IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 1, 2021, accepted September 28, 2021, date of publication October 4, 2021, date of current version October 12, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3117405

Machine Learning Techniques for Detecting Attackers During Quantum Key Distribution in IoT Networks With Application to Railway Scenarios

HASAN ABBAS AL-MOHAMMED[1], AFNAN AL-ALI[1], ELIAS YAACOUB[1], (Senior Member, IEEE), UVAIS QIDWAI[1], KHALID ABUALSAUD[1], (Senior Member, IEEE), STANISŁAW RZEWUSKI[2], AND ADAM FLIZIKOWSKI[2]

[1]Computer Science and Engineering Department, Qatar University, Doha, Qatar
[2]IS-Wireless, 05-500 Piaseczno, Poland

Corresponding author: Elias Yaacoub (eliasy@ieee.org)

This publication was jointly supported by Qatar University and IS-Wireless - International Research Collaboration Co-Fund Grant no. IRCC-2021-003. The findings achieved herein are solely the responsibility of the authors. Open Access funding was provided by the Qatar National Library.

ABSTRACT Internet of Things (IoT) deployments face significant security challenges due to the limited energy and computational power of IoT devices. These challenges are more serious in the quantum communications era, where certain attackers might have quantum computing capabilities, which renders IoT devices more vulnerable. This paper addresses the problem of IoT security by investigating quantum key distribution (QKD) in beyond 5G networks. An algorithm for detecting an attacker between a transmitter and receiver is proposed, with the side effect of interrupting the QKD process while detecting the attacker. Afterwards, Artificial neural network (ANN) and deep learning (DL) techniques are proposed in order to detect the presence of an attacker during QKD without the need to disrupt the key distribution process. An architecture for implementing QKD in beyond 5G IoT networks is proposed, offloading the heavy computational tasks to IoT controllers. In addition, an implementation scenario for securing IoT communications for sensors deployed in railroad networks is described. The results show that the proposed ML techniques can reach 99% accuracy in detecting attackers.

INDEX TERMS 5G and beyond, IoT security, quantum key distribution, machine learning, railway communications.

I. INTRODUCTION

The deployment of billions of internet of things (IoT) devices under the fifth generation (5G) networks [1]–[3] is expected to increase under sixth generation (6G) networks [4]. The massive machine type communication (mMTC) 5G use case takes into account this deployment [1]–[3]. In addition, mission critical services relying on the deployment of IoT devices are also increasing, thus causing the transition from the ultra-reliable low-latency communications (URLLC) use case in 5G into massive URLLC (mURLLC) in 6G [4].

This increases the security challenges faced by IoT devices, due to the limited battery energy and computational power of many of these devices [1]–[3]. These challenges are expected to be exacerbated in beyond 5G networks, notably due to the advances in quantum communications, as IoT devices might have to face attackers that are equipped with quantum computing capabilities [5]–[7].

This paper addresses this problem, and proposes the use of quantum cryptography techniques in order to protect IoT devices in the beyond 5G and 6G era. However, it is impractical for low-power IoT devices to support advanced quantum communications. However, this can be taken care of by the IoT controllers. Indeed, these controllers are more powerful devices that are deployed in several IoT networks, where each controller is in charge of handling data collection, aggregation, and processing from a group of several IoT sensors. Afterwards, the controllers transfer this data over the network to remote servers in the cloud where it is stored, processed and analyzed [8].

The associate editor coordinating the review of this manuscript and approving it for publication was Ibrar Yaqoob.

Given their superior power and computational capabilities, the controllers can use quantum communications to exchange long encryption keys with the server. Therefore, the approach proposed in this paper consists of performing quantum key distribution (QKD) between the server and the controllers. Afterwards, each controller can distribute the generated keys to the IoT devices connected to it. Then, these devices can encrypt their data while transmitting it to the controller over the traditional radio frequency (RF) communication links between them. Thus, QKD is performed by the controllers to protect the weak 'IoT device - Controller' link from malicious attackers.

The work in [9] is the closest we could find in the literature that it related to the contributions of this paper. In [9], software defined networking was used to demonstrate QKD experimentally, with the controller connected using fiber optics. Energy savings were obtained using the proposed approach. In our previous work [10], we considered a general approach that can be used with either fiber or free space optics (FSO). Moreover, we described in detail the QKD process and investigated the various obtained key lengths in different conditions. However, our work in [10] did not involve any security investigation for protecting the QKD process itself in the presence of an attacker.

In this paper, we investigate the security of QKD in the presence of an attacker. Artificial neural network (NN) and deep learning (DL) techniques are used to identify if an attacker is present or not, and results show high accuracy of the proposed method. The results are applicable to any IoT network where controllers can have both an optical link and a traditional radiofrequency (RF) link, with particular emphasis on securing control and management information in railroad networks. Most of the exiting work that used machine learning with QKD to enhance the communication is summarized in the survey paper [11].

Hence, the main contributions of this paper can be summarized as follows:

- Proposing an architecture for performing QKD in IoT networks, without affecting the computation and power consumption limitations of the IoT sensors,
- Describing the implementation of the proposed architecture in a railroad IoT scenario as a practical example,
- Designing a simple algorithm for detecting an attacker between a transmitter and receiver, without resorting to machine learning techniques, with the side effect of interrupting the QKD process while detecting the attacker,
- Overcoming the limitations of the algorithm by implementing artificial neural network (ANN) and deep learning techniques to detect the presence of an attacker with high accuracy, and
- Comparing the performance of the machine learning techniques in terms of accuracy and speed.

The rest of this paper is organized as follows. Section II presents the related work. Section III describes quantum key distribution. The system model is described in Section IV. Section V describes a proposed method to detect an attacker without ML, whereas the proposed ML method is discussed in Section VI. The results of the ML techniques for attacker detection are presented and discussed in Section VII. Finally, Section VIII concludes the paper.

II. RELATED WORK

To the best of the authors' knowledge, no previous works in the literature detect an attacker using machine learning in a QKD scenario based on the final key length. All previous works have used machine learning in QKD for other implementations. For example, in [12], machine learning is used to detect wavelength attacks. The framework of [12] suggests an intelligent control technology based on optical spectrum analysis. An irregular optical spectrum signal can be automatically observed by using the linear discriminant analysis support vector machine algorithm through the machine learning-based optical spectrum analysis methodology, so as to understand attack identification and device intelligent monitoring. The results of [12] demonstrate that the linear discriminant analysis support vector machine algorithm can correctly distinguish the original spectral data and the irregular spectral data after the attack.

