et al. (2020) combines VAE with kernel density estimate (KDE) to enhance anomaly detection. Nevertheless, there have been few implementations of QDL in anomaly detection.

3 QML and QDL for anomaly detection

Detecting unexpected behavior that deviates from expected or usual behavior is the process of anomaly or outlier detection. It is especially important to detect the anomalies in the networks as that might indicate the formation of an unexpected phenomenon in the network, such as a malfunctioning system or an aggressive security attack. It is critical to automatically detect the outliers when the bulk of the input data comes from sources that are unclear or of dubious reliability. Learning to correctly identify the outliers from data, mostly consisting of normal instances, is the goal of anomaly detection. Kyriienko and Magnusson (2022) observed that the quantum methods perform better in QML anomaly detection than conventional computer techniques.

We investigate the feasibility of combining a deep neural network and a parameterized quantum circuit to tackle the anomaly detection problem. Our objective is to develop hybrid models that combine the best features of quantum and deep learning techniques to deliver efficient anomaly identification.

Anomalies are often defined based on the fact that they are uncommon and very distinct from usual (normal) points. In contrast to most analytical and learning tasks, anomaly detection presents several unique challenges. The challenges are listed below, including unknownness, heterogeneous anomaly classes, rarity and class imbalance, and diverse types of anomalies (Pang et al. 2021).

- Anomalies are characterized by unknown factors, such as instances with unexpected abrupt behaviors, diverse data structures, and varying distributions. Examples of these anomalies include innovative attacks, fraud, and network intrusions. Anomalies often remain unknown until they actually occur.
- Anomalies may exhibit distinct unusual characteristics across different classes due to their erratic nature. For example, network data and its structure differ significantly in the case of network attacks, making identifying anomalous attack patterns challenging.
- In contrast to normal instances, which typically make up most of the data, anomalies are infrequent occurrences. Consequently, collecting a substantial number of labeled anomalous instances is a difficult task, if not impossible. Large-scale labeled data is often unavailable in most applications. Misclassifying anomalies is typically more costly than misclassifying typical (normal) instances.

Deep learning algorithms have gained popularity in anomaly identification due to their excellent accuracy (Pang et al. 2021; Erfani et al. 2016; Zhang et al. 2021; Ruff et al. 2018). However, only a limited prior work exists that performs QDL-based anomaly detection for network traffic attacks. One of the works in Gouveia and Correia (2020) compared a quantum support vector machine (QSVM) and a traditional support vector machine (SVM) to compare its performance in classifying the attack classes using the NSL-KDD (Shiravi et al. 2012) dataset, an enhanced variant of the KDD99 dataset (MIT Lincoln Labs 1998). While the standard SVM obtained an accuracy score of 93%, the reported QSVM accuracies in the simulation were 92%. Interestingly, only 150 data samples were used in the training procedure. According to Gong et al. (2022), its model evaluation was conducted on a quantum computer. Five features were chosen out of the initial 41 features. A z rotation was used after a Hadamard gate to encode the characteristics. This shows the potential of the use of a quantum mechanism for anomaly detection, however, the above method used a super-

vised technique to classify the attacks. In contrast, in this paper, we focus on devising an unsupervised anomaly detection methodology to detect security attacks.

We propose three frameworks based on QML and QDL to address the challenges associated with detecting anomalies in network and IoT traffic. Next, we briefly introduce the autoencoder and the quantum autoencoder that form the main component of our proposed framework.

3.1 Autoencoders

Autoencoders are neural networks that have been taught to reconstruct their inputs (McClelland et al. 1987) as accurately as possible. Their primary objective is to create an "informative" representation of the data in an unsupervised way that may be utilized for numerous implications, such as anomaly detection and dimensionality reduction. Figure 1 shows a graphical depiction of an autoencoder. The input data undergoes an encoding phase that generates a distinct representation (embedding) of the data with smaller dimensions, as illustrated in the latent space. Subsequently, the data proceeds through a decoder stage where it is reconstructed as accurately as possible to the original input.

Classical autoencoders, which are neural networks capable of efficiently training low-dimensional representations of data in higher-dimensional spaces, serve as the foundational inspiration for quantum autoencoders. An autoencoder maps an input x to a lower-dimensional point y with the aim of enabling the potential recovery of x from y. By altering the underlying autoencoder network's architecture to represent the data in a reduced dimension, the input may be efficiently compressed (Ranzato et al. 2007).

A linear autoencoder (Baldi and Hornik 1989) has its encoder and decoder components comprising linear operations. If the autoencoder were linear, it would achieve the same latent representation as principal component analysis (PCA) (Plaut 2018). Since the autoencoder usually learns a non-linear manifold instead of locating a low-dimensional hyperplane in which the data lies, an autoencoder is a generalization of PCA.

Autoencoders operate based on the fundamental principle of identifying the optimal encoding-decoding scheme through an iterative optimization process by employing encoder and decoder neural networks. In this process, data is first fed to the encoder component of the autoencoder architecture. Subsequently, the output of the encoder is fed to the decoder to reconstruct the input. The decoder output is compared with the input data, and the error is propagated backward through the architecture to update the network weights using a technique known as backpropagation. Essentially, the combined encoder-decoder design of the entire autoencoder ensures that only the primary structured information can pass through and be reconstructed. Gradient descent is employed to optimize the encoder and decoder configurations with a minimal reconstruction error.

When employing autoencoders for dimensionality reduction, it is important to bear in mind that achieving a substantial reduction in dimensionality without incurring reconstruction loss often comes at a cost. Namely, the hidden space, or the reduced-dimensional representation, may frequently lack usable and comprehensible structures. Additionally, it is essential to recognize that the primary objective of dimensionality reduction is usually not merely to reduce the number of data dimensions, but to do so while retaining the majority of the underlying data structure's information within the smaller representations.

3.2 Quantum machine learning

Information processing through quantum system manipulation is the goal of quantum computing. Since operations can be carried out on numerous states in parallel at once, the superposition feature of quantum states can dramatically reduce computation complexity.

The qubit, $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ where $|0\rangle, |1\rangle$ existing in the two-dimensional Hilbert space H^2, serves as the fundamental unit of quantum computation. The probability of measuring a qubit in the 0 or 1 state is determined by the absolute squares of its amplitudes. Quantum dynamics upholds the principle of probability conservation, expressed as $|\alpha|^2 + |\beta|^2 = 1$. Mathematically, transformations mapping quantum states onto other quantum states, known as quantum gates, must be unitary.

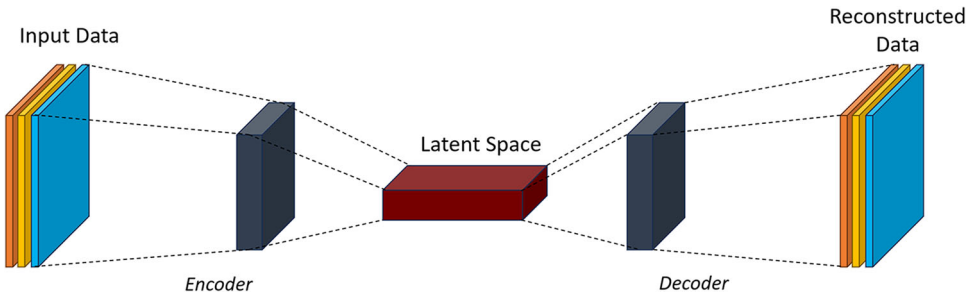


Fig. 1 A graphical representation of an autoencoder. The encoder block encodes the input data into a smaller dimension latent space. The Decoder block subsequently makes an effort to reconstruct the input data at the output as accurately as possible

Single-qubit quantum gates enable us to manipulate the basis state, amplitude, or phase of a qubit. For instance, the X-gate, Z-gate, and Y-gate manipulate these properties, respectively. Additionally, the Hadamard (H)-gate transforms a qubit with $\beta = 0(\alpha = 0)$ into an equal superposition of $\alpha = \beta = 1/\sqrt{2}(\alpha = 1/\sqrt{2}, \beta = -1/\sqrt{2})$.

Multi-qubit gates frequently involve controlled operations that execute a single-qubit operation only when another qubit (known as the ancilla or control qubit) is in a specific state. Among these gates, the two-qubit XOR-gate is particularly significant, as it flips the basis state of the second qubit when the first qubit is in the $|1\rangle$ state. Another notable two-qubit gate is the SWAP-gate, which exchanges the states of two qubits.

Quantum gates are typically represented as unitary matrices that act on two n-dimensional vectors. These vectors encompass the amplitudes of the 2^n basis states of a n-dimensional quantum system. In order to create a quantum state with a relatively large amplitude for states that represent solutions to the given issue, designing quantum algorithms requires the use of such relatively straightforward gates. A measurement in the computational base yields the desired result with an elevated probability.

