Laxminarayana, Nikhil; Mishra, Nimish; Tiwari, Prayag; Garg, Sahil; Behera, Bikash K.; Farouk, Ahmed

Quantum-Assisted Activation for Supervised Learning in Healthcare-based Intrusion Detection Systems

Published in:
IEEE Transactions on Artificial Intelligence

DOI:
10.1109/TAI.2022.3187676

Published: 01/03/2024

Document Version
Peer-reviewed accepted author manuscript, also known as Final accepted manuscript or Post-print

Please cite the original version:
Laxminarayana, N., Mishra, N., Tiwari, P., Garg, S., Behera, B. K., & Farouk, A. (2024). Quantum-Assisted Activation for Supervised Learning in Healthcare-based Intrusion Detection Systems. *IEEE Transactions on Artificial Intelligence*, *5*(3), 977-984. Article 9813378. https://doi.org/10.1109/TAI.2022.3187676

Quantum-Assisted Activation for Supervised Learning in Healthcare-based Intrusion Detection Systems

Nikhil Laxminarayana, Nimish Mishra, Prayag Tiwari, Sahil Garg, Bikash K. Behera, Ahmed Farouk

***Abstract*—Intrusion detection systems (IDS) are amongst the most important automated defense mechanisms in modern industry. It is guarding against many attack vectors, especially in healthcare, where sensitive information (patient's medical history, prescriptions, electronic health records, medical bills/debts, and many other sensitive data points) is open to compromise from adversaries. In the big data era, classical machine learning has been applied to train IDS. However, classical IDS tend to be complex: either using several hidden layers susceptible to overfitting on training data or using overly complex architectures such as convolutional neural networks (CNNs), long-short term memory systems (LSTMs), and recurrent neural networks (RNNs). This paper explored the combination of principles of quantum mechanics and neural networks to train IDS. A hybrid classical-quantum neural architecture is proposed with a quantum-assisted activation function that successfully captures patterns in the dataset while having less architectural memory footprint than classical solutions. The experimental results are demonstrated on the popular KDD99 dataset while comparing our solution to other classical models.**

***Impact Statement*—IDS are dynamic defenses against network breach attacks. Lately, machine learning has been leveraged to perform automated intrusion detection. However, classical machine learning needs a large amount of data and overly complex architectures to "learn" patterns from data. In this work, machine learning concepts are applied to derive a novel quantum activation function that greatly simplifies neural architectural complexity while achieving the same level of accuracy and performance. In addition, the proposed architecture is much simpler than state-of-the-art-based classical systems, simpler to train, and easy to handle.**

***Index Terms*—Quantum Machine Learning, Intrusion Detection, Activation Function, Supervised Learning**

I. Introduction

Every facet of how medical institutions operate is being radically altered and driven by the digital revolution. The sheer volume of medical data created, manipulated, and stored by medical institutions is increasing, creating new horizons and necessitating more data governance. Furthermore, computing environments are more complex than ever: mainly covering the private cloud storage in medical networks storing sensitive patient data. Such addition to the attack surface makes it more difficult to track attackers and secure medical networks. The value of data to the medical industry in today's data-driven world is so immense that a data breach can result in a significant loss in revenue not just to the medical sector but to several interlinked industries [1] (including the pharmaceutical industry). These losses are not just capital losses, but also intangibles, presenting us with the daunting task of keeping medical information secure [1].

With such an inflow of data from all domains, the volume of data collected by various medical institutions has grown several manifolds. However, the traditional cybersecurity measures are now proving insufficient to protect this inflow. Monteith *et al.* [2] document the rise of cybercrime during the global pandemic (and since medical institutions play a central role in such pandemics, cyberattackers aim to target such institutions). They present evidence that the types of cyberattacks are now more fundamentally non-distinguishable from actual access requests. Furthermore, existing cybersecurity infrastructure lacks the foresight to warn the user from some potential attack vectors, a vulnerability that adversaries are increasingly capable of exploiting. Therefore, due to a large number of avenues available to adversaries, the potential to exploit the network and its data is much more advanced than ever before [3]. All these statements make a compelling case to consider the role of deep learning in cybersecurity in a prudent way. Deep learning-assisted cybersecurity systems are much more suited to examining patterns and learning from them to help prevent repeated intrusions and react to changing behaviors [4]. Furthermore, it can assist cybersecurity teams in becoming more proactive in terms of avoiding risks and responding to active attacks in real-time.

With the advent of quantum machine learning, the following question should be asked: can quantum machine learning pave new avenues in machine learning-assisted cybersecurity systems specially suited to medical networks? Quantum machine learning is simply the application of quantum circuits and algorithms within machine learning programs [5]. By integrating quantum computing with deep learning structures, the potential of various quantum phenomena can be leveraged to enhance the learning capabilities of a classical neural network

N. Laxminarayana is with the Department of Electronics and Communications Engineering, Indian Institute of Information Technology, West Bengal, India; Email: laxminarayana.nikhil@gmail.com

N. Mishra is with the Department of Computer Science and Engineering, Indian Institute of Information Technology, West Bengal, India; Email: nimish_bt18@iiitkalyani.ac.in

P. Tiwari is with Department of Computer Science, Aalto University, Espoo, Finland; Email: prayag.tiwari@aalto.fi

S. Garg is with the École de Technologie Supérieure, Montréal, QC H3C 1K3, Canada; Email: sahil.garg@ieee.org

B. K. Behera is with the Bikash's Quantum (OPC) Pvt. Ltd., Mohanpur, WB, 741246 India; Email: bikash@bikashsquantum.com

A. Farouk is with the Department of of Computer Science, Faculty of Computers and Artificial Intelligence, South Valley University, Hurghada, Egypt; Email: ahmed.farouk@sci.svu.edu.eg

Corresponding Authors: Sahil Garg, Prayag Tiwari, and Ahmed Farouk

through embedding a layer of quantum perceptrons within a classical neural network [6]. So, it will eliminate the various shortcomings of an intrusion detection system designed by some of the other current machine learning algorithms.