In [13], machine learning is used to detect different attack techniques that undermine the functional security of a continuous-variable quantum key distribution (CVQKD) framework. The authors of [13] suggest a security technique for CVQKD systems to the most recognized forms of attacks. They analyzed multiple pulse characteristics that would be influenced by various types of attacks, extracted a feature vector based on these characteristics as an artificial neural network (ANN) model input, and illustrated the ANN model's preparation and testing method for attack identification and classification. Simulation findings demonstrate that most of the known attacks can be detected successfully by the proposed scheme at the expense of reducing a limited portion of the hidden keys and transmission distance.

In [14], [15], the authors have used random forest ML algorithm to choose the optimal QKD protocol for communication no matter whether the distances are long or short between the sender and the receiver.

III. QUANTUM KEY DISTRIBUTION

The QKD process requires the existence of a transmitter (Alice), a receiver (Bob), and two communicating networks. The first network is a quantum channel connected to the transmission of quantum random-bit signals between transmitter and receiver, whereas the second network is a conventional channel [16].

Alice has to send a stream of random photons to Bob. In order to do that, she uses polarized filters such that each photon in the stream would have one out of four distinct polarizations: rectilinear polarizations of 0°, 90° and diagonal polarizations of 45° and 135°. Alice and Bob agree arbitrarily on which of these states correspond to a "0" bit and which

TABLE 1. The polarization states and corresponding bit represented.

Polarization/ bit	0	1
Rectilinear +	↔	↕
Diagonal x	⤢	⤡

ones correspond to a "1". For example, as shown in Table 1, 0° and 45° represent "0", whereas 90° and 135° correspond to "1".

In the first step, Alice produces a stream of randomly polarized photons. Then, she transmits them over the quantum channel.

At the receiving end, Bob uses two detectors:

- A rectilinear filter (+): Photons with a rectilinear state pass through this filter without undergoing any change to their polarization. However, the rectilinear filter switches the state of a diagonally polarized photon (45° or 135°) passing through the filter into one of the rectilinear states (0° or 90°).

- A diagonal filter (x): Photons with a diagonal state pass through this filter without undergoing any change to their polarization. However, the diagonal filter switches the state of a rectilinearly polarized photon (0° or 90°) passing through the filter into one of the diagonal states (45° or 135°).

Consequently, Bob cannot guess the polarization of the photons. The best he can do is to arbitrarily guide each photon, as it arrives, to one of his two detectors. Then, Alice and Bob interact over the traditional RF communication channel to address Bob's detector preference, and eliminate the bits corresponding to incorrect filter selection by Bob. The remaining correct bits are then used to extract a shared secret key.

Any attacker (Eve), who tries to intercept the communications between Alice and Bob in order to capture the key, will not be able neither to access nor to guess the polarization filters used at the sender or the receiver [17]. The best thing Eve can do is to follow the same approach used by Bob, by randomly selecting polarization filters to intercept the photons. When Eve's choice of detector is correct, the photon will continue on its path to the destination with its prior polarization. However, when Eve makes the wrong decision, this will lead to changing the polarization of the photon before it continues its way to the receiver (Bob).

Thus, the photon stream reaching Bob will be modified due to Eve's intervention. After interacting with Alice over the traditional RF communication channel, Alice and Bob perform a final check before deciding to use the exchanged mutual key for encrypting the data: They randomly select and compare a number of bits from their key streams. If the error exceeds a certain threshold (pre-agreed upon between them), they discard the whole key stream and repeat the process to generate a new key.

IV. SYSTEM MODEL

In this section, we describe the system model adopted in this paper. It is based on a proposed architecture that allows QKD to take place in an IoT scenario, without affecting the power-limited and computation-limited IoT sensors. The proposed architecture limits the QKD process to an exchange between the server and IoT controllers. Moreover, the machine learning techniques described in Section VI are implemented at the server. Thus, no additional power consumption is incurred at the IoT sensors. A key for each Controller-Sensor link is exchanged between the server and controller using QKD. Afterwards, the controller exchanges the key with each sensor using traditional key distribution techniques over the wireless Controller-Sensor channel. Consequently, the power consumption for the computation and exchange of a long secure key is offloaded to the server and controller.

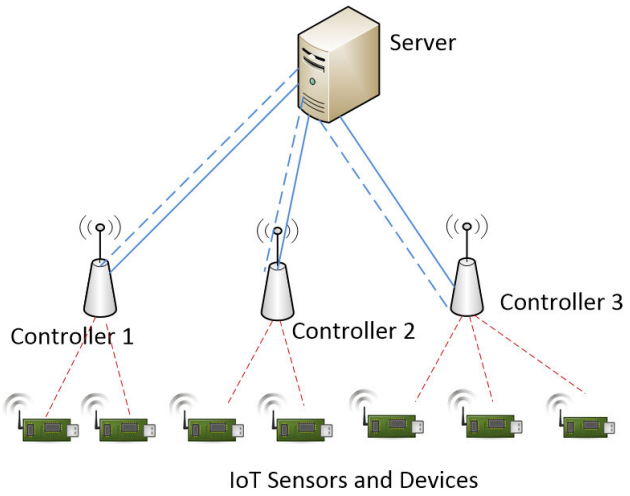

FIGURE 1. The structure of controller, IoT devices, and server/base station sharing the quantum key.

Fig. 1 shows the main configuration between the server and the IoT devices, where we consider an example with three IoT controllers, each connected to several IoT sensors that exchange information with the server. In this figure, solid lines indicate optical connections, whereas dashed lines indicate wireless connections.

A. USE CASE SCENARIO 1: MOBILE HEALTH (mHEALTH)

The scenario of Fig. 1 could be implemented in a mHealth scenario for example. In this case, IoT sensors would be placed over a patient's body to form a body area network (BAN), and will send their measurements of the patient's vital signs to a local controller (which could be the patient's smartphone for example). The controller would then send this data to a local WiFi access point (AP) for example, from which it will be forwarded to a cloud server for storage and further processing and analysis. QKD could be performed over an FSO link between the AP and controller (in case they are equipped with an optical channel in addition to their RF channels). When the exchanged secret key is used to encrypt the data transmitted over the RF communication links, this

would protect the privacy of the patient's data from potential eavesdroppers.

B. USE CASE SCENARIO 2: RAILROAD NETWORKS

Another example is the implementation of the model of Fig. 1 in railroad networks. In this case, the rail track parameters (temperature, tilt, dip, shock, and vibration measurement for example) can be monitored in real-time using a variety of IoT sensors [18], [19]. The sensors' measurements would be sent to an IoT controller, and from there they will be forwarded to a control center where the railroad parameters are monitored in real-time for the purpose of maintaining the safety of the track. In this scenario, QKD would be performed over fiber optic cables. In fact, it is common for fiber optic cables to be deployed along the rail tracks (this can help in providing backbone internet connectivity for remote areas). Hence, the remote server and the local controller could perform QKD multiple times over the fiber optic cable. Afterwards, the keys obtained can be used for encrypting the data transmitted over the RF wireless communications on the "IoT Sensors – Controller" links. Various technologies could be used on these links, e.g., millimeter wave (mmWave) communications, Zigbee, or 5G mMTC communications over more "traditional" frequencies.