Quantum algorithms are probabilistic, thus in order to reduce errors, they are executed multiple times. Readers are referred to Nielsen and Chuang (2010) for a thorough introduction to quantum computing. In the realm of quantum machine learning, innovative quantum algorithms are being developed to tackle typical machine learning problems by leveraging the power of quantum computing. These algorithms are often derived from classical algorithms, or their computationally intensive subroutines, adapted to run on potential quantum computers. In the foreseeable future, these machines could become widely available for practical applications, enabling the efficient processing of the rapidly growing volumes of global information.

Furthermore, the quantum algorithms developed enable known machine learning techniques to further enhance and improve quantum information theory. These techniques can be applied to find "quantum decision functions" or "quantum strategies," or to optimize system parameters like unitary operators. Challenges exist on how to possibly create and apply effective quantum learning processes. One of the main obstacles is discovering how to use coherent and reversible quantum computers to efficiently execute optimization tasks, which are usually handled using dissipative and iterative techniques like gradient descent. Furthermore, it is necessary to investigate the use of quantum states in the translation and processing of significant structural information, such as distance metrics. Addressing these issues in the future will enhance the decision-making capability within the context of quantum mechanics.

3.3 Quantum autoencoders

Information compression is an essential issue in information theory, and in the realm of quantum computing, quantum autoencoders (QAEs) have been introduced as viable means. QAEs aim to compress quantum states into a low-dimensional representation. They provide a framework to perform machine learning tasks on quantum systems without incurring the exponential memory costs associated with traditional methods. Because the number of factors needed to adequately represent a quantum state grows exponentially, classical computers encounter difficulties when working with quantum systems. The classical memory required to store and analyze the essential data expands exponentially with the scale of a quantum system, making it computationally intensive and sometimes unworkable (Romero et al. 2017).

The utilization of quantum encoders for our proposed anomaly detection framework enables us to minimize the dimension of quantum data and accomplish similar machine learning tasks for quantum systems without the need for exponentially expensive conventional memory (Romero et al. 2017). QAEs are particularly useful for compressing datasets of quantum states when conventional compression techniques are not practical. Furthermore, the parameters of a QAE can be learned through traditional optimization methods.

The implementation of the QAE closely parallels that of classical autoencoders, with the distinction that both the data and operations are governed by quantum mechanics. A QAE's fundamental function is to encode an input state $|\boldsymbol{\psi}\rangle$ to a quantum state of reduced dimension $|\boldsymbol{\phi}\rangle$ before decoding $|\boldsymbol{\phi}\rangle$ back to $|\boldsymbol{\psi}\rangle$.

A quantum circuit is parameterized in our QAE, and we attempt to find the best combination of parameters to reduce the discrepancy between the state at the input and the state after reconstruction. The procedure of encoding can be expressed as $\boldsymbol{U(\theta)|\psi\rangle = |\phi\rangle}$, where $|\boldsymbol{\psi}\rangle$ is the input state, $\mathbf{U}(\boldsymbol{\theta})$ is the unitary transformation representing the quantum circuit parameterized by $\boldsymbol{\theta}$, and $|\boldsymbol{\phi}\rangle$ is the encoded (compressed) state. The method of decoding can be represented as $\boldsymbol{U^{'}(\theta)|\phi\rangle = |\psi^{'}\rangle}$, where $|\boldsymbol{\phi}\rangle$ is the encoded state, $\boldsymbol{U^{'}(\theta)}$ is the decoding unitary transformation, and $|\boldsymbol{\psi^{'}}\rangle$ is the reconstructed state. By representing the data on a lower dimension by changing the underlying autoencoder network's topology, the input can be effectively compressed (Romero et al. 2017).

In this scenario, data compression is not inherently achieved since unitary transformations maintain probabilities and typically operate on spaces with similar dimensions. To attain data compression, some qubits in the initial encoding stage are intentionally excluded and replaced with freshly generated reference states. This setup is particularly well-suited for a three-dimensional latent space and a six-feature

input, as depicted in Fig. 2. Three of the outputs of the unitary operators are replaced with freshly formed reference states that are ignorant of the input states during the encoding process, even though they yield the same amount of qubits. For a more comprehensive understanding of the principles of quantum autoencoding and data compression, the reader can refer to the details of a QAE in Romero et al. (2017).

Our quantum autoencoder uses quantum circuits to build its encoder (E-block) and decoder (D-block) (Romero et al. 2017). Because these circuits are parameterized, it is possible to optimize them during the training process. The goal of this optimization is to minimize a cost function associated with the fidelity of the reconstruction quantum states. Figures in the subsequent sections will show the precise arrangement of ansatzes (parameterized quantum circuits) in the E- and D-blocks.

Shortening the wavefunction representation in quantum autoencoders is driven by pragmatic factors. When the number of qubits in a quantum system increases, the amount of entanglement may become too high to compute the entire wavefunction. The wavefunction can be truncated to improve the efficiency of managing computing resources and to lessen the effects of the quantum state's exponential expansion.

The purpose of this truncation is to safeguard against overfitting and improve the model's capacity for generalization, similar to the regularization approaches used in classical machine learning. Furthermore, truncating the wavefunction representation in the framework of quantum autoencoders avoids certain problems, such as the saturation of probability amplitudes in high-dimensional spaces, which can affect optimization algorithms' performance. Additionally, the identification of quantum anomalies in simulated quantum states was investigated in Kottmann et al. (2021).

In this paper, we focus on input states that are fundamentally quantum mechanical. The theoretical framework known as quantum mechanics explains how matter and energy behave at the quantum level. When comparing the quantum states to classical ones, the former have distinct computing capabilities due to their superposition and entanglement.

Another fact that we considered in the proposed framework is the embedding of classical data that determines the nature of the resulting quantum state. Proposing a hybrid classical-quantum approach highlights the part that classical data plays in forming the quantum state. Many quantum machine learning models assume that pure quantum data are readily available for training and testing the model. This work emphasizes how crucial classical facts are used to derive the final quantum state.

For real-world applications where classical data is large, this hybrid technique could offer greater flexibility and practicality, facilitating a more seamless transition into quantum-enhanced processing. Because the quantum state depends on classical data embedding, it will be possible to create customized quantum representations using certain classical datasets, which might optimize quantum information processing for various applications.

With its focus on the quantum nature of input states, the paper provides a way to investigate the benefits of quantum computing, allowing it to outperform its classical equivalents in particular tasks. A practical integration of quantum approaches into current classical machine learning frameworks is suggested by the use of classical data to shape the quantum state. Because there is a lot of classical data available, the hybrid classical-quantum technique could make the switch to quantum-enhanced computation much easier. Quantum autoencoders, programmable circuit methods, and unsupervised anomaly detection together represent a complete and flexible quantum machine learning approach.

Creating quantum representations that balance between encoding latent information and offering a reference state for a variety of data occurrences is made easier with the help of the organized subsystem splitting technique. During the angular encoding process, the qubits that compose these quantum states can be split. To entangle these distinct qubits, we employ CNOT gates all across the unitary development process (see Appendix for an introduction of CNOT gates and other gates).

An unsupervised challenge of notable importance is the identification of anomalies. Specifically, it involves training a model exclusively on normal examples to establish a normal profile, which can then be used to detect samples that deviate from this normal profile as anomalies. In this work, we posit that a trained autoencoder will effectively learn the latent subspace associated with normal samples.

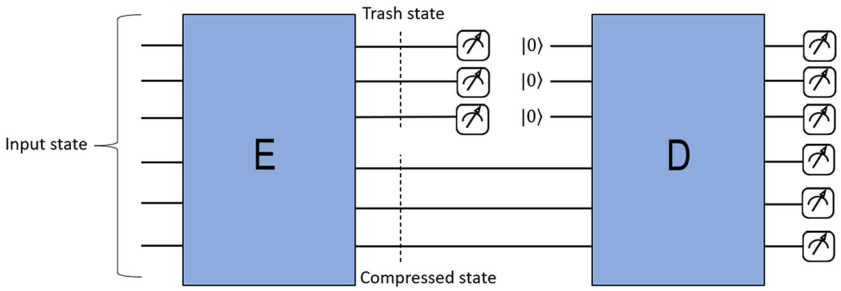


Fig. 2 Quantum autoencoder circuit illustration. It consists of 6 input states, 3 compressed states, and the other 3 trash states, then it is reconstructed back to 6 states. (The diagram is adapted from Romero et al. (2017))

It is important to note that the quantum autoencoder (QAE) is a hybrid approach that combines both classical and quantum techniques. After preparing the input state, a parameterized unitary operation is employed to compress the state. To assess the degree of overlap between the reference state and the portion that was discarded during compression, a SWAP test is utilized. A classical optimization approach is used to create the cost function.

The optimal compressed state is determined through a SWAP-gate involving the discarded portion and the reference state. These states typically contain fewer qubits, making it simpler to compare them with fewer gates. Achieving maximum fidelity between the discarded and reference states is essential for identifying the optimal compression for our input circuit. We fine-tune our encoder's settings and then run a SWAP test to evaluate how well these discarded states match reference states during the training phase. This involves introducing an additional qubit, which serves as an auxiliary qubit for measuring the overall fidelity between the discarded and reference states during the SWAP test.