To the best of authors' knowledge, the following points are explored:

- how quantum machine learning can help in cybersecurity?
- how quantum machine learning-based systems contrast against classical machine learning-based systems?
- how quantum computers can be used to power next-generation cybersecurity systems for the healthcare industry?

This paper proposed architectures consisting of layer(s) of parameterized quantum circuits working in harmony with the classical neural network. The hybrid classical and quantum machine learning-driven system has been developed, which can learn upon structured data to predict intrusion attempts in connected networks. The popular KDD99 dataset used for intrusion detection system studies has been employed to set our system against current classical machine learning systems. We explore how one can develop complex non-linear activation functions to learn more complex relationships in structured data. Unfortunately, counterparts' architectures use hidden layers with massive perceptron and have a high memory footprint with complex architectures like CNN, LSTM, RNN, etc. On the other hand, our architecture achieved a higher accuracy, capturing the input-output relationship in just 8 node quantum. The main objective of this work is to guide more research into the hybrid of classical and quantum neural networks for various learning tasks.

A. *Classical neural networks*

A classical neural network aims to mimic, to a large extent, certain cognitive functions as performed by the human mind. These cognitive functions include all the functions associated with the five senses of humans and have had, so far, applications in pattern recognition and classification [7]. Haykin et. al. [8] defined neural network that can help to come up with a good idea of how a neural network tries to mimic cognitive functions. A neural network, in essence, is a parallel combination of simple processing units that can acquire knowledge from a learning environment and store it in its connections or synaptic weights. The theoretical basis for a classical neural network stems from a fundamental analysis of models capable of cognition [7]:

- Existence of parallel simple processing logical units.
- Coherence between processing units for the system to solve problems without needing a centralized algorithm.
- The logical units and their relationships are characteristic to the system as they define the knowledge the network holds.
- The system is designed to be domain-independent but acquires knowledge and trains itself into a domain-specific system.
- The system is supposed to learn from the training phase and, by design, must be able to apply the knowledge to solving similar problems.

These theoretical ideas form a basis of what a classical neural network must be able to do in principle. The most important question is: how are these ideas translated into a classical neural network capable of performing a cognitive function, having sharp resemblance with the minds of living beings? It is indeed an elegant question, for one must keep in mind the definiteness associated with any computing system when a non-definite task is set to such a system. The way humans think does not resemble how a simple computing architecture works. Subsequently, how a system can be derived that is definite in itself but manages to learn from its environment and then applies learned ideas to perform the task it was designed to perform. Any general neural network has some core ideas associated with its design. These core ideas are defined in four levels [7].

- **State vector and Weight matrix**: A classical neural network contains a constant number of neurons organized in layers that interact to propagate information for generating output by converging the input from the neurons in the previous layer. Each neuron has an activation value associated with it, and each neuron-to-neuron connection has a weight attached to it. Moreover, for the network to work as required, a certain bias may be applied to each connection to reflect which connection has the most influence on the output. The weights of all connections may be organized into an $N \times N$ matrix.
- **Activation Spreading**: In each state change step, a neuron gets its input from the neurons in the previous layer (or the learning environment). Then an activation function acts upon the input to give an updated activation value to the neuron, which then serves as the output sent to the neurons in the next layer. The state vector changes whenever this process occurs and this process is repeated until the state vector yields a stable value. This is the essence of training a classical neural network.
- **Setting Parameters**: The weights of the connections can be understood like a knob that affects the generated output. The learning phase of a neural network is characterized by finding a suitable setting for all such knobs to reach the closest to finding the ideal output. Therefore, training cases need to be repeatedly presented to the system to make the weight matrix values converge to a single value.
- **Derived Function**: As a result of the repetition of training, until the values for input vectors converge to the output vector, the network acts as a derived function that maps a certain input vector to a particular output vector.

Due to their ability to adapt to input and provide a definitive model for non-linear processes, classical NNs find varied applications in diverse fields: medicine, quantum chemistry, 3D reconstruction, face recognition, and data mining. In addition, classical NNs are used in cybersecurity to classify malware for threat identification, penetration testing, and network intrusion detection. Unfortunately, these NNs lack processing, memory, and accuracy, affecting performance and security. Therefore, applying the quantum computing laws and resources will improve the development of IDS.

B. Quantum-assisted neural networks

Despite the ongoing advancements in conventional computing, accessible modern hardware imposes restrictions on the practicality of certain machine learning models. Moreover, deep learning is expensive in terms of time and resources. The pace at which classical computers are advancing is bound to hit huge roadblocks in years. Hence, there is a surge in demand for various alternatives to classical neural networks. Among all those alternatives, quantum computers provide a new horizon of possibilities yet to be fully explored. What quantum computing presents is a set of new principles inherent to computing in the near future. The principles of superposition, quantum entanglement, and interference have been utilized in our development to achieve a portable and robust IDS.

Quantum computations are, in essence, challenging to be carried out on classical computers as the resource demand for the same increases exponentially with the number of qubits. The new and expanding regime of Noisy Intermediate-Scale Quantum (NISQ) devices provides a precise model of the errors that occur and gives us an excellent framework to build quantum machine learning and deep learning models. Qiskit Aer is used along with Qiskit Terra, forms a simulation backend offered by Qiskit, and is a highly configurable noisy model where computations can be done.

With the advent of NISQ devices, Parameterized quantum circuits (PQCs) provide a practical framework for implementing algorithms and leveraging those quantum phenomena [9]. PQCs are composed of the CNOT gates and qubit rotation adjustable gates. Benedetti *et al.* [10] mention the various advancements in the performance of PQCs. They have addressed one of the most critical challenges: the effects of noise. They enumerate the various proposed learning approaches for a quantum circuit. A layer of perceptrons has a rather characteristic working flow: the input data received from the previous layer is encoded into the corresponding qubits, after which an activation function, a parameterized function of rotation and CNOT gates, defines that qubit till it stores the data. Afterward, the expected value of the Hamiltonian is used to obtain the transformed qubit state vector using Pauli gates. The outputs serve as input to the next layer of perceptrons after decoding the data [11].