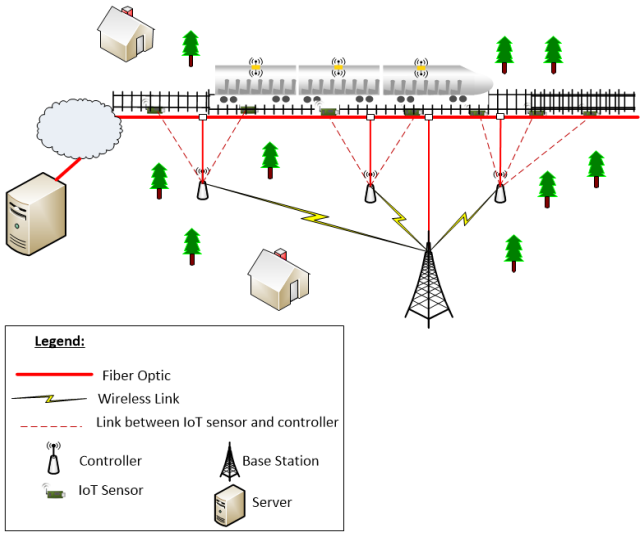

FIGURE 2. Physical implementation of the logical scenario of Fig. 1 in a railroad network.

Fig. 2 shows the implementation of the logical scenario of Fig. 1 in a railroad network. Fiber can be deployed at lower costs by using micro-trenching on the border of roadways [20]. Several fiber optic cables, each containing a multitude of fibers, can run parallel to the rail track. A pair of fibers could be extracted from the cable and allocated to each IoT controller (as shown in Fig. 2), to perform QKD over the optical link. The RF link where the data, encrypted using the symmetric keys exchanged through QKD, will be transmitted, corresponds to the connection between the controllers and possibly WiFi access points, Zigbee controllers, or cellular base stations (BSs) and/or remote radio heads (RRHs) deployed along the rail track (Fig. 2 shows a wireless connection to the BS, which is also connected to the 5G core network through a fiber connection).

V. DETECTING ATTACKERS WITHOUT ML

This section presents an approach that can be followed between Alice and Bob (IoT controllers and server) to detect the presence of an attacker without resorting to ANNDL techniques. However, it requires that the two parties occasionally interrupt their communications to transmit a pre-agreed sequence of data, with the sole purpose of detecting whether an attacker is present or not. The approach presented in Section VI allows integrating the attacker detection process with the regular QKD process, by using ANN and DL, thus providing a smooth operation and avoiding communication disruption.

A. QUANTUM KEY DISTRIBUTION IN THE PRESENCE OF AN ATTACKER

In this section, we analyze the impact of an attacker on the exchanged key length between the sender and receiver. In the scenario of Figs. 1 and 2, we simulate the transmission of 1000 photons for each of the three controllers shown in these figures. Fig. 3 displays the key lengths securely obtained by each IoT controller.

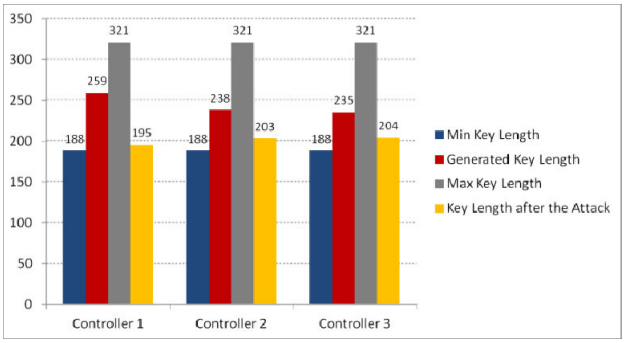

FIGURE 3. The key length for each controller before and after the attack.

The figure also indicates the maximum (321 bits) and minimum (188 bits) numbers of potential key lengths that have been measured in the simulation environment.

From Fig. 3, it can be noted that before the attack, the key lengths obtained for Controllers 1, 2, and 3 are 259, 238, and 235 bits, respectively. After the attack, the correctly exchanged key lengths (bits not altered due to the intervention of the attacker) become 195, 203, and 204 bits for Controllers 1, 2, and 3, respectively. Interestingly, although no measures were taken to mitigate the attack, it can be noted that these lengths are still longer than keys used in typical encryption algorithms, like AES where the length is 192 bits.

B. METHOD FOR DETECTING AN ATTACKER WITHOUT USING ML

This section describes an approach that can be applied regularly between sender and receiver in order to check if there

is a man-in-the-middle attack. Thus, this is a "discovery" process to detect if there is an attacker, not an actual key generation process (but the attacker does not know this and would attempt to detect a "key").

In this approach, Alice and Bob will mimic the actual QKD process by transmitting a pre-agreed upon sequence of N photons where Bob knows the polarization filters to use. Consequently, in the absence of attack, the agreement between Alice and Bob should be for 100% of the photons. On the other hand, if an attacker (Eve) is present, it is extremely unlikely for her to guess 100% of the correct polarization filters, especially when N is large (in the results of this section we consider $N = 1000$ for illustration purposes). Therefore, whenever she makes an incorrect guess and alters the photon polarization, this will be detected by Bob.

Algorithm 1 Algorithm for Simulating the Detection of an Attacker (Man in the Middle)

```
1:  attacker[] = 0; // declaration array for attacker detecting keys.
2: attacked_ph = 0; // initialization of n of attacked photons
3: A[] = N;// the agreed of polarization photons ( e.g., N = 1000) sent to Bob(B)
4: B[] = N;// the agreed of polarization photons (e.g., N = 1000) received.
5: for i: = N;//the agreed photons number.
6:      If attacker[i] == A[i]&& attacker[i] == B[i].
7:           det_key[i] = A[i];// correctly detected by attacker
8:   for k: = N;// number of agreed photons (e.g., N = 1000)
9:       if B[k] ! = A[k]//changed by the attacker due to wrong filter choice
10:          attacked_ph = attacked_ph + 1//number of photons attacked.
```

Algorithm 1 demonstrates this approach. The number of predicted photons at the receiver should be N, so a number significantly less than N means that an attacker is attempting to capture the key,

Since each of the loops at Lines 5 and 8 of Algorithm 1 corresponds to N iterations, the algorithm has a complexity of order N, O(N).