An approach to variational quantum circuits is used to develop the model. Therefore, by gradually learning, the ideal settings for the quantum gates may be determined. Only unitaries with $2^n \times 2^n$ dimensions may produce such gates, which provide a n-qubit unitary gate. On the other hand, it produces exponentially more parameters in relation to the qubit count, making the optimization process intractable. Therefore, we utilize the programmable circuit technique described in Romero et al. (2017) to decode the huge unitary into single-qubit rotation gates and CNOTs.

Following the selection of the programmable circuit, the defined model serves as an encoder and the structural basis of the design is formed. Unlike regular autoencoders, a quantum autoencoder's decoder can be the inverse of the encoder, whose decoder is learned from the ground up. For quantum autoencoders, this is possible since unitary matrices can be efficiently inverted. Since the encoder as a whole will provide a unitary, if we characterize the encoder network as $U^{\overrightarrow{p}}$, where $\overrightarrow{p}$ represents the ideal network parameters, then the decoder network can be represented as $(U^{\overrightarrow{p}})^{'}$.

In order to realize the quantum encoder, two subsystems, A and B, are formed. In subsystem A, the encoder generates a "latent code" that may be used to reconstruct the input later on. But for subsystem B, the goal is to establish an ideal "reference state" for all possible data occurrences. If the encoding is finished to a high degree of accuracy, the identical reference qubits might be placed in the latent space to replicate the output. In our design, we have selected $|0\rangle$ as the reference state for simplicity; the number of qubits in this state might vary depending on the size of the latent space. Therefore, after the encoder, subsystem A must contain the latent code, and subsystem B must generate the state $|0\rangle$ for all supplied input values.

One method to achieve this is to apply a sequence of SWAP-gates between the subsystems B and $B^{'}$, which comprise the reference state ansatz. It means that if the network creates a latent space, it will generate an input by switching the fixed reference state into the subsystem B. Creating a loss function for the training of the variational circuit is the final part of the model. As the traditional loss function for autoencoders, the $L2$ norm of the input and output may be transformed into QAE in the manner described below:

$$C_1\left(\overrightarrow{P}\right) = \sum_i p_i . F\left(|\psi_i\rangle, \rho_{i,\overrightarrow{p}}^{out}\right) \tag{1}$$

$$F\left(|\psi_i\rangle_{AB} \otimes |a\rangle_{B'}, U_{AB}^{'} V_{BB'} U_{AB} |\psi_i\rangle_{AB} \otimes |a\rangle_{B'}\right), \tag{2}$$

where the parameterized unitaries at subsystems A and B are described by the density matrix $\rho_{i,p\rightarrow}^{out}$, the reference state is denoted by $|a\rangle$, and the unitary of the SWAP-gate is represented by V. The cost function is defined in Eq. 1 by the degree to which the output, which is a reconstruction of the input, resembles the original input, $|\psi\rangle$. At this point, only subsystems A and B need to be monitored because the subsystem $B^{'}$ has been traced out. The *fidelity* of quantum systems is a metric to measure the similarity of such states. Therefore, we define a successful autoencoding as one that $F\left(\psi_i, \rho_i^{out}\right) \approx 1$ for all input states. It is possible to ascertain whether the states are pure by examining their inner product, as shown in Eq. 3:

$$F(\rho, \sigma) = |\langle\psi_A|\psi_B\rangle|^2, \tag{3}$$

where $\rho = |\psi_A\rangle\langle\psi_A|$ and $\sigma = |\psi_B\rangle\langle\psi_B|$. A further simplification of the cost function in Eq. 1 yields the following findings:

$$C_2\left(\overrightarrow{p}\right) = \sum_i p_i . F\left(Tr_A\left[\overrightarrow{U}|\psi_i\rangle\langle\psi_i|_{AB}\left(\overrightarrow{U}\right)^{'}\right], |a\rangle_B\right). \tag{4}$$

The results obtained from the original cost function and the reduced cost version are identical, indicating the degree of accuracy between the reference state and the subsystem B's anticipated value following the encoder. It can trace out the subsystem A by avoiding measuring its qubits. Since the measured component must match the set reference state perfectly for every possible input state, it is also referred to as the "*trash state*". The leftover traced-out qubits generate a "compressed state" or "latent space" that may be stored or utilized for further inference or learning tasks. During testing, we can measure subsystem B and acquire the latent space state instead of keeping an eye on subsystem A. Instead of measuring the subsystem, it would be necessary to carry out the

desired procedure progressively after the encoding to prevent irreversibly altering the state, as the compressed state may be entangled.

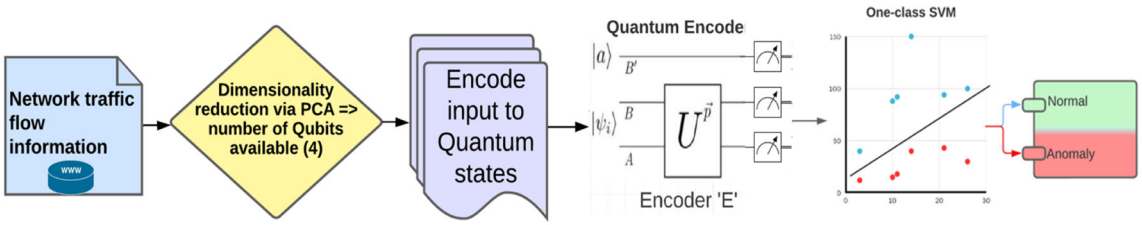
4 Proposed frameworks

We propose three approaches for anomaly detection with quantum autoencoders. A quantum autoencoder and a one-class SVM are integrated in the first framework; a quantum autoencoder and a quantum random forest are integrated in the second; and a quantum autoencoder and a quantum k-nearest neighbor are combined in the third.

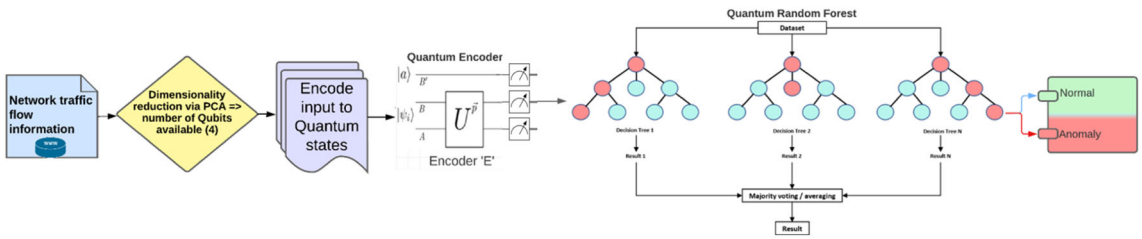
An autoencoder can be trained on a dataset of "normal" data samples. A naive approach to determining an anomaly is to compute the difference between the original sample and its reconstruction, or the reconstruction error, which can be used to determine how "normal" or "anomalous" the sample is. Nevertheless, when we tested this approach, it performed worse than the proposed frameworks, where it scored only 75% accuracy and 77% F1-score. Therefore, before feeding the encoded data as an input to several quantum machine learning classification algorithms, employing the quantum autoencoder as a dimensionality reduction technique, as used in our proposed frameworks, improved detection accuracy and attained faster training times.

Figure 3a illustrates the first framework. Prior to the data being processed in the quantum autoencoder (QAE) quantum circuit, the data input is encoded to quantum states. This is done by first utilizing a PCA (principle component analyzer) to decrease the dimensionality of the network traffic data, where the number of features matches the number of available qubits. Following a successful encoding process using the quantum autoencoder as a dimensionality reduction approach, the data is mapped via a kernel before being used as input for a one-class support vector machine (SVM), which employs a hyperplane to distinguish between normal and anomalous data. Two one-class SVM techniques-one using a *classical kernel* and the other using a *quantum kernel* have been tested.

Figure 3b shows the second framework. The process entails dimensionality reduction with the quantum autoencoder, followed by the data being fed into a quantum random forest algorithm (a machine learning ensemble approach) that combines many variational quantum classifier (VQC) models to provide predictions. Following a predefined number of



(a) Framework 1: Union of QAE and one-class SVM.



(b) Framework 2: Union of QAE and quantum random forest.

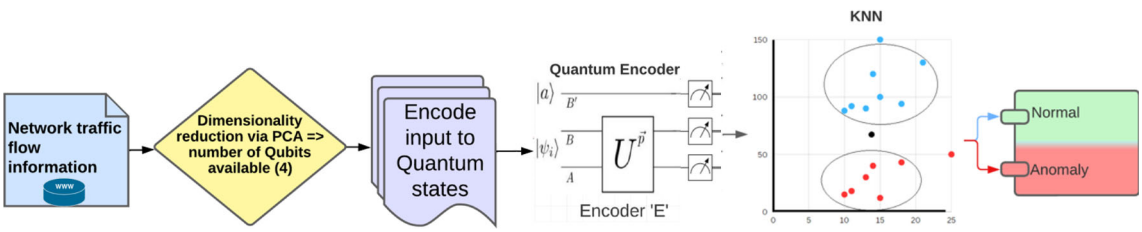


Fig. 3 Three novel quantum autoencoder frameworks for anomaly detection. The figures describe a process for analyzing network traffic to identify anomalies using quantum autoencoders for dimensionality reduction and one-class SVM, quantum random forest, and quantum kNN for classification

VQC models (trees) being trained, the function uses majority voting to provide predictions for the test data input.