C. Contribution

Contributions of the work as follows:

- A quantum-assisted non-linear activation function is proposed that cannot be efficiently simulated on classical computers.
- A hybrid neural network is trained on the KDD99 dataset, thereby demonstrating the ability of the quantum layers to capture patterns in structured data with over a hundred features of varying correlations.
- The architecture is able to capture patterns in structured data despite having a lesser number of parameters than its classical counterparts.

D. Organisation

The rest of the paper is organized as follows. In Sec. II, we start with an analysis on the rival architectures presented in existing literature. We then explain and derive the quantum-assisted activation function using parameterized operations in Sec. III-B, which is followed by explaining the types of major attacks undertaken by the exploiter and explaining why KDD99 dataset is the most suitable to train a model to stand up to those attacks and elaborate on the findings in Sec. IV-A. Finally, the results are briefly discussed in Sec. V and the concluding statements are presented in Sec. VI along with the scope of future works.

II. Related Works

Correia *et al.* [12] proposed an approach to implementing unsupervised quantum machine learning for IDS. However, a supervised learning algorithm would prove more reliable for IDS where the system results have high stakes. The outcome was expected to be a better and more flexible IDS that carries training-based knowledge of a wider spectrum of requests. Payares *et al.* [13] focused on the implementation of quantum models to detect denial of service (DOS) attacks, while the focus here is on a much wider spectrum of attack vectors. Jeyakarthic *et al.* [14] use simple quantum multi-layer perceptrons, which are known not to learn complex functions as effectively as neural networks do. We focus on QNNs, which capture more interesting patterns.

Yong *et al.* [15] use a technique that is not well suited to learning large-scale data. The proposed model is trained on a much larger dataset. Dong *et al.* [16] proposed a reliable system capable of learning complex functions by employing the quantum beetle swarm algorithm and quantum neural networks. Bishop [17] has presented an exciting solution depending on photo-detectors that is not feasible in near-term quantum computing applications. Yin [18] worked on quantum optimization, while an exploration of a tangent topic is undertaken here, namely quantum hybrid neural networks. Wang *et al.* [19] used quantum-behaved particle swarm optimization, but we explore a tangent method, QNNs. Our proposed system is capable of learning much more complex functions as compared to [19].

III. Proposed Methodology

A. IDS design for healthcare

The proposed intrusion detection architecture fits in with the problem of securing medical networks. As shown in Fig. 1, a typical medical network comprises several interconnected subsidiary systems. These involve:

- **client endpoints** with clients/patients, which contain their private medical data
- **doctors' systems**, which contain their own private data along with the allotted patients' data
- **EHR database**, which contains electronic health records of all patients registered in the network
- **hospital's SCADA systems**, which include systems like fire-control and a power supply that keeps the building power.

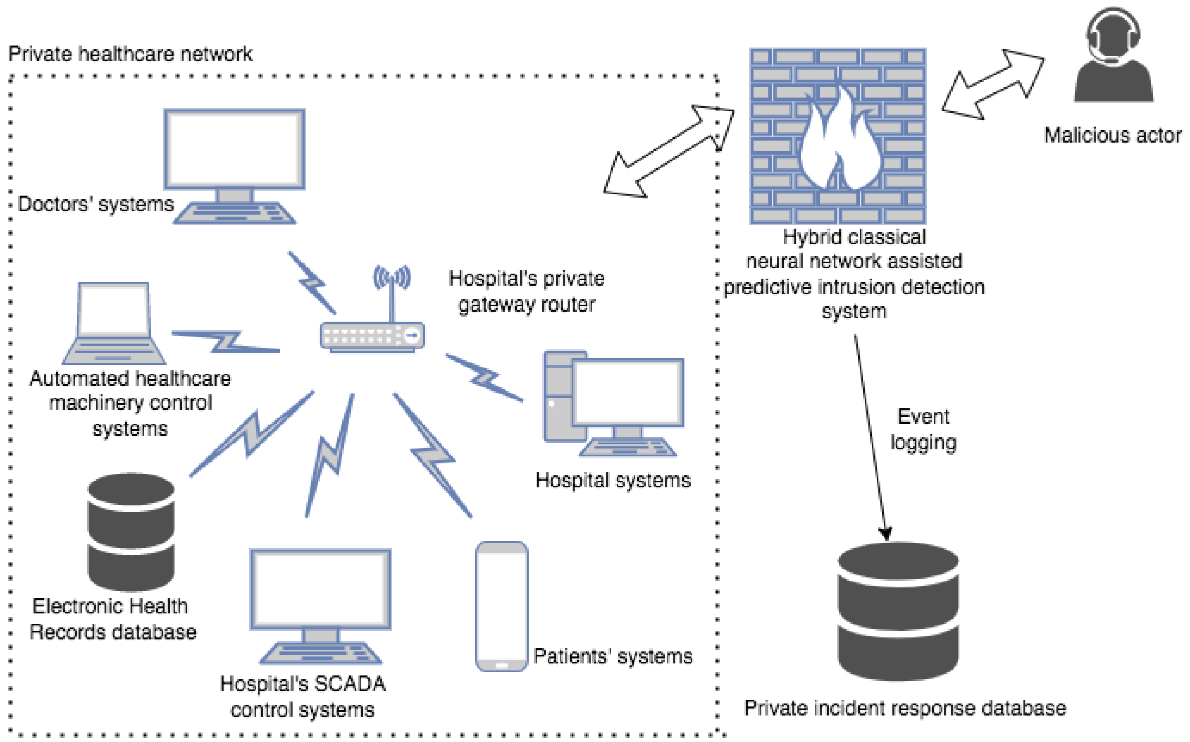

Private healthcare network
Doctors' systems
Automated healthcare machinery control systems
Hospital's private gateway router
Hospital systems
Electronic Health Records database
Hospital's SCADA control systems
Patients' systems
Hybrid classical neural network assisted predictive intrusion detection system
Malicious actor
Event logging
Private incident response database

Fig. 1: A schematic diagram of the hybrid quantum-classical neural network-assisted intrusion detection system fitting in with the rest of the system in a private healthcare system. The quantum IDS protects all incoming traffic to the network and logs its prediction to an event logger database to let the incident response team focus on them in case of any malicious activity.