C. SIMULATION RESULTS

This section shows the implementation of Algorithm 1 between sender and receiver for checking if there is a man-in-the-middle attack. Alice and Bob will transmit a pre-agreed upon sequence of photons where Bob knows the polarization filters to use. As discussed previously, this is a "discovery" process to detect if there is an attacker, not an actual key generation process (but the attacker does not know). The amount of photons transmitted from the controller to the base station is seen in Fig. 4. The number of photons is $N = 1000$. In comparison, the number of photons correctly detected by the intruder is 251, and the number of photons arriving at the destination without changed polarization is 749. The sender and receiver will know that they have an attacker in the middle attempting to determine the transmitted key. Consequently, they will take this into account while trying to regenerate their actual key.

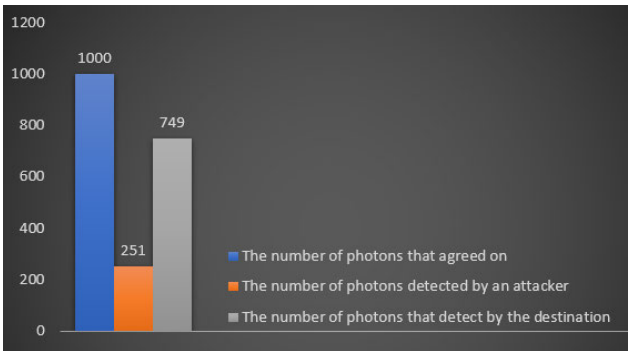

FIGURE 4. The number of agreed photons, the number of detected polarizations, and the number of photons with correct polarizations at the destination.

VI. ARTIFICIAL NEURAL NETWORK AND DEEP LEARNING TECHNIQUES FOR DETECTING ATTACKERS

Due to the computational power and the performance of the server, the machine learning algorithms are implemented at the server-side, where all the training and testing data will be processed. This data is represented by the generated key's length (quantum key) that will be sent to the server for testing. The IoT devices and the controller will not be affected by the power consumption due to using machine learning and deep learning for detecting the attacker.

Non-linear mathematical data structures that replicate the function of biological neural networks (NNs) are referred to as artificial neural networks (ANNs) which are considered the most widely studied and used approach for predictive patterns [21], [22]. In many challenges, ANNs can effectively model complex or multi-complex tasks, as compared to traditional sequence approaches [23]. The generalization and learning capabilities of these networks are considered as a mathematical translation of biological neural networks. There are several applications of Neural networks in various areas, like finance, space education, sports and so on. This technique has been utilized to solve diagnosis, prediction and pattern recognition problems. When the relationship between the input and output is unknown or complex then this technique is suitable to be applied [24].

The architecture of the neural network is simply consisting of three layers, the input layer where the data is received from the user, the hidden layer(s) which convert the input data into a suitable form using specific parameters (weights and bias) to be used easily later by the output. The output layer is supposed to generate the final outcomes. All the layers are composed of basic nodes called neurons. The artificial neural networks have been classified into single layer feedforward

neural network, multilayer feedforward network, recurrent network, or mesh network [25].

The main advantage of using the neural network techniques is that they are data-driven methods that can process the data without any previous restrictions or back-knowledge assumptions about the model's form. Also, these techniques can learn by training on real data which make the model able to generalize on previously unseen data. The last advantage is due to processing the data using nonlinear activation function, which allows the network to detect the complex nonlinear type of relationships between the input and the output variables. Mainly this type of machine learning techniques works on trial and error, so there is no specific structure or design that can work for all the problems. Basically, the network structure can be trained on the training part of the dataset then tested on the testing data and this process is repeated for a number of epochs and different parameters can be adjusted (like the number of hidden neurons and number of hidden layers) until reaching the least error obtained. In this case this model design will be selected [24].

Deep learning is a subfield of machine learning that can be defined by the ability to learn deep representations from data, which entails learning several levels of forms and concepts [11]. A deep learning structure is a multilayer perceptron with numerous hidden layers. In [26], the authors developed the notion of deep learning in 2006. They suggested a deep belief network (DBN)-based unsupervised greedy layer-by-layer training technique, which offered hope for solving the deep strut optimization problem. To identify distributed feature representations of data, deep learning integrates low-level characteristics to build more abstract high-level representation attribute categories or features. Both supervised and unsupervised learning tasks have shown great success with deep learning just like machine learning techniques [27], [28].

Deep learning is an effective tool for modeling and analysis. Many tasks, such as video categorization, speech recognition, and natural language processing, can be accomplished because of applying DL techniques. In difficult prediction tasks, deep learning approaches also produce high-accuracy results specifically in non-linear dynamical systems where a diverse amount of data set can be generated with some nonlinear changings [28].

The first thing that springs to mind when we have any kind of sequential information or dataset and wish to apply Deep Learning algorithms to it is Recurrent Neural Networks (RNNs). Traditional neural networks have inputs that are independent of each other, while Recurrent Neural Networks, as the name implies, conduct a repeating task for each sequence, with outputs that are reliant on past communications [29]. However, in circumstances where the input information is non-sequential, such as image captioning, where the image is a single non-sequential data point, RNN has shown excellent performance. It must be understood that the powerful framework of the Recurrent Neural Network can be employed in that case as well, even if the input and output vectors are static vectors to be processed sequentially [29]. Even if our data does not come in the form of sequences, we can still create effective models and teach them to process data in a sequential fashion [30].

In this paper, we used two types of deep learning methods to detect an attacker based on the final key length: Section VI.B describes the shallow neural network and Section VI.C presents the deep learning method, respectively.

A. THE DATA SET

The data set that has been used for the training depends on the quantum key's length. It is around 20,000 generated keys, with 10K key length generated in a safe environment (without attacker, using Algorithm 1 without the attacker lines), and the other 10K generated with the effect of the attacker in the middle (by using Algorithm 1). The input of the machine learning algorithm is 70% of this data and the remaining 30% are used for testing.

B. NEURAL NETWORK PATTERN RECOGNITION

One of the main neural networks types is the Multilayer Perceptron (MLP) that uses the multilayer feedforward architecture which is considered widely used for prediction and pattern recognition and it includes feedforward backpropagation, cascade feedforward backpropagation, and perception networks [25]. In this study, the feedforward backpropagation network was chosen based on the properties of the problem and the promising results obtained from our test. The main reason for choosing the MLP for our test instead of using pre-trained deep learning models is that deep learning needs a large number of training samples like hundreds of thousands or maybe million to get successful training, it takes more computation time, it could lead to overfitting, and in our case we do not need this type of models as the number of training samples generated for this test is several thousand as mentioned earlier [14], [15].