Figure 3c illustrates the third framework. The steps involve using the quantum autoencoder as a dimensionality reduction technique and then the data is fed as input to a quantum k-nearest neighbors component, which categorizes a test state that is unknown by locating its k closest neighbors among a set of M train states. Fidelity serves as the similarity metric in quantum kNN. The primary purpose is to encode as quantum state amplitudes the fidelity information between the test state and all train states. We complete the reduction by converting this amplitude-encoded information to a digital representation, allowing us to compare them effectively.

5 Experiments

5.1 Setup

The experiments were conducted using real NISQ quantum devices and a quantum simulator. We utilized Jupyter notebooks to design and simulate our quantum circuits using PennyLane with Python 3.9 and the IBM Qiskit Python library version 0.37. In the experiments, we used IBM quantum labs to access quantum computers and quantum simulators. Matplotlib is used for visualization, PyTorch is used for cost function optimization, and Scikit-learn is used for evaluation. Our investigation was carried out using an Apple M1 Pro ARM-based system, which has a 10-core CPU for processing and a 16-core GPU.

5.2 Data preprocessing

In the data processing step, we utilize a set of methods, including one-hot encoding, undersampling, normalization, and PCA dimensionality reduction.

- To express nominal or categorical data as binary vectors, one-hot encoding is used. A binary vector of 0 s and 1 s is created for each category in one-hot encoding, with the vector's length being equal to the entire number of categories. Each vector contains 0 s everywhere else and a value of 1 at the index that corresponds to the category. This ensures that no numerical correlation between categories is implied.
- Undersampling is a technique for balancing unequal datasets by keeping all of the data in the minority class and decreasing the size of the majority class. We used undersampling to extract accurate data from datasets that were previously uneven.
- The min-max normalization was used to normalize our data. It is used to scale numerical data to a predetermined range of values. By deducting the feature's minimum value from each value and then dividing the result by the feature's range, this method scales the data values to a range between 0 and 1.
- We used principal component analysis (PCA), which converts high-dimensional data into a lower-dimensional space while preserving as much information as feasible. When working with datasets that contain numerous features or strongly linked variables, PCA is especially helpful. In this paper, we utilized PCA to reduce the dimensionality of the dataset to match the number of qubits we have access to.

Finally, the datasets were split as 80% training and 20% testing and used in the evaluations.

5.3 Quantum-based anomaly detector implementations

We used Python to implement the quantum autoencoder. We used datasets from KDD99 (MIT Lincoln Labs 1998), IoT-23 (Garcia et al. 2020), and CIC IoT 23 (Neto et al. 2023) for the evaluations. Training the parameters of the quantum autoencoder is done by traditional optimization techniques. A key component of our research is the conversion of odd-dimensional classical data to a quantum state. A common mitigation strategy is to apply feature selection or dimensionality reduction. In order to achieve this, a subset of the most pertinent characteristics are chosen, or the features are changed to create a new set such that the total number of features is a power of 2. Here, principal component analysis (PCA) was employed.

Quantum embeddings are used to encode a collection of features into the quantum state once the features are powers of two. AngleEmbedding and AmplitudeEmbedding from PennyLane were utilized in our approach to achieve this. Both methods adhere to the principle of $n = \log_2 N$ qubits.

We combine the classical and quantum approaches in our quantum autoencoder. The parameterized unitary is used to compress the state once the input state has been produced. A SWAP test is used to measure how much of the trash state and the reference state overlap via compression. A conventional optimization strategy is utilized to generate the cost function that is minimized from the outcomes for all the states in the train data.

Our work primarily differs from other existing implementations in that the input states of those other implementations are essentially quantum mechanical. On the other hand, the kind of quantum state in our study is determined by the input embedding of the classical data. The qubits that comprise the quantum states may be divided in angular encoding. But

to entangle the different qubits, we use CNOT gates in the unitary evolution.

Hadamard gates and controlled rotating gates are components of the AE quantum circuit. `PennyLane`, a quantum Python library, is used in its implementation. Angle embedding and amplitude embedding are two aspects of the quantum process. The QAE circuit SWAP test is represented by the circuit in Fig. 4, which consists of 4 total qubits. The compressed qubits (2 and 3) go through an angle embedding function followed by the first layer of rotational gates, a controlled rotational gate (CNOT gate) between each qubit, and finally the second layer of rotational gates.

Moreover, it demonstrates a SWAP test where the qubit 2 is swapped with qubit 1. Subsequently, the result is measured to calculate the fidelity of the 2 quantum states that pass through a classical cost function for the optimization. Finally, qubit 0 represents the control qubit. For further clarification, the quantum circuit model and matrix representation of qubit states, and unitary gates are provided in the Appendix.

A popular data encoding method in quantum computing is through angle encoding, also known as qubit encoding. The following transformation is used to produce a quantum state in this process, according to LaRose and Coyle (2020):

$$S_x|0\rangle = \otimes_{i=1}^{n} \cos(x_i)|0\rangle + sin(x_i)|1\rangle \tag{5}$$

The transformation is constructed using a single rotation with a normalized angle corresponding to x_i (normalized to be in $[-\pi, \pi]$) for each qubit. This method allows us to encode n features with n qubits.

The number of qubits needed for this method equals the number of features in your data. It can be effective to deal with data with only a few features. However, building the quantum circuit for angle encoding can be challenging, particularly when dealing with high-dimensional data. With n characteristics, the number of gates needed increases exponentially ($\mathcal{O}(2^n)$). Due to this complexity, it is challenging to develop and run on actual quantum computers.

Another technique for encoding data in quantum computing is amplitude encoding, sometimes referred to as wavefunction encoding. It entails applying the following transformation to produce a quantum state (LaRose and Coyle 2020):

$$S_x|0\rangle = \frac{1}{\|x\|} \sum_{i=1}^{2^n} x_i|i\rangle \tag{6}$$

In this formula, each x_i is a feature of a data point x, and $|i\rangle$ is a basis of n-qubit space. The advantage of this encoding is that we can store 2^n features using only n qubits. However, more often than not, this circuit S_x will have a depth of $\mathcal{O}(2^n)$ and can be challenging to construct.

It is a desirable method because amplitude encoding requires just n qubits to hold 2^n (exponential) data points. This makes data with a large number of features efficient. Amplitude encoding's drawback is that it requires creating the appropriate state vector for each basis state with a different amplitude. Due to the need for intricate quantum procedures, this may be computationally costly. Furthermore, extracting information from the encoded state can be difficult without elaborate measurements.

The execution was carried out using the Qiskit Python library version 0.37. First, we declare the number of qubits used in this circuit. Then, we initialize x as a parameter vector. The quantum kernel circuit quantum feature map $U(x)$ is based on an IQP-like circuit for embedding data sample x. A layer of Hadamard gates will put the qubits in a superposition, then feature-dependent z-basis rotations, and finally, ZZ gates across all the qubits. Lastly, the quantum feature map is passed to a `QuantumKernel` function in Qiskit to create the quantum kernel that will eventually be passed to a one-class SVM algorithm (Figs. 5). We used the 'ibmq_lima' architecture as our IBM quantum device to execute the code.

The quantum circuit for the quantum random forest classifier comprises two primary components: the feature map and the ansatz. Either a feature map or an ansatz is a parameterized quantum circuit. The feature map's role is to convert classical input data into a quantum state, effectively translating the input data into a format understandable by the quantum circuit (Fig. 6).