- **hospital's administration systems**, which include data as well as systems related to the smooth functioning of the hospital (for example, financial control systems)

A breach of any of these systems can ultimately compromise the stored data and security of personnel on the hospital premises. For example, an attacker who compromises a hospital's SCADA systems can disrupt the power supply to sensitive infrastructure like intensive care units. As depicted in Fig. 1, the proposed intrusion detection architecture validates all incoming traffic. The outgoing traffic does not need to be validated as they are assumed to be exchanged over a trusted connection. The proposed IDS captures potentially harmful traffic, blocks its entry to the network, logs the incident in a private incident response database, and blacklists the IP address of the attacker to prevent further attacks. In parallel, the medical institution's incident response teams review the incident response database and may take further legal action against the malicious actors.

B. *Quantum-assisted activation function for IDS*

The proposed model of a quantum layer is in the form of parameterized quantum circuits (PQCs) [10]. Formally, PQCs receive a classical input vector $\mathbf{x} \in \mathbb{R}^{n\times 1}$ and begin with an initial quantum state $|\psi\rangle$ of at least n qubits with an optional set of additional trainable parameters $\mathbf{W}$ of suitable dimension. Parameterized unitary operations U_x and U_W are defined that act on $|\psi\rangle$ in the order $U_x U_W|\psi\rangle$ or $U_W U_x|\psi\rangle$. Finally, the expectation value of a set of observables is estimated from measurement. The optional classical post-processing can be applied to map the expectations to the desired output vector with elements from $\mathbb{R}$. In the proposal, trainable parameters $\mathbf{W} \in \mathbb{R}^{k\times n}$ are defined where k denotes the desired dimension of the output from the quantum layer and n denotes the number of qubits in the current layer. Here let k equal the batch size in the training step. The initial quantum state $|\psi\rangle$ is the uniform superposition state of all

Algorithm 1: QuantumLayer(): Algorithm for operation of quantum layer

```
Require QUBITS = 8; input_parameters[8]
quantumRegister = qiskit.QuantumRegister(QUBITS)
quantumCircuit = qiskit.QuantumCircuit(QUBITS)
classicalCircuit = qiskit.ClassicalCircuit(QUBITS)
for $q_i \in$ quantumRegister and $c_i \in$ classicalRegister do
    -Apply hadamard gate $\mathcal{H}$ to $q_i$
    -Apply controlled-Y gate with input_parameters[i]
     between $q_i$ and $q_{i+1}$
    -Apply controlled-Y gate with input_parameters[i]
     between $q_i$ and $q_{i-1}$
    -Measurement: Map $q_i$ to $c_i$
Compute $Z$ expectation on the returned measurements
Send the output to the next layer in the neural network
```

Algorithm 2: Algorithm to construct hybrid classical-quantum neural network **ARCH1**

```
inputLayer = tensorflow.keras.layers.Input(120)
layer1 = tensorflow.keras.layers.Dense(8) (inputLayer)
layer2 = QuantumLayer (layer1)
layer3 = tensorflow.keras.layers.Dense(10) (layer2)
outputLayer = tensorflow.keras.layers.Dense(23) (layer3)
```

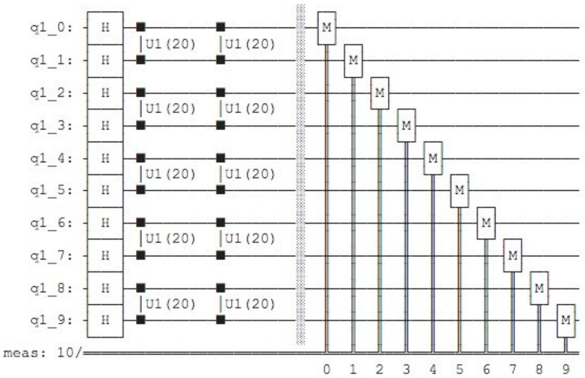


Fig. 2: The schematic diagram of the quantum circuit, in which all parameters are fixed to 20. In reality, as the reader will explore in algorithm 1, these parameters are more complex and varying over time. Moreover, the original gates are controlled R_y gates, which are optimized to U_1 by Qiskit.

n qubits:

$$|\psi\rangle = \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} |i\rangle, \tag{1}$$

where $N = 2^n$. Now, some standard matrices used elsewhere in the paper, are given below:

$$\begin{aligned} R_y(\theta) &= \begin{bmatrix} \cos\left(\frac{\theta}{2}\right) & -\sin\left(\frac{\theta}{2}\right) \\ \sin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right) \end{bmatrix}, \\ Z &= \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \\ U_1(\theta) &= \begin{bmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{bmatrix}, \end{aligned} \tag{2}$$