In this paper, we implemented two different scenarios, the first one is to generate quantum keys without any attacker on their way (ideal situation), and for the second scenario, we added an attacker in the middle. To confirm the outcomes of both scenarios, the first technique that was used to distinguish between attack keys and non-attack ones is ANN. The Neural Network Toolbox from MATALB was used for our test. This toolbox provides algorithms and pre-trained models, as well as applications for creating, training, visualizing, and simulating neural networks either with one hidden layer (called shallow neural network) or with multiple hidden layers called (deep learning neural network). For our test, we used the shallow neural network to deal with the large dataset (around 20,000 samples). The structure of our network is as shown in Fig. 5.

In our test, the learning algorithm was trained on 70% of the generated keys (training part) before being tested to predict whether there is an attacker or not for each sample in the rest of the data set (testing part). For each test set sample, a set of outcome probabilities should be assigned regarding

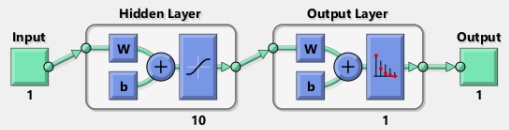

FIGURE 5. The structure of the neural network.

each of the attacker/non attacker groups—and the chosen group will be the one with highest probability. Here the test set is unseen and all the parameter adjustments were done during training to be used later, so in this case the overfitting issue was avoided to provide a robust performance.

C. LONG-SHORT TERM MEMORY (LSTM)

LSTM is a type of RNN with feedback connections, unlike standard feedforward neural networks. The presence of feedback links transforms the LSTM into a type of ''general purpose computer,'' allowing it to perform all the computations that a Turing machine can perform [31].

Memory cells are used in an LSTM neural network as independent activation functions and identity functions with fixed weights that are coupled to each other. Errors back-propagated through a neuron cannot vanish or explode because of their constant weight. In traditional RNNs, the weight matrices are also trained via backpropagation over time data, just as the standard neural network training procedure [32]. As a result, the gradients disappearing problem occurs in RNNs as the network's complexity grows, implying that typical RNNs lack the ability to find information or preserve dependencies buried in long-term time series. To deal with difficulties with long-term dependencies, LSTM was developed to prevent back-propagated mistakes from gradients vanishing or inflating in RNN [32].

The LSTM-based method delivers higher prediction accuracy for distinct zones of the time series than standard prediction methods and has been widely applied in the anomaly detection research field [33]. The LSTM design is based on a memory cell that can preserve its state over time and nonlinear gating devices that control the flow of information into and out of the cell. In contrast to standard RNNs, the LSTM neural network establishes connections among inputs and outputs using memory cells with forget gates rather than traditional neurons [32]. These implemented forget gates can effectively govern the use of data in cell states, allowing LSTM to grasp dynamic behavior in time series visual data and build effective machine representations [32].

At each time step, we can define a unit of LSTM as a collection of vectors comprising of forget gate, input gate, tanh gate and output gate, as shown in Fig 6, rather than a single layer as in a traditional RNN [31].

In the notation used in the subsequent equations, σ represents the sigmoid function, x_t corresponds to the input x at time t, and h_t is the output with respect to the input at time t. W_α, W_β, W_γ, and W_o are the weights, and b_α, b_β, b_γ, and b_o are the biases of the layers α, β, γ, and o, respectively [31], [34].

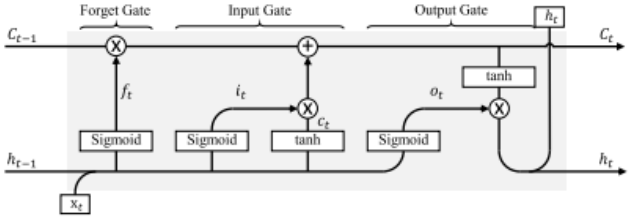

FIGURE 6. The architecture of LSTM [34].

TABLE 2. Binary classification confusion matrix.

		Predicted	
		Positive	Negative
Actual	Positive	TP	FN
	Negative	FP	TN

The first layer α is a sigmoid layer, also known as a forget gate layer, which in the previous cell state returns a value between 0 and 1, with 0 indicating no information pass and 1 indicating all information pass. The first layer's equation can be written as:

$$\alpha_t = \sigma(W_\alpha \cdot [h_{t-1}, x_t] + b_\alpha) \tag{1}$$

The input gate layer is the second layer, and it is updated as follows:

$$\beta_t = \sigma(W_\beta \cdot [h_{t-1}, x_t] + b_\beta) \tag{2}$$

The third layer is the tanh layer γ:

$$\gamma_t = \tanh(W\gamma \cdot [h_{t-1}, x_t] + b\gamma) \tag{3}$$

Then, we can update the previous state by using:

$$C_t = \alpha_t \cdot C_{t-1} + \beta_t \gamma_t \tag{4}$$

The output layer is a sigmoid function layer, whose output is expressed as:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{5}$$

Now, the cell states pass through tanh function to form the final output.

$$h_t = o_t \tanh(C_t) \tag{6}$$

D. PERFORMANCE EVALUATION METRICS

Four traditional assessment metrics, namely: accuracy, F1 score, precision and recall, were chosen. Any of these metrics is determined using the True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN) values calculated and represented by the uncertainty matrix throughout the test process. For a binary classification query, Table 2 shows the general confusion matrix.

Any of the chosen metrics will offer some insights into the model's results, which will strengthen the assessment process. A brief overview of each is shown below [35]:

TABLE 3. The evaluation metrics.

Dataset- split	Accuracy	Precision	Recall	F1-score
70%-30%	99.1%	98.8%	99.3%	99.04%

- **Accuracy:** The ratio of accurate predictions to the total number of predictions is calculated. This can be measured in a binary hierarchy as:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FN + FP} \tag{7}$$

- **Precision:** The ratio between the correctly expected data and the overall optimistic predicted details. This ensures that a high-precision model is capable of accurately defining much of the expected:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{8}$$

- **Recall:** This metric gives an analysis of the model's sensitivity. That is, the percentage of the positive data that was accurately defined as positive and the positive total data:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{9}$$

- **F1 score**: Using precision and recall, the fourth evaluation metric is calculated as follows:

$$\text{F1\_Score} = 2 \times \frac{\text{Recall} + \text{Precision}}{\text{Recall} \times \text{Precision}} \tag{10}$$

The F1 score is used to demonstrate the model's overall success in relation to both accuracy and recall. The benefit of using the F1 score for assessing a model's overall success is that the F1 score takes into account the distribution of data and the unequal class situation where false positive and false negative are at stake, which is typically the case with all the algorithms.