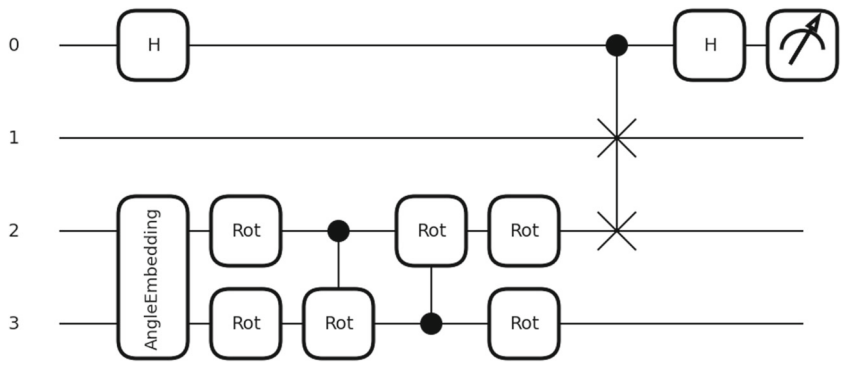


Fig. 4 QAE circuit implementation. It consists of 4 qubits: qubit 0 is the control qubit which goes through a Hadamard gate to put it in a superposition. Qubit 1: is the reference state and the first qubit of the SWAP test. Qubit 2: is the second qubit of the SWAP test and it also the trash state. Qubit 3: is the compressed state

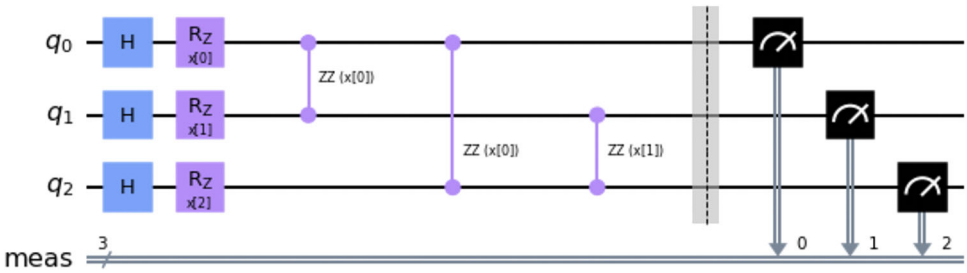


Fig. 5 The OC-SVM quantum kernel circuit. The circuit contains a layer of Hadamard gates, a single Z rotation, and connections between qubits

In our example, we used the `ZZFeatureMap` component from Qiskit, which involves a second-order $Pauli - Z$ evolution circuit followed by a sequence of Hadamard gates. The heart of this structure consists of parameterized entangling blocks, built using rotation gates (RZ) and controlled rotation gates (CZ). These gates encode the input data into the initial quantum state, which serves as the starting point for subsequent processing.

The ansatz, a vital part of the quantum circuit, represents the trainable portion of the quantum classifier. It is responsible for finding a solution to the categorization problem. In our case, we employed the hardware-efficient *RealAmplitudes* ansatz from the Qiskit library, comprising alternating layers of single-qubit Y rotations (RY) and two-qubit entangling gates (CZ). While the overall structure of the ansatz remains fixed, the settings for the RY gates become trainable. The ansatz helps perform a series of quantum operations on the starting state produced by the feature map to construct a measurable state that determines the classifier's output.

The quantum random forest classifier uses an ensemble of the circuits. Each circuit is trained using a distinct subset of the input data that is obtained through bootstrapping. To arrive at the final categorization, a majority voting mechanism is employed to aggregate the circuits' outputs.

The quantum SWAP test is used to determine the proximity of one object to another. To quantify the proximity, one can measure the distance between them in a two-dimensional Euclidean space. In a two-dimensional plane, the distance between two points is simply the length of the path connecting them. In a three-dimensional space, this distance is the space between two real-valued vectors. The Euclidean distance is a widely used method for measuring distances in machine learning. Many machine learning algorithms rely on distance measures between feature vectors at their core (Fig. 7). Such a popular ML algorithm is the k-nearest neighbors algorithm (kNN).

Figure 8 shows an example of two data clusters — one denoted by blue circles and the other by red circles. We have a new sample represented by an orange circle, and our goal is to determine whether it belongs to the red or the blue category. To make this decision, we choose k examples from the dataset that are closest to the orange circle. If we set $k = 3$, the three nearest neighbors consist of two blue circles and one red circle. As a result, the new example is classified as belonging to the blue category because they are in the majority.

Fundamentally, a distance metric evaluates the similarity between two feature vectors. In the quantum version of k-nearest neighbors, different distance metrics can be employed to assess the similarity between feature vectors in a Hilbert space. The main distinction between a Hilbert space and other vector spaces, such as Euclidean space, is that a Hilbert space defines an inner product operation. This inner product can be performed between any two vectors to produce a scalar value. For two vectors x and y in a Hilbert space, we denote the inner product as $\langle x|y\rangle$, where $\langle x|$ is equal to the conjugate transpose of $|x\rangle$. In the context of quantum mechanics and quantum computation, the inner product between two state vectors provides a scalar quantity that indicates how closely the first vector aligns with the second vector. It serves as a measure of similarity between vectors x and y. The absolute square of the inner product can be obtained using a quantum routine known as a SWAP test.

To perform the SWAP test, we require two registers, each containing a vector (a and b, for example), and one ancillary qubit initially set to zero. A Hadamard transformation is used to place the ancilla into a superposition state. Then,

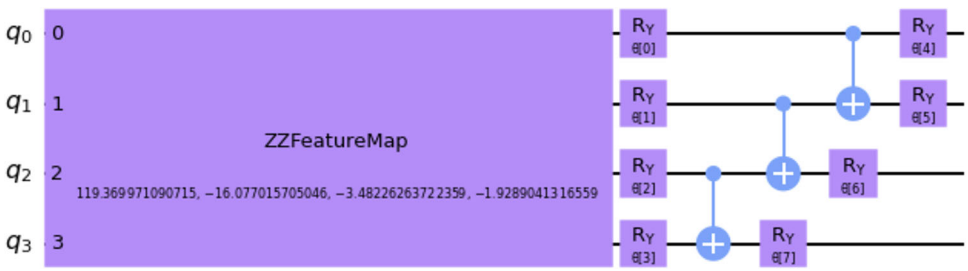


Fig. 6 Quantum random forest circuit. It consists of a feature map that encodes the input data, a layer of y rotation gates, CNOT gates between each qubit, and lastly a layer of z rotation gates

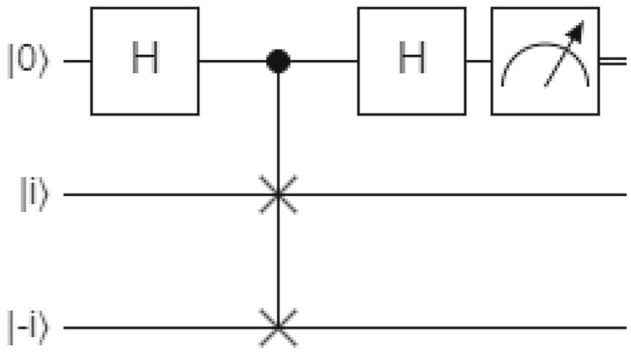


Fig. 7 QkNN SWAP test. It contains 3 qubits: the first qubit is the control qubit which goes through 2 layers of Hadamard gates after which it gets measured, qubits 2 and 3 are the qubits involved in the SWAP test

a controlled SWAP-gate is applied to swap the two registers, but only if the ancillary qubit is in the state one. Finally, a second Hadamard gate on the control qubit.

QRAM works as an oracle in Basheer et al. (2020) to store the training data and initializes one register with a state to categorize. Every training dataset is written in superposition to a register by this oracle. Next, a SWAP-test circuit is designed to carry out the measurement of fidelity. However, our empirical evaluation showed that this method suffers from significantly higher time complexity than our approach when both achieved similar accuracy results.

6 Datasets

In our empirical study, we employed KDD99 (MIT Lincoln Labs 1998), IoT-23 (Garcia et al. 2020), and CIC IoT 23 (Neto et al. 2023) datasets.

6.1 KDD99

Since 1999, KDD'99 has been the most widely used data set for evaluating anomaly detection methods. Stolfo et al. Lee et al. (1999) developed this dataset using data from DARPA's 1998 IDS assessment program. The DARPA'98 dataset contains around 4 gigabytes of raw (binary) compressed *tcpdump* data from 7 weeks of network traffic, which can be processed into approximately 5 million connection records, each with 100 bytes. The test data for the first 2 weeks contains almost 2 million connection records.

The KDD training set includes approximately 4,900,000 single connection vectors. Each vector has 41 features and is labeled as either normal or an attack, with exactly one specific type of attack. The simulated attacks fall into one of the four types (Tavallaee et al. 2009), namely denial of service (DoS) attacks, probe attacks, remote to local (R2L) attacks, and User to root (U2R) attacks. The challenge is made authentic

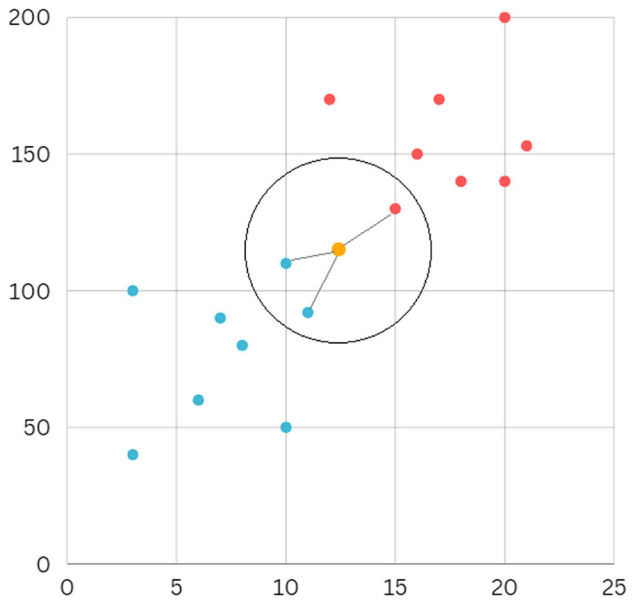


Fig. 8 A kNN graphical example. It illustrates whether the data in the orange circle belongs to the blue or red group of dots. Here, $k = 3$

because the test data includes specific kinds of attacks outside the training data and is not from the same probability distribution. Because most novel attacks are mutant variants of existing ones, the signature of existing attacks can often be used to detect new variants. While the datasets have 24 training attack types, the test data alone contains an additional 14 types.