In a training loop working with k batches, the quantum layer receives inputs as $\mathbf{X} \in \mathbb{R}^{k \times n}$. k PQCs with each having inputs $\mathbf{x} \in \mathbb{R}^{1 \times n}$ and $\mathbf{w} \in \mathbb{R}^{1 \times n}$ are created. U_x as $R_y^{\otimes n}$ are defined with parameters from $\mathbf{x}$. Concretely,

$$U_x|\psi\rangle = R_y(x_1.\pi) \otimes ...R_y(x_i.\pi)... \otimes R_y(x_n.\pi)|\psi\rangle, \tag{3}$$

where the subscript i denotes the i-th element of the vector $\mathbf{x}$ as well as the application of the unitary $R_y(x_i.\pi)$ on the i-th qubit. Next, U_w parameterized by input $\mathbf{w}$ is defined in a way to generate entanglement between qubits:

$$U_w = \ ... CR_y(w_i.2\pi, i, i+1).CR_y(w_{i+1}.2\pi, i+1, i) \ ... \ , \tag{4}$$

where CR_y denotes the controlled-R_y operation parameterized by the first argument, with the second argument acting as the control, and the third argument acting the target. Pairs of qubits can be made and are entangled to create connections between parameters, which no classical system can make. It is worth noting why non-linearity arises from U_x and U_w. These are unitary operations which may be denoted as exponential of Hermitians. [20], thereby introducing non-linearity. Finally, the expectation of the observable Z on all qubits is calculated. Precisely, an operator is created,

TABLE I: A comparison of trainable and non-trainable parameters in the proposed architectures. **T.** implies trainable parameters while **NP.** implies non-trainable parameters. Non-trainable parameters arise because their values are updated from algorithms other than back-propagation (in this case, these belong to the quantum layers).

Arch.	Layers	T. parameters	NP. parameters
ARCH1	1	1823	510
ARCH2	2	2529	930

TABLE II: Comparison of results and architectures from other related works. It should be noted that the architectures are chosen with minimum memory footprint and considerably high accuracies from Figs. 3 for this comparison. **RNN:** recurrent neural network, **LSTM:** long short term memory network, **CNN**: convolutional neural network

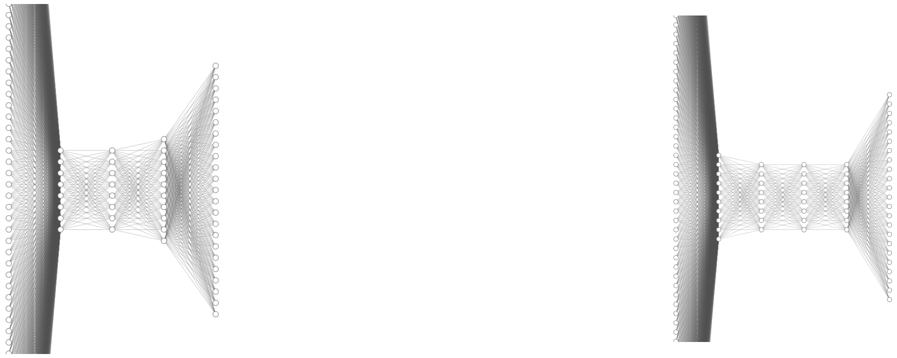
Source	Arch.	Loss	Acc.
ARCH1	120 - 8 - 8_q - 10 - 23	0.0240	0.9956
ARCH2	120 - 10 - 8 - 8_q - 8_q - 23	0.0874	0.9825
[23]	decision trees	-	0.9292
[24]	41 - 21 - 21 - 5	-	0.8023
[25]	RNN and variations	-	0.9924
[26]	LSTM	-	0.9899
[27]	41 - 100 - 100 - 100 - 100 - 5 - 5	-	0.999
[28]	LSTM	-	0.9912
[29]	CNN	-	0.9984
[30]	hidden layers with 17 nodes	-	0.9850

$$Z^{\otimes n} : Z^{\otimes n} = Z_1 \otimes Z_2 \otimes ... \otimes Z_n \ , \tag{5}$$

where the subscripts denote the qubit on which Z acts. For the quantum state $|\phi\rangle = U_w U_x |\psi\rangle$, the expectation is given as,

$$E_i = f(Re(\langle\phi \mid Z^{\otimes n} \mid \phi\rangle)) \ , \tag{6}$$

Where the subscript i denotes the i-th vector of the k-sized batch under consideration, Re denotes taking the real part of the expectation to be used by simulator, and f is a classical function that converts the expectation to a vector of suitable dimension as required by the output of the current layer (alternatively by the input of the next layer in the neural network). In this implementation, f is a linear function that creates a vector $\in \mathbb{R}^{1 \otimes n}$ from $Re(\langle\phi|Z^{\otimes n}|\phi\rangle)$ to a vector in $\mathbb{R}^{1 \otimes n}$ by multiplying with different scaling factors. The linear nature of f ensures all the non-linearity arises from the quantum portion of the computation. When considered over the entire batch of inputs $\mathbf{X} \in \mathbb{R}^{k \times n}$, the quantum layer outputs $\mathbb{R}^{k \times n}$ which is fed to further layers. The main advantage of a quantum computer over a classical computer when considering such activation functions is the inefficiency of estimating $\langle\phi|Z^{\otimes n}|\phi\rangle$ on classical computers. Several works [21], [22] point out the problems of efficiently simulating quantum measurements and estimating the expectation value of Hamiltonians on a given quantum system.



(a) ARCH1: input:120 nodes - dense:8 nodes -quantum:8 nodes - dense:10 nodes - softmax: 23 nodes.

(b) ARCH2: input:120 nodes - dense:10 nodes - dense:8 nodes - quantum:8 nodes - quantum:8 nodes - softmax: 23 node

Fig. 3: A schematic diagram of the entire architecture. From the schematic, it is clear how the architectures capture input information as large as the KDD99 dataset (120 input features after initial pre-processing), pass them on to a relatively smaller quantum layer (8 nodes), and still succeed in capturing the dataset patterns with high accuracy.