VII. RESULTS AND DISCUSSION FOR DETECTING ATTACKERS WITH ANN AND DL

The results corresponding to the methods of Sections VI.B and VI.C are shown in Sections VII.A and VII.B, respectively.

A. NEURAL NETWORK

The network was trained for 1000 epochs with data splitting of 70% training part, 15% validation part and 15% testing part, to test for the algorithm's robustness against any bias towards data split. Table 3 shows the results of the evaluation metrics mentioned above.

Fig. 7 shows the performance plot for the data set. The performance plot represents the relationship between the cross-entropy loss which measures the performance of the classification model and the number of epochs. It is clear from the graph that a significant decrease in error between

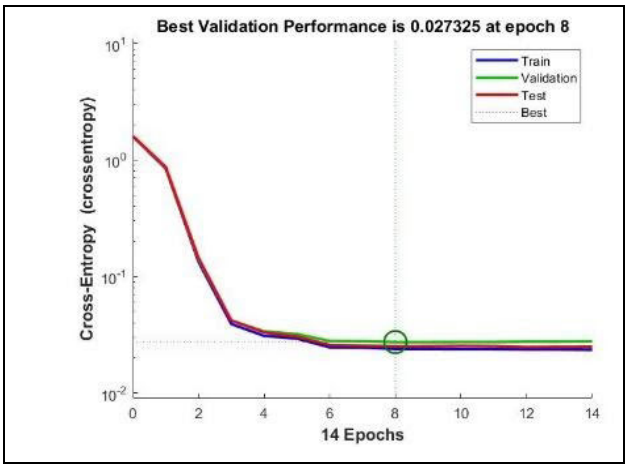

FIGURE 7. The performance plot of the classification test.

the target and the measured output for training, validation and testing partitions of the dataset is noted to almost reach zero. This result is also confirmed by the histogram error graph (Fig. 8), where it shows the testing set bar and the training set bar around the zero-error rate.

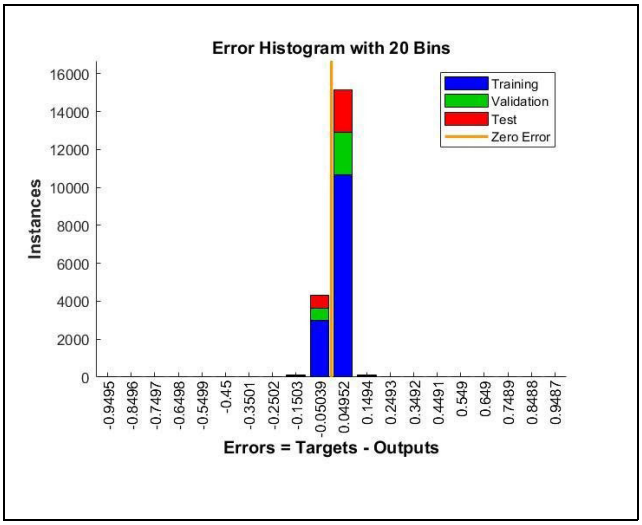

FIGURE 8. The histogram error graph.

B. LONG-SHORT TERM MEMORY

The procedure explores a binary classifier that can differentiate the attackers from non-attackers' samples within threshold scale. After loading the dataset, we prepared it for training by splitting the samples into 70% training and 30% testing, a training set to train the classifier and a testing set to test the accuracy of the classifier on new data. Before training, the neural network shuffles the data at random to ensure that consecutive signals do not have the same labeling. We utilize the bidirectional LSTM layer in our experiment since it looks at the sequence in both forward and backward orientations.

Because the input dataset only has one dimension, we set the input size to one-dimensional sequences. Then, we specify the training options for the classifier such as the number of

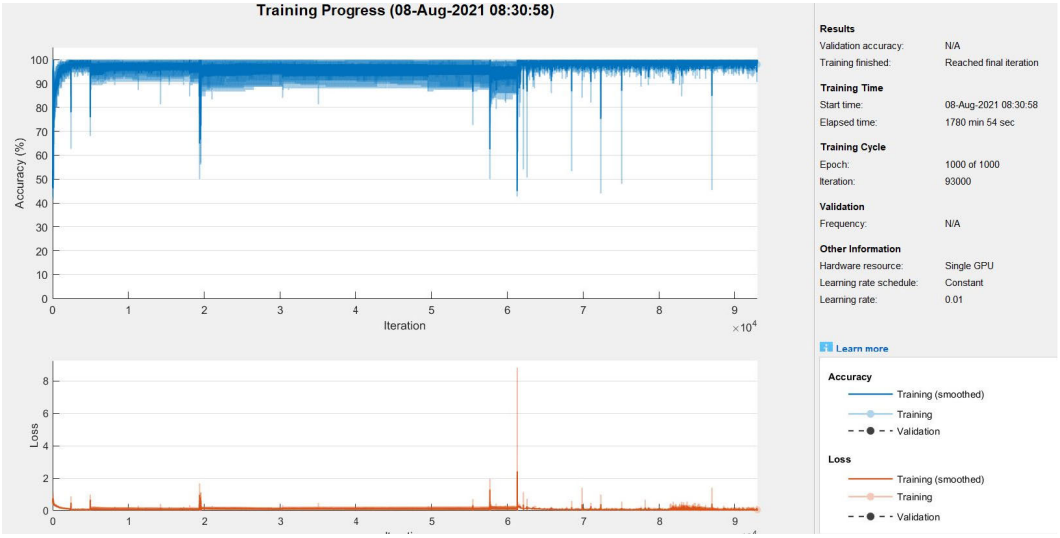

FIGURE 9. The loss and the accuracy.

epochs to be 1000 to allow the network to make 1000 passes through the training data and the 'Initial Learning Rate' of 0.01 helps speed up the training process. The results of our test are shown in Fig. 9, and the confusion matrix is shown in Fig. 10. Table 4 shows the results of the various evaluation metrics.

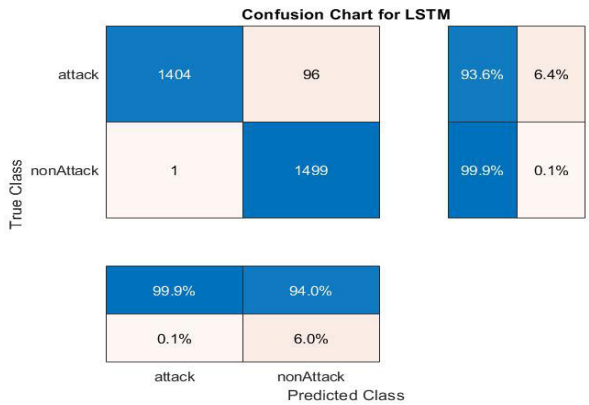

FIGURE 10. The confusion chart for the LSTM.