The KDD'99 features can be organized into three categories:

1. Basic features: This category includes all the attributes derived from a TCP/IP connection. Most of these features introduce implicit detection delays.
2. Traffic features: Traffic characteristics are categorized into two classes based on how they are calculated concerning a window interval:
   - Same host features: These features only look at connections with the same destination host as the current connection during the last 2 s. They compute statistics about protocol behavior and service use.
   - Same service features: These features only look at connections from the last 2 s that use the same service as the one you are on now.

   The two mentioned "traffic" feature categories are referred to as time-based. However, several slow probing exploits use considerably longer time intervals than 2 s, for instance, one per minute, to scan the hosts (or ports). Therefore, these techniques do not result in intrusion patterns within a 2-s time frame. The "same host" and "same service" features are updated to address this issue, but

they are based on a connection window of 100 connections instead of a period of 2 s. These features are called connection-based traffic features.

3. Content features: Unlike most DoS and probing attacks, R2L and U2R attacks do not exhibit any intrusion-regular sequential patterns. DoS and probing attacks, which often involve only one connection, can be distinguished from R2L and U2R attacks that are usually contained in the data parts of packets. To detect these attacks, we need some tools to search for unusual behaviors in the data section, such as the number of unsuccessful login attempts. These are called content features.

6.2 IoT-23

IoT device network traffic is captured in a new dataset called IoT-23. It consists of twenty malware captures performed on IoT devices and three grabs of benign IoT device traffic. The files were captured between 2018 and 2019 and released in January 2020. The network traffic from the IoT was recorded at the Stratosphere Laboratory, AIC group, FEL, CTU University, located in Europe. To assist academics in creating machine learning algorithms, it aims to provide a sizable dataset of actual, tagged IoT malware infections as well as benign IoT traffic. Sponsored by Avast Software, this dataset and study were made possible.

Twenty-three captures (referred to as scenarios) of various IoT network traffic make up the IoT-23 dataset. These scenarios are separated into three network captures of actual IoT device traffic and twenty network captures (*pcap* files) from infected IoT devices which have the name of the malware sample executed on each scenario. They ran a unique malware sample on a Raspberry Pi that utilized many protocols and carried out various operations in each harmful situation. A Somfy smart door lock, an Amazon Echo home intelligent personal assistant, and a Philips HUE smart LED bulb were the three IoT devices whose network data was collected for the benign scenarios. Not to be overlooked is the fact that the three IoT devices. It enables the recording and analysis of actual network behavior. Like any other actual IoT device, both benign and malicious scenarios operate in a regulated network environment with unrestricted internet access.

6.3 CIC IoT 23

This dataset provided a new and large-scale IoT attack dataset in order to support the creation of security analytics tools for actual IoT operations. It was designed to facilitate the development of security analytics software for the IoT environment (Neto et al. 2023). It comprises 33 attacks executed across 105 IoT devices, categorized into seven groups: DDoS, Denial of Service (DoS), Recon, Web-based, Brute Force, Spoofing, and Mirai. These attacks are launched by IoT devices with malicious intent toward other IoT devices.

DDoS attacks include flooding attacks, such as UDP and ICMP floods, and fragmentation-based attacks. DoS attacks disrupt services by overwhelming a single source with traffic. Web-based attacks target web applications using techniques like SQL injection and XSS. Brute force attacks attempt to gain unauthorized access through repeated trials. Spoofing attacks involve impersonating entities or manipulating network traffic. Finally, Mirai attacks employ strategies like GREIP flood and UDPPlain attacks, primarily targeting IoT devices.

In addition, the dataset provides an extensive overview of network attacks, their corresponding frequencies represented by row counts, and their classification into broader attack types. The row counts serve as indicators of the severity of these threats by reflecting the frequency of each unique attack type within the dataset.

Table 3 lists a comprehensive and carefully crafted set of features extracted from network traffic data, offering detailed insights into the characteristics and behaviors of packets within a network.

The "timestamp" attribute assigns a specific recording time to each packet. "flow duration" indicates the duration of a packet's flow. "protocol type" categorizes packets based on their network protocols, including IP, UDP, and TCP. Indicators for application layer protocols such as "HTTP," "HTTPS," and "DNS" are also included, allowing for the identification of specific application-level behaviors in network traffic. "Rate" provides information on data throughput and packet transmission rate.

Furthermore, various flags, including "FIN," "SYN," "RST," "PSH," "ACK," "ECE," and "CWR," provide insights into specific packet-level interactions and potential anomalies. Statistical metrics such as "covariance" and "Variance Ratio" assess the variability in packet lengths, helping us to understand the relationship between incoming and outgoing packet lengths. "Weight" measures the combined count of incoming and outgoing packets, offering a comprehensive view of traffic patterns.

Additional attributes such as "magnitude," "radius," "standard deviation," "packet length," "inter-arrival time," and "packet count" provide depth to the analysis, enabling network specialists to gain valuable insights into the network's performance and security posture.

7 Results and discussion

This section presents the outcomes of our experiments. IBM quantum simulators and real NISQ quantum devices were used in the evaluations.

To assess the effectiveness of our anomaly detection models, we rely on a range of evaluation metrics, including accuracy, F1-score, precision, recall, and the ROC curve. However, in the evaluation of our model, certain metrics carry more weight than others. It is crucial to note that our datasets are unbalanced with a limited number of data samples from the minority class. As a result, accuracy is not a reliable measure of the model's performance because even an inadequate model is capable of performing well in the majority class. In contrast, the F1-score is a metric that accurately reflects the model's performance when dealing with unbalanced datasets. This is particularly relevant in anomaly detection datasets, as the F1-score takes into account the data distribution, providing a more balanced assessment of the model's performance than accuracy.

The performance of our three new frameworks — framework 1 (combination of QAE and OC-SVM), framework 2 (union of QAE and QRF), and framework 3 (union of QAE and QkNN)-is compared in Table 1. As a balanced metric of precision and recall, F1-score is the harmonic mean of accuracy and indicates how well a strategy anticipates anomalies.

With 97% accuracy and 98% F1-score, the combination of QAE and QkNN with the CIC IoT dataset is the best-performing framework. Interestingly, framework 1 performed best with network flow information datasets while Framework 3 performed best with the IoT datasets. The second-best approach is to combine QAE with quantum OC-SVM, which produces a 97% accuracy and 97% F1-score. The least efficient approach is the combination of QAE and quantum OC-SVM with the IoT-23 dataset, despite its 82% accuracy and 79% F1-score. The F1-score, on the other hand, was selected as the primary assessment metric since it takes into account both precision and recall, offering a more balanced measure than accuracy, which can be deceptive on unbalanced datasets — a characteristic of datasets used for anomaly identification.

When evaluating the IoT-23 and CIC IoT 23 datasets, the most effective quantum framework is framework 3, which combines quantum autoencoder (QAE) and quantum k-nearest neighbors (kNN). This is because kNN is a distance-based algorithm that makes decisions based on the similarity between data points. When dealing with network flow datasets exhibiting well-defined clusters or neighborhoods, kNN excels at identifying anomalies by flagging data points that are distant from their nearest neighbors.

Network flow data frequently displays local patterns and clusters, where anomalies may be isolated in specific regions (Ruan et al. 2017; Miao et al. 2018). Specifically, kNN is well-suited for capturing local patterns. Moreover, kNN is highly interpretable; when it identifies an anomaly, one can easily comprehend the reasons by examining its nearest neighbors. This feature enhances the ability to validate results and take appropriate actions.