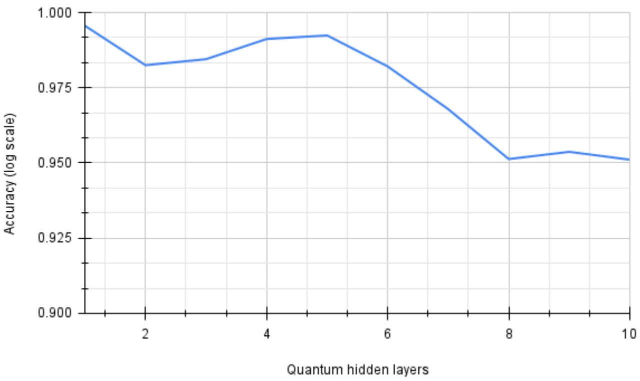


Fig. 4: The plot of test accuracy against the number of hidden quantum layers. The vertical axis is presented in log-scale. When the number of hidden quantum layers starts increasing, an improvement in accuracy can be seen. However, too many layers cause a sharp decline in test accuracy as the model begins over-fitting on train data.

IV. Experimental Results

The KDD99 dataset [31] presents an interesting dilemma: a large dataset that requires a sufficient amount of time and resources for further use of model training. The most compelling reason to use this dataset is the lack of availability to train an intrusion detection system (IDS), and KDD99 comes the closest to resembling such an ideal dataset. Most machine learning and data mining algorithms assume the static nature of the data. However, with IDS design, as more resources are now available to exploiters, the data used to train an intrusion detection system is undoubtedly dynamic. Hence, the training and testing data must constantly change to represent the most prevalent attack types at the model training time. This is also why many machine learning algorithms approaches are successful in various other domains but decreased performance in IDS studies [31].

In general, attacks can majorly be categorised into 4 major types [32]:

(1) Denial of Service (DOS) Attacks: Such types of attacks attempt to overload the server's resources to force the server into ignoring legitimate requests and denying access to legitimate users. Various DOS attacks exist; some attempt to abuse a legitimate feature, others try to create malicious packets to disrupt the TCP/IP stack of the recovery machine and even take advantage of bugs in the network daemon.

(2) User to Root (U2R) Attacks: These attacks are characterized by an exploiter, starting as a non-root user managing to exploit some vulnerability and hence gaining root access to the system. The infamous buffer overflow and load module attacks are among the most prominent U2R exploits.

(3) Remote to Local (R2L) Attacks: This involves an attacker having no association with a machine but having access to a network and gaining access as a local user on that machine.

(4) Probe Attacks: They are, by design, aimed at collating information about a network from an external source, not necessarily belonging to the network in a general sense.

Table III shows a sample of the KDD99 dataset, and the various features and their sources are also listed. Hence, for a detection system to have real-world applications, it must achieve high accuracy for intrusion prevention against all attacks. The factors that make KDD99 a suitable dataset to train such a model are :

(1) It has 24 attack types in training and 14 more attacks in testing for 38 attacks. The 14 new attacks are theoretically challenging for machine learning-based IDS to detect.

(2) It is a dataset with a particular bias towards attack instances. Although approximately 80% of the dataset is attack traffic, typical networks have normal samples accounting for 99.99% of the traffic flow.

(3) Certain duplicate records exist in both training and testing datasets, resulting in a bias on results for DOS attacks and normal instances, which constitute almost all the traffic on a network.

A. Results

The implementation details are as such: the experiments were performed on Google Colab CPU using TensorFlow Keras and Qiskit. Two different architectures are considered here: **ARCH1** (Fig. 3a, where one quantum layer is sandwiched between classical layers) and **ARCH2** (Fig. 3b, where two quantum layers are sandwiched between classical layers). Both architectures receive input from $\mathbb{R}^{120}$. In **ARCH1**, a dense layer of 8 nodes serves to set up the input for the quantum layer. The quantum layer is connected to a dense layer of 10 nodes connected to a softmax layer of 23 nodes. A total of 1823 trainable parameters are present in **ARCH1**. **ARCH2**, on the other hand, has two quantum layers and a total of 2529 trainable parameters. As training hyper-parameters, the number of epochs (the number of iterations to train the model for) is set to 1000, the batch size (the number of training examples to use in one iteration) to 64, the learning rate (the step size for the learning algorithm) is set to 0.01, the Adam optimizer is used with a decay (the change in learning rate over time) of 0.001, and the categorical cross-entropy is used as a loss function. The results are summarised in Table II.Algorithms 1 and 2 are designed to reproduce **ARCH1**. The algorithm 2 details how the different layers are connected to form one end-to-end architecture. On the other hand, the algorithm 1 details the operations of the quantum layer (with a sample circuit given in Fig. 2). It is shown that Hadamard gates are applied on all qubits to enable uniform superposition, after which pairwise controlled R_y are performed to introduce entanglement (where R_y is a parameterized rotational gate along the y axis). Note that Qiskit optimizes the R_y gates to U_1 gates, where U_1 is a phase gate). Other architectures can be replicated precisely in the same manner. The parameter sizes for **ARCH1** and **ARCH2** are summarized in Table I. Suppose the number of quantum layers increases. In that case, the number of non-trainable parameters also increases (trainable parameters are the parameters that can be trained by back-propagation, and non-trainable parameters cannot be trained by back-propagation and need other algorithms to adapt). All the nodes in a certain hidden layer are connected through weight vectors to all the nodes in the next hidden layer. The trainable and non-trainable parameters define weight vectors. For **ARCH1**, the variation of train accuracy and loss over time can be seen as in Fig. 5. Here train accuracy implies learning what the model is doing over time, whereas the loss implies the prediction inaccuracies. A steady increase in train accuracy and decline in loss implies the model is learning fast on the training dataset.