TABLE 4. The accuracy results for testing model in LSTM.

Precision	Recall	F1-score	Accuracy
99.2%	99%	99.09%	99.1%

The results of the Deep learning model achieved 99.1% accuracy. However, LSTM took an excessively long time of 1780 min 54 sec, compared to less than 2 seconds with the shallow neural network, that achieved almost the same accuracy (98.8%). Additional comparison details are shown in Table 5. The reason behind this time difference is related to the type of our dataset which consists of a single feature

TABLE 5. Illustrats the differences between ANN and LSTM.

Method	Dataset-split	Accuracy	Precision	Recall	F1-score
Shallow NN	70%-30%	99.1%	98.8%	99.3%	99.04%
Deep NN	70%-30%	99.1%	99.2%	99%	99%

for the 20k samples and deep learning mainly performs better with high dimensional datasets. Thus the use of the shallow NN is more suited for the purpose of this paper, which is detecting potential attacks during the transmission of the keys between the transmitter and the receiver.

VIII. CONCLUSION

In this paper, quantum key distribution (QKD) for Internet of things (IoT) was investigated in the presence of attackers attempting to steal the encryption key. Indeed, IoT deployments face significant security challenges due to the limited energy and computational power of IoT devices, especially when attackers might have quantum computing capabilities. Therefore, we proposed an algorithm for detecting an attacker between a transmitter and receiver, at the cost of interrupting the communications to detect the attacker. Afterwards, Artificial Neural Network and Long Short Term Memory techniques were proposed in order to detect the presence of an attacker during QKD without the need to disrupt the key distribution process. The results showed that the proposed techniques can reach 99% accuracy in detecting attackers. A typical use case for implementing the proposed approach was described, namely for securing IoT communications of sensors deployed in railroad networks. This is a suitable scenario because fiber optic cables can be laid out parallel to rail tracks, while coexisting with RF communications used by sensors monitoring the status of the track to report their measurements.

REFERENCES

[1] A. V. Dastjerdi and R. Buyya, "Fog computing: Helping the Internet of Things realize its potential," *Computer*, vol. 49, no. 8, pp. 112–116, Aug. 2016.

[2] B. Hamdaoui, M. Alkalbani, A. Rayes, and N. Zorba, "IoTShare: A blockchain-enabled IoT resource sharing on-demand protocol for smart city situation-awareness applications," *IEEE Internet Things J.*, vol. 7, no. 10, pp. 10548–10561, Oct. 2020.

[3] M. A. Al-Garadi, A. Mohamed, A. K. Al-Ali, X. Du, I. Ali, and M. Guizani, "A survey of machine and deep learning methods for Internet of Things (IoT) security," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 3, pp. 1646–1685, 3rd Quart., 2020.

[4] W. Saad, M. Bennis, and M. Chen, "A vision of 6G wireless systems: Applications, trends, technologies, and open research problems," *IEEE Netw.*, vol. 34, no. 3, pp. 134–142, May/Jun. 2020.

[5] M. Abomhara and G. M. Køien, "Cyber security and the Internet of Things: Vulnerabilities, threats, intruders and attacks," *J. Cyber Secur.*, vol. 4, no. 1, pp. 65–88, 2015.

[6] A. Lohachab, A. Lohachab, and A. Jangra, "A comprehensive survey of prominent cryptographic aspects for securing communication in post-quantum IoT networks," *Internet Things*, vol. 9, Mar. 2020, Art. no. 100174.

[7] T. M. Fernandez-Carames, "From pre-quantum to post-quantum IoT security: A survey on quantum-resistant cryptosystems for the Internet of Things," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 6457–6480, Jul. 2020.

[8] Z. Dawy, W. Saad, A. Ghosh, J. G. Andrews, and E. Yaacoub, "Towards massive machine type cellular communications," *IEEE Wireless Commun.*, vol. 24, no. 1, pp. 120–128, Feb. 2017.

[9] A. Mavromatis, F. Ntavou, E. H. Salas, G. T. Kanellos, R. Nejabati, and D. Simeonidou, "Experimental demonstration of quantum key distribution (QKD) for energy-efficient software-defined Internet of Things," in *Proc. Eur. Conf. Opt. Commun. (ECOC)*, Sep. 2018, pp. 1–3.

[10] H. A. Al-Mohammed and E. Yaacoub, "On the use of quantum communications for securing IoT devices in the 6G era," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, Jun. 2021, pp. 1–6.

[11] M. Alfarhood and J. Cheng, "Deep learning-based recommender systems," *Adv. Intell. Syst. Comput.*, vol. 1232, no. 1, pp. 1–23, 2021, doi: 10.1007/978-981-15-6759-9_1.

[12] H. D. Z. He and Y. Wang, "Wavelength attack recognition based on machine learning optical spectrum analysis for the practical continuous-variable quantum key distribution system," *J. Opt. Soc. Amer. B, Opt. Phys.*, vol. 37, no. 6, pp. 1689–1697, 2020.

[13] Y. Mao, W. Huang, H. Zhong, Y. Wang, H. Qin, Y. Guo, and D. Huang, "Detecting quantum attacks: A machine learning based defense strategy for practical continuous-variable quantum key distribution," *New J. Phys.*, vol. 22, no. 8, Aug. 2020, Art. no. 083073, doi: 10.1088/1367-2630/aba8d4.

[14] L. J. Ba and R. Caruana, "Do deep nets really need to be deep," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 3, Jan. 2014, pp. 2654–2662.

[15] S. J. Nawaz, S. K. Sharma, S. Wyne, M. N. Patwary, and M. Asaduzzaman, "Quantum machine learning for 6G communication networks: State-of-the-art and vision for the future," *IEEE Access*, vol. 7, pp. 46317–46350, 2019, doi: 10.1109/ACCESS.2019.2909490.

[16] M. Sasaki, "Quantum key distribution and its applications," *IEEE Secur. Privacy*, vol. 16, no. 5, pp. 42–48, Sep. 2018.

[17] B. Muruganantham, P. Shamili, S. Ganesh Kumar, and A. Murugan, "Quantum cryptography for secured communication networks," *Int. J. Electr. Comput. Eng.*, vol. 10, no. 1, pp. 407–414, 2020.

[18] K. H. Hummer, "Operation control and signaling system for high-speed lines," in *Proc. 1st Int. Symp. Adv. Train Control*, Denver, CO, USA, Jun. 1991, pp. 114–121.