Quantum k-nearest neighbors (QkNN) analysis:

- Quantum speedup: Quantum algorithms in theory, such as QkNN, hold the potential to outperform classical algorithms, especially when handling large datasets. However, in practice and our experiments, achieving this quantum speedup is currently hindered by the limitations and restrictions of existing quantum hardware.
- Distance calculation: In anomaly detection, it is crucial to calculate distances between data points efficiently and accurately. Quantum algorithms can perform quantum amplitude amplification, which can be highly efficient for distance-based calculations, potentially outperforming classical ML algorithms (Lloyd et al. 2020).
- Quantum superposition: Quantum algorithms can represent data points in a superposition of states, allowing for parallel processing of data. This quantum representation is a clear advantage over classic computing when anomalies are rare and spread across the dataset.
- Dimensionality reduction: Quantum algorithms help handle high-dimensional data efficiently when data is projected into a quantum state using various quantum techniques, including quantum feature maps, quantum

Table 1 Comparison among the proposed quantum frameworks

Dataset	Method	Accuracy	Precision	Recall	F1-score
KDD99	Framework 1	97.48%	95.06%	**99.43%**	97.19%
	Framework 2	92.39%	89.63%	97.16%	93.41%
	Framework 3	91.73%	90.89%	96.34%	93.15%
IOT-23	Framework 1	82.53%	70.31%	98.01%	79.69%
	Framework 2	87.70%	78.49%	94.17%	86.25%
	Framework 3	96.80%	95.09%	98.73%	96.19%
CIC IoT 23	Framework 1	97.30%	96.83%	98.14%	97.67%
	Framework 2	96.10%	94.63%	97.72%	97.61%
	Framework 3	**97.79%**	**98.37%**	98.81%	**98.26%**

The bolded values are to indicate the best result in their respective columns/categories

data encoding, and the like. This knowledge is beneficial to deal with network flow data with many features (Liang et al. 2020; Cong and Duan 2016).

Our method's combination of parameterized quantum circuits with deep neural networks shows promise for tackling the anomaly detection issue. We explore the finer points of how these two elements work together to form a potential hybrid model throughout our discussion. We examine each of their unique contributions and draw attention to the synergistic benefits that result from their combination.

In Table 4 of the Appendix, we conducted a comparison between identical machine learning (ML) classifiers employing the same data preprocessing steps. The only difference lies in the classical implementation rather than our quantum frameworks. We aim to examine how the quantum versions' performance compares to their classical counterparts in a controlled environment. The results reveal that our quantum frameworks consistently outperformed the classical ML classifiers across all the datasets. Notably, the classical ML classifiers exhibited a consistent performance trajectory across different datasets. For instance, in the KDD99 dataset, the one-class SVM demonstrated superior performance compared to RF and kNN. In the case of IoT datasets, kNN exhibited the best performance among the classifiers. These findings remain consistent when analyzing only the quantum frameworks' performance. Importantly, these results suggest a notable correlation between quantum coherence and anomaly detection, providing valuable insights into the relationship between quantum computing and ML performance.

The time required to train the QAE with varying numbers of qubits is presented in Table 2. We find that the more qubits are used, the lower its fidelity scores are; however, the time spent on training increased exponentially with the number of qubits employed. Upon the completion of the SWAP test, we employed 4 total qubits and 2 latent qubits based on our quantum autoencoder technique to convert the data into a linear form. As a result, we had to perform substantial pre-processing on the data and modify the quantum circuits to fit the particular job throughout our testing which make our solution limited and not scalable. Our results show that the QML paradigm could yield positive outcomes with huge potential for improvements.

Table 2 Time comparison with different numbers of qubits used

Total qubits	Latent qubits	Time taken	Fidelity
4	2	1.64 h	0.9963
4	3	1.49 h	0.9812
6	4	7.85 h	0.8610

Finally, the $L2$ norm was applied in our QAE implementation. In classic computing, the $L2$ norm between two vectors ($\mathbf{v}$ and $\mathbf{w}$) is calculated as follows:

$$\|v - w\|_2^2 = \sum_i (v_i - w_i)^2 \tag{7}$$

The $L2$ norm in quantum mechanics is calculated slightly differently. The computation of the squared $L2$ norms between a state vector $|\psi\rangle$ and a density matrix ρ in the context of quantum mechanics are described as follows:

1. State vector $|\psi\rangle$:
   - A quantum state represented by a state vector $|\psi\rangle$ is a pure state.
   - The $L2$ norm of a state vector $|\psi\rangle$ is denoted by: $\||\psi\rangle\|_2^2 = \langle\psi|\psi\rangle$.
2. Density matrix ρ:
   - A mixed state is represented by a density matrix ρ, which may be a statistical mixture of pure states.
   - The $L2$ norm of a density matrix ρ is given by the trace of the square of the matrix: $\|\rho\|_2^2 = Tr(\rho^2)$.

The squared $L2$ norm between the state vector $|\psi\rangle$ and the density matrix ρ is computed as the trace of the square of the difference:

$$F(|\psi\rangle, \rho) = Tr((|\psi\rangle\langle\psi| - \rho)^2) \tag{8}$$

The difference between the mixed state, represented by the density matrix ρ, and the pure state, $|\psi\rangle$, is quantified by this formula. The basis-independence of the calculation is guaranteed by the trace operation.

8 Conclusion and future work

This paper draws attention to the growing number of network threats and highlights how important anomaly detection is to improving cyber security. Our study intends to close this research gap and provide innovative frameworks that make use of quantum computers, acknowledging the limitations in the literature that currently exists about the integration of quantum technologies and substantiating the effectiveness of hybrid models that unite deep neural networks and parameterized quantum circuits for anomaly detection.

The three proposed approaches that integrated quantum autoencoders with quantum kNN, quantum random forest, and quantum SVM show promises for precisely detecting abnormalities in network traffic. The QAE approach integrated with a quantum kNN model has the best performance

among the other implemented frameworks and the state-of-the-art quantum and classical anomaly detection techniques, with a higher accuracy and F1 scores using benchmark attack datasets.

Our results signify a novelty in the synergy between quantum autoencoders and anomaly detection approaches, and their significance goes beyond the immediate implications. Our study fills a research absence in the literature and paves the way for enhanced anomaly detection methods that take advantage of quantum computing capabilities. This advances the field of cyber security and advances the larger investigation of quantum technology for the resolution of other complex problems.

In the future, these frameworks should be expanded and improved upon. Further strengthening anomaly detection systems' resilience might involve investigating various quantum algorithms, improving quantum autoencoder designs, and expanding assessments to more complex datasets. The technical implementation of these frameworks will also depend on examining their scalability for large-scale network settings and considering hardware limits. Our research findings pave a potential pathway for integrating quantum technologies with anomaly detection for real-world applications. While acknowledging our role in proposing novel technical methods, we recognize the evolving nature of research. We anticipate that future work may uncover superior frameworks, but our contribution stands as a robust foundation for advancing anomaly detection.

Appendix

Quantum circuit model and matrix representation of qubit states, and unitary gates.

Qubit states:

$|0\rangle \begin{bmatrix}1\\0\end{bmatrix}$ $|1\rangle \begin{bmatrix}0\\1\end{bmatrix}$

X-gate:

Hadamard gate:

$CNOT$-gate:

$SWAP$-gate:

Table 3 Summary of utilized datasets in terms of features and data points used

Dataset	No. of features	No. of data points
KDD99	41	4,900,000
IoT-23	21	1,444,674
CIC IoT 23	47	243,649

Table 4 Comparison among classical ML classifiers with the same data preprocessing steps using the 10-fold cross-validation

Dataset	Method	Accuracy	F1-score
KDD99	SVM	90.21%	78.53%
	RF	87.57%	91.63%
	kNN	85.28%	84.30%
IOT-23	SVM	71.78%	76.34%
	RF	83.92%	82.60%
	kNN	92.57%	93.50%
CIC IoT 23	SVM	93.74%	90.83%
	RF	91.90%	94.19%
	kNN	**94.62%**	**95.49%**

The reported results are averaged over 5 iterations
The bolded values are to indicate the best result in their respective columns/categories

Funding Open Access funding enabled and organized by CAUL and its Member Institutions. The authors wish to acknowledge that this research received no external funding or financial support. All expenses associated with this paper were covered by the authors.

Availability of data and materials The data that support the findings of this paper are openly available and include the following:
•KDD99 is available at http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html, reference (MIT Lincoln Labs 1998).
•IoT-23 is available at http://www.stratosphereips.org/datasets-iot23, reference (Garcia et al. 2020).
•CIC IoT-23 is available at http://www.unb.ca/cic/datasets/iotdataset-2023.html, reference (Neto et al. 2023).
The code that supports the findings of this paper is available from the corresponding author, M.H., upon a reasonable request.