TABLE III: A listing of set of features defined for connection records. These features help in classifying normal connection requests and attacks

Feature	*Source of feature data*	*Type of feature*
duration	TCP connection feature	continuous
protocol_type	TCP connection feature	discrete
source_bytes	TCP connection feature	continuous
destination_bytes	TCP connection feature	continuous
number_root	system logs	continuous
number_shells	system logs	continuous
number_access_files	system logs	continuous
number_failed_logins	system logs	continuous
syn_error_rate	capturing traffic metrics	continuous
rej_error_rate	capturing traffic metrics	continuous
service_count	capturing traffic metrics	continuous
different_service_rate	capturing traffic metrics	continuous
same_service_rate	capturing traffic metrics	continuous

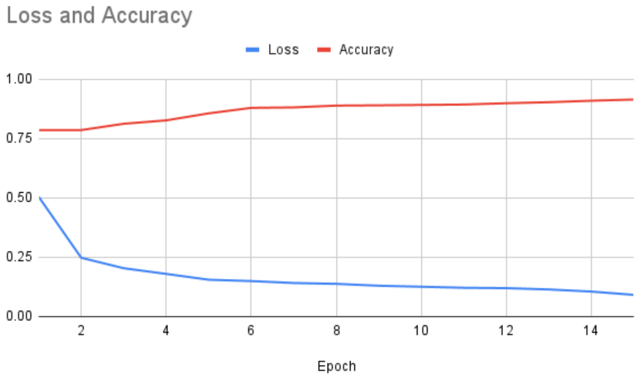


Fig. 5: Plot of train accuracy and loss over epochs/iterations while training **ARCH1**.

V. Discussion

From Table II and Fig. 3, it is apparent that quantum layers enable us to design smaller architectures that can still capture the input-output relationship in a large dataset as KDD99. When compared to classical solutions, it is observed that they either use complex learning architectures such as RNNs (recurrent neural networks) [25], CNNs (convolutional neural networks) [29], and LSTMs (long-short term memories) [26]. The architectures that used only feed-forward neural networks had linearly more hidden layers and nodes than the other architectures (thus exponentially more hidden parameters to train and validate). Moreover, with such a sharp rise in the number of hidden parameters, doubts arise about the results of such dense architecture, the more susceptible to over-fitting, and most work does not provide safeguards against over-fitting in their solutions.

VI. Conclusion

Recently, the healthcare industry has benefited from IoT, blockchain, and artificial intelligence technologies. However, various sensor devices collect and share sensitive data with a remote server. Therefore, a considerable challenge poses to healthcare network security as the attack surface widens, so the patients' lives stand at an increased risk in case of such a breach. The fact that quantum systems, being non-deterministic by nature, provide an excellent structure to build upon, especially in dynamic fields like healthcare, where the attack types change every single second, and the intention of every user may be doubtful. So, we have proposed a robust, portable, and efficient quantum-assisted hybrid neural network function as an intrusion detection system for healthcare networks. The proposed architectures tend to be more efficient (similar accuracy with lesser memory footprints) than classical counterparts for intrusion detection. We have successfully realized an architecture consisting of 8 nodes in the quantum layer, which achieves an equivalence performance compared to more complex classical counterparts. This improvement can open further investigation for applying the principles of hybrid quantum-classical neural networks to solve complex use cases in the future. As the number of qubits available for deployment increases, more complex quantum activation might seem possible in the near future which may be able to capture significantly more features in the dataset and fit more accurately on the given data.

References

[1] R. Anderson, C. Barton, R. Böhme, R. Clayton, M. J. G. van Eeten, M. Levi, T. Moore, and S. Savage, *Measuring the Cost of Cybercrime*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 265–300. [Online]. Available: https://doi.org/10.1007/978-3-642-39498-0_12

[2] S. Monteith, M. Bauer, M. Alda, J. Geddes, P. C. Whybrow, and T. Glenn, "Increasing cybercrime since the pandemic: Concerns for psychiatry," *Current Psychiatry Reports*, vol. 23, no. 4, p. 18, Mar 2021. [Online]. Available: https://doi.org/10.1007/s11920-021-01228-w

[3] J. Jang-Jaccard and S. Nepal, "A survey of emerging threats in cybersecurity," *Journal of Computer and System Sciences*, vol. 80, no. 5, pp. 973–993, 2014, special Issue on Dependable and Secure Computing. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0022000014000178

[4] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, "Deep learning approach for intelligent intrusion detection system," *IEEE Access*, vol. 7, pp. 41 525–41 550, 2019. [Online]. Available: https://doi.org/10.1109/ACCESS.2019.2895334

[5] N. Mishra, M. Kapil, H. Rakesh, and *et. al.*, "Quantum machine learning: A review and current status," in *Data Management, Analytics and Innovation*, N. Sharma, A. Chakrabarti, V. E. Balas, and J. Martinovic, Eds. Singapore: Springer Singapore, 2021, pp. 101–145. [Online]. Available: https://doi.org/10.1007/978-981-15-5619-7_8

[6] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, "Quantum machine learning," *Nature*, vol. 549, no. 7671, pp. 195–202, sep 2017. [Online]. Available: https://doi.org/10.1038%2Fnature23474

[7] P. Wang, "Artificial general intelligence and classical neural network," *2006 IEEE International Conference on Granular Computing*, pp. 173–190, 2006. [Online]. Available: https://doi.org/10.1109/GRC.2006.1635771

[8] S. Haykin, *Neural Networks: A Comprehensive Foundation*, 2nd ed. USA: Prentice Hall PTR, 1998.

[9] A. Kariya and B. K. Behera, "Investigation of Quantum Support Vector Machine for Classification in NISQ era," *arXiv:2112.06912*, December 2021. [Online]. Available: https://arxiv.org/abs/2112.06912

[10] M. Benedetti, E. Lloyd, S. Sack, and M. Fiorentini, "Parameterized quantum circuits as machine learning models," *Quantum Science and Technology*, vol. 4, no. 4, p. 043001, November 2019. [Online]. Available: https://doi.org/10.1088/2058-9565/ab4eb5

[11] Y. Kwak, W. Yun, S. Jung, and J. Kim, "Quantum neural networks: Concepts, applications, and challenges," in *2021 Twelfth International Conference on Ubiquitous and Future Networks (ICUFN)*, August 2021, pp. 413–416. [Online]. Available: https://doi.org/10.1109/ICUFN49451.2021.9528698