References

Alchieri L, Badalotti D, Bonardi P, Bianco S (2021) An introduction to quantum machine learning: from quantum logic to quantum deep learning. Quantum Mach Intell 3:1–30

An J, Cho S (2015) Variational autoencoder based anomaly detection using reconstruction probability. Special Lecture on IE 2(1):1–18

Aspuru-Guzik A, Dutoi AD, Love PJ, Head-Gordon M (2005) Simulated quantum computation of molecular energies. Science 309(5741):1704–1707

Baldi P, Hornik K (1989) Neural networks and principal component analysis: learning from examples without local minima. Neural Netw 2(1):53–58

Bartůšková L, Černoch A, Filip R, Fiurášek J, Soubusta J, Dušek M (2006) Optical implementation of the encoding of two qubits to a single qutrit. Phys Rev A 74(2):022325

Basheer A, Afham A, Goyal SK (2020) Quantum k-nearest neighbors algorithm. arXiv:2003.09187

Bravo-Prieto C (2021) Quantum autoencoders with enhanced data encoding. Mach Learn: Sci Technol 2(3):035028

Cong I, Duan L (2016) Quantum discriminant analysis for dimensionality reduction and classification. New J Phys 18(7):073011

Dang Y, Jiang N, Hu H, Ji Z, Zhang W (2018) Image classification based on quantum K-nearest-neighbor algorithm. Quantum Inf Process 17:1–18

Ding Y, Lamata L, Sanz M, Chen X, Solano E (2019) Experimental implementation of a quantum autoencoder via quantum adders. Adv Quantum Technol 2(7–8):1800065

Erfani SM, Rajasegarar S, Karunasekera S, Leckie C (2016) High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning. Pattern Recognition 58:121–134

Garcia S, Parmisano A, Erquiaga MJ (2020) IoT-23: a labeled dataset with malicious and benign IoT network traffic (Version 1.0.0) [Data set]. Zenodo https://doi.org/10.5281/zenodo.4743746

Gong C, Guan W, Gani A, Qi H (2022) Network attack detection scheme based on variational quantum neural network. J Supercomput 78(15):16876–16897

Gouveia A, Correia M (2020) Towards quantum-enhanced machine learning for network intrusion detection. In: 2020 IEEE 19th international symposium on Network Computing and Applications (NCA), pp 1–8. IEEE

Hashemzehi R, Mahdavi SJS, Kheirabadi M, Kamel SR (2020) Detection of brain tumors from MRI images base on deep learning using hybrid model CNN and NADE. Biocybernetics and Biomedical Engineering 40(3):1225–1232

Huang CJ, Ma H, Yin Q, Tang JF, Dong D, Chen C, Xiang GY, Li CF, Guo GC (2020) Realization of a quantum autoencoder for lossless compression of quantum data. Phys Rev A 102(3):032412

Kottmann K, Metz F, Fraxanet J, Baldelli N (2021) Variational quantum anomaly detection: unsupervised mapping of phase diagrams on a physical quantum computer. Phys Rev Res 3(4):043184

Kulkarni V, Kulkarni M, Pant A (2021) Quantum computing methods for supervised learning. Quantum Mach Intell 3(2):23

Kyriienko O, Magnusson EB (2022) Unsupervised quantum machine learning for fraud detection

Lamata L, Alvarez-Rodriguez U, Martín-Guerrero J, Sanz M, Solano E (2018) Quantum autoencoders via quantum adders with genetic algorithms. Quantum Sci Technol 4(1):014007

LaRose R, Coyle B (2020) Robust data encodings for quantum classifiers. Phys Rev A 102(3):032420

Lee W, Stolfo SJ, Mok KW (1999) Mining in a data-flow environment: experience in network intrusion detection. In: Proceedings of the 5th ACM SIGKDD international conference on knowledge discovery and data mining, pp 114–124

Liang JM, Shen SQ, Li M, Li L (2020) Variational quantum algorithms for dimensionality reduction and classification. Phys Rev A 101(3):032323

Li W, Deng DL (2022) Recent advances for quantum classifiers. Sci China Phys Mech Astron 65(2):220301

Lloyd S, Schuld M, Ijaz A, Izaac J, Killoran N (2020) Quantum embeddings for machine learning. arXiv:2001.03622

Lu LH, Li YQ (2019) Quantum approach to fast protein-folding time. Chin Phys Lett 36(8):080305

Lv P, Yu Y, Fan Y, Tang X, Tong X (2020) Layer-constrained variational autoencoding kernel density estimation model for anomaly detection. Knowl Based Syst 196:105753

Mangini S, Marruzzo A, Piantanida M, Gerace D, Bajoni D, Macchiavello C (2022) Quantum neural network autoencoder and classifier applied to an industrial case study. Quantum Mach Intell 4(2):13

McClelland JL, Rumelhart DE, Research Group PDP et al (1987) Parallel distributed processing, vol 2: explorations in the microstructure of cognition: psychological and biological models vol. 2, MIT press, ???

Mete B, Gutierrez IL, Mendl C (2021) Hamiltonian simulation using quantum autoencoders

Miao Y, Ruan Z, Pan L, Zhang J, Xiang Y (2018) Comprehensive analysis of network traffic data. Concurr Comput Pract Exp 30(5):e4181

MIT Lincoln Labs (1998) KDD Cup 1999 Data — kdd.ics.uci.edu. http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html Accessed 30 Oct 2023

Neto ECP, Dadkhah S, Ferreira R, Zohourian A, Lu R, Ghorbani AA (2023) CICIoT2023: a real-time dataset and benchmark for large-scale attacks in IoT environment

Neto ECP, Dadkhah S, Ferreira R, Zohourian A, Lu R, Ghorbani AA (2023) CICIoT2023: a real-time dataset and benchmark for large-scale attacks in IoT environment. Sensors 23(13):5941. https://doi.org/10.3390/s23135941

Nielsen MA, Chuang IL (2010) Quantum computation and quantum information. Cambridge University Press

Pang G, Shen C, Cao L, Hengel AVD (2021) Deep learning for anomaly detection: a review. ACM Comput Surv 54(2):1–38

Pepper A, Tischler N, Pryde GJ (2019) Experimental realization of a quantum autoencoder: the compression of qutrits via machine learning. Phys Rev Lett 122(6):060501

Plaut E (2018) From principal subspaces to principal components with linear autoencoders. arXiv:1804.10253

Pol AA, Berger V, Germain C, Cerminara G, Pierini M (2019) Anomaly detection with conditional variational autoencoders. In: Proceedings of the 2019 18th IEEE International Conference on Machine Learning and Applications (ICMLA), pp 1651–1657. IEEE

Pu Y, Gan Z, Henao R, Yuan X, Li C, Stevens A, Carin L (2016) Variational autoencoder for deep learning of images, labels and captions

Rajasegarar S, Leckie C, Bezdek JC, Palaniswami M (2010) Centered hyperspherical and hyperellipsoidal one-class support vector machines for anomaly detection in sensor networks. IEEE Trans Inf Forensics Secur 5(3):518–533

Ranzato M, Huang FJ, Boureau YL, LeCun Y (2007) Unsupervised learning of invariant feature hierarchies with applications to object recognition. In: IEEE Conf. on Computer Vision and Pattern Recognition, pp 1–8. IEEE

Rizvi B, Belatreche A, Bouridane A, Watson I (2020) Detection of stock price manipulation using kernel based principal component analysis and multivariate density estimation. IEEE Access 8:135989–136003

Romero J, Olson JP, Aspuru-Guzik A (2017) Quantum autoencoders for efficient compression of quantum data. Quantum Sci Technol 2(4):045001

Ruan Z, Miao Y, Pan L, Patterson N, Zhang J (2017) Visualization of big data security: a case study on the KDD99 cup data set. Digital Communications and Networks 3(4):250–259

Ruff L, Vandermeulen R, Goernitz N, Deecke L, Siddiqui SA, Binder A, Müller E, Kloft M (2018) Deep one-class classification. In: Proceedings of the 2018 international conference on machine learning, pp 4393–4402. PMLR

Shilton A, Rajasegarar S, Palaniswami M (2020) Multiclass anomaly detector: the CS++ support vector machine. Journal of Machine Learning Research 21(1):8753–8791

Shiravi A, Shiravi H, Tavallaee M, Ghorbani AA (2012) Toward developing a systematic approach to generate benchmark datasets for intrusion detection. Comput Secur 31(3):357–374

Srikumar M, Hill CD, Hollenberg LCL (2021) Clustering and enhanced classification using a hybrid quantum autoencoder. Quantum Sci Technol 7(1):015020

Steinbrecher GR, Olson JP, Englund D, Carolan J (2019) Quantum optical neural networks. Npj Quantum Inf 5(1):60

Tavallaee M, Bagheri E, Lu W, Ghorbani AA (2009) A detailed analysis of the KDD CUP 99 data set. In: Proceedings of the 2009 IEEE symposium on computational intelligence for security and defense applications, pp 1–6. IEEE

Tax DM, Duin RP (2004) Support vector data description. Mach Learn 54:45–66

Trugenberger CA (2002) Quantum pattern recognition. Quantum Inf Process 1:471–493

Wang MM, Jiang YD (2022) Data reconstruction based on quantum neural networks. arXiv:2209.05711

Zhang XM, Kong W, Farooq MU, Yung MH, Guo G, Wang X (2021) Generic detection-based error mitigation using quantum autoencoders. Phys Rev A 103(4):L040403

Zhang C, Liu J, Chen W, Shi J, Yao M, Yan X, Xu N, Chen D (2021) Unsupervised anomaly detection based on deep autoencoding and clustering. Secur Commun Netw 2021:1–8

Zhang XM, Li T, Yuan X (2022) Quantum state preparation with optimal circuit depth: implementations and applications. Phys Rev Lett 129(23):230504