[12] A. Gouveia and M. Correia, "Towards Quantum-Enhanced Machine Learning for Network Intrusion Detection," in *2020 IEEE 19th International Symposium on Network Computing and Applications (NCA)*, November 2020, pp. 1–8. [Online]. Available: https://doi.org/10.1109/NCA51143.2020.9306691

[13] E. D. Payares and J. C. Martinez-Santos, "Quantum machine learning for intrusion detection of distributed denial of service attacks: a comparative overview," *Proceedings Volume 11699, Quantum Computing, Communication, and Simulation*, vol. 11699, 2021. [Online]. Available: https://doi.org/10.1117/12.2593297

[14] A. Thirumalairaj and M. Jeyakarthic, "Perimeter intrusion detection with multi layer perception using quantum classifier," in *2020 Fourth International Conference on Inventive Systems and Control (ICISC)*, 2020, pp. 348–352. [Online]. Available: https://doi.org/10.1109/ICISC47916.2020.9171159

[15] H. Yong and F. Z. Xue, "Quantum growing hierarchical self organized map-based intrusion detection system," in *2010 International Conference on System Science, Engineering Design and Manufacturing Informatization*, vol. 2, 2010, pp. 110–115. [Online]. Available: https://doi.org/10.1109/ICSEM.2010.118

[16] Y. Dong, W. Hu, J. Zhang, M. Chen, W. Liao, and Z. Chen, "Quantum beetle swarm algorithm optimized extreme learning machine for intrusion detection," *Quantum Information Processing*, vol. 21, no. 1, p. 9, January 2022. [Online]. Available: https://doi.org/10.1007/s11128-021-03311-w

[17] C. A. Bishop, T. S. Humble, R. S. Bennink, and B. P. Williams, "Intrusion detection based on quantum interference," in *CLEO: 2013*, 2013, pp. 1–2. [Online]. Available: https://doi.org/10.1364/CLEO_AT.2013.JW2A.77

[18] X. Yin, "Quantum evolutionary algorithm based network intrusion detection," in *2010 3rd International Conference on Computer Science and Information Technology*, vol. 4, Chengdu, China, 2010, pp. 683–685. [Online]. Available: https://doi.org/10.1109/ICCSIT.2010.5564538

[19] J. Wang, C. Liu, X. Shu, H. Jiang, X. Yu, J. Wang, and W. Wang, "Network intrusion detection based on xgboost model improved by quantum-behaved particle swarm optimization," in *2019 IEEE Sustainable Power and Energy Conference (iSPEC)*, 2019, pp. 1879–1884. [Online]. Available: https://doi.org/10.1109/iSPEC48194.2019.8975295

[20] J. R. McClean, S. Boixo, V. N. Smelyanskiy, R. Babbush, and H. Neven, "Barren plateaus in quantum neural network training landscapes," *Nature communications*, vol. 9, p. 4812, 2018. [Online]. Available: https://doi.org/10.1007/s11128-021-03311-w

[21] A. Ambainis, "On physical problems that are slightly more difficult than qma," *2014 IEEE 29th Conference on Computational Complexity (CCC), IEEE*, 2014. [Online]. Available: https://doi.org/10.1109/CCC.2014.12

[22] Z. Chen and X.-S. Gao, "Qdnn: Dnn with quantum neural network layers," *arXiv:1912.12660*, 2019.

[23] I. Levin, "KDD-99 classifier learning contest LLSoft's results overview," *ACM SIGKDD Explorations Newsletter*, vol. 1, no. 2, pp. 67–75, 2000. [Online]. Available: https://doi.org/10.1145/846183.846201

[24] H. Ji, D. Kim, D. Shin, and D. Shin, "A Study on comparison of KDD CUP 99 and NSL-KDD using artificial neural network," *Advances in computer science and ubiquitous computing, Springer, Singapore*, pp. 452–457, 2017. [Online]. Available: https://doi.org/10.1007/978-981-10-7605-3_74

[25] C. Xu, J. Shen, X. Du, and F. Zhang, "An intrusion detection system using a deep neural network with gated recurrent units," *IEEE Access*, vol. 6, pp. 48 697–48 707, 2018. [Online]. Available: https://doi.org/10.1109/ACCESS.2018.2867564

[26] S. Xiao, J. An, and W. Fan, "Constructing an Intrusion Detection Model based on Long Short-term Neural Networks," *2018 IEEE/ACIS 17th International Conference on Computer and Information Science (ICIS)*, 2018. [Online]. Available: https://doi.org/10.1109/ICIS.2018.8466445

[27] Y. Jia, M. Wang, and Y. Wang, "Network intrusion detection algorithm based on deep neural network," *IET Information Security*, vol. 13, no. 1, pp. 48–53, 2019. [Online]. Available: https://doi.org/10.1049/iet-ifs.2018.5258

[28] C. Luo, L. Wang, and H. Lu, "Analysis of LSTM-RNN based on attack type of KDD-99 dataset," *International Conference on Cloud Computing and Security*, 2018. [Online]. Available: https://doi.org/10.1007/978-3-030-00006-6_29

[29] N. Ding, Y. Liu, Y. Fan, and D. Jie, "Network Attack Detection Method Based on Convolutional Neural Network," *Chinese Intelligent Systems Conference, Springer, Singapore, 2019*, 2018. [Online]. Available: https://doi.org/10.1007/978-981-32-9686-2_68

[30] D. E. Kim and M. Gofman, "Comparison of shallow and deep neural networks for network intrusion detection," *2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC)*, 2018. [Online]. Available: https://doi.org/10.1109/CCWC.2018.8301755

[31] A. Özgür and H. Erdem, "A review of KDD99 dataset usage in intrusion detection and machine learning between 2010 and 2015," April 2016. [Online]. Available: https://doi.org/10.7287/PEERJ.PREPRINTS.1954

[32] Lincoln Laboratory, Massachusetts Institute of Technology, "Intrusion detection attacks database archives," 1999, [Accessed 11-February-2022]. [Online]. Available: https://archive.ll.mit.edu/ideval/docs/attackDB.html#dos