[19] K. Kumar, "HS—Automatic train control: Concept of system," Urban Transp. High Speed Directorate, Res. Des. Standards Org., India, Technol. Surv. Rep., 2012.

[20] V. Diaz, "Backhauling with fibre," *Fibre Syst.*, no. 6, pp. 33–34, Winter 2015.

[21] A. K. Jain, J. Mao, and K. M. Mohiuddin, "Artificial neural networks: A tutorial," *Computer*, vol. 29, no. 3, pp. 31–44, Mar. 1996.

[22] H. Y. Priyangga and D. Ruliandi, *Application of Pattern Recognition and Classification Using Artificial Neural Network in Geothermal Operation*. Stanford, CA, USA: Stanford Univ., 2018, pp. 1–9.

[23] L. Lazli and M. Boukadoum, "Hidden neural network for complex pattern recognition: A comparison study with multi- neural network based approach," *Int. J. Life Sci. Med. Res.*, vol. 3, no. 6, pp. 234–245, Dec. 2013, doi: 10.5963/lsmr0306003.

[24] M. Şahin and R. Erol, "A comparative study of neural networks and ANFIS for forecasting attendance rate of soccer games," *Math. Comput. Appl.*, vol. 22, no. 4, p. 43, Nov. 2017, doi: 10.3390/mca22040043.

[25] I. N. da Silva, D. H. Spatti, R. A. Flauzino, L. H. B. Liboni, and S. F. dos Reis Alves, "Artificial neural networks: A practical course," in *Artificial Neural Networks*. Cham, Switzerland: Springer, 2017, pp. 21–27.

[26] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, 2006.

[27] R. Mu, "A survey of recommender systems based on deep learning," *IEEE Access*, vol. 6, pp. 69009–69022, 2018.

[28] B. Liu, Q. Zeng, L. Lu, Y. Li, and F. You, "A survey of recommendation systems based on deep learning," *J. Phys., Conf. Ser.*, vol. 1754, no. 1, Feb. 2021, Art. no. 012148, doi: 10.1088/1742-6596/1754/1/012148.

[29] C. Chopra, S. Sinha, S. Jaroli, A. Shukla, and S. Maheshwari, "Recurrent neural networks with non-sequential data to predict hospital readmission of diabetic patients," in *Proc. ACM Int. Conf.*, Oct. 2017, pp. 18–23, doi: 10.1145/3155077.3155081.

[30] Z. C. Lipton, D. C. Kale, C. Elkan, and R. Wetzel, "Learning to diagnose with LSTM recurrent neural networks," in *Proc. 4th Int. Conf. Learn. Represent. (ICLR) Conf. Track*, 2016, pp. 1–18.

[31] M. K. Aditi and E. Poovammal, "Image classification using a hybrid LSTM-CNN deep neural network," *Int. J. Eng. Adv. Technol.*, vol. 8, no. 6, pp. 1342–1348, 2019.

[32] Z. Li, J. Li, Y. Wang, and K. Wang, "A deep learning approach for anomaly detection based on SAE and LSTM in mechanical equipment," *Int. J. Adv. Manuf. Technol.*, vol. 103, nos. 1–4, pp. 499–510, 2019, doi: 10.1007/s00170-019-03557-w.

[33] Y. Tan, C. Hu, K. Zhang, K. Zheng, E. A. Davis, and J. S. Park, "LSTM-based anomaly detection for non-linear dynamical system," *IEEE Access*, vol. 8, pp. 103301–103308, 2020, doi: 10.1109/ACCESS.2020.2999065.

[34] J. J. Q. Yu, A. Y. S. Lam, D. J. Hill, and V. O. K. Li, "Delay aware intelligent transient stability assessment system," *IEEE Access*, vol. 5, pp. 17230–17239, 2017, doi: 10.1109/ACCESS.2017.2746093.

[35] H. Daumé, III, *A Course in Machine Learning*. College Park, MD, USA: UMIACS, Jan. 2017. [Online]. Available: http://ciml.info/ and http://ciml.info/dl/v0_99/ciml-v0_99-ch00.pdf

HASAN ABBAS AL-MOHAMMED received the bachelor's degree in computer engineering from Iraq University College, Basra, Iraq, in 2014, and the master's degree in computing from Qatar University, in June 2021. He has more than ten publications in international journals and conferences. His research interests include quantum radar, quantum computing, quantum communications, and security, in addition to the Internet of Things (IoT) and sensor networks.

AFNAN AL-ALI received the Master of Science degree in computer engineering from the University of Basra, Basra, Iraq. She is currently pursuing the Ph.D. degree with Qatar University. Her research interests include machine learning, AI, computer vision, and object tracking. She is also interested in machine learning for healthcare.

ELIAS YAACOUB (Senior Member, IEEE) received the B.E. degree in electrical engineering from Lebanese University, in 2002, and the M.E. degree in computer and communications engineering and Ph.D. degree in electrical and computer engineering from the American University of Beirut (AUB), in 2005 and 2010, respectively. He worked as a Research Assistant with the American University of Beirut, from 2004 to 2005, and Munich University of Technology, in Spring 2005. From 2005 to 2007, he worked as a Telecommunications Engineer with Dar Al-Handasah, Shair, and Partners. From November 2010 to December 2014, he worked as a Research Scientist/Research and Development Expert with Qatar Mobility Innovations Center (QMIC), where he led the Broadband Wireless Access Technology Team. Afterward, he joined the Strategic Decisions Group (SDG), where he worked as a Consultant, till February 2016. Then, he joined Arab Open University (AOU) as an Associate Professor and a Coordinator of the M.Sc. Program in information security and forensics. From February 2018 to August 2019, he worked as an Independent Researcher/a Consultant and he was also affiliated with AUB as a part-time Faculty Member. He has been an Associate Professor with the Computer Science and Engineering Department, Qatar University, since August 2019. His research interests include wireless communications, resource allocation in wireless networks, intercell interference mitigation techniques, antenna theory, sensor networks, and physical layer security.

UVAIS QIDWAI received the B.S. degree from NED University of Engineering and Technology, in 1994, the M.S. degree from KFUPM, Saudi Arabia, in 1997, and the Ph.D. degree from the University of Massachusetts–Dartmouth, in 2001, all in EE. He taught with the Electrical Engineering and Computer Science Department, Tulane University, New Orleans, as an Assistant Professor. In 2005, he joined the Computer Science and Engineering Department, Qatar University, where he is currently an Associate Professor of computer engineering. He has participated in several government- and industry-funded projects in USA, Saudi Arabia, Qatar, United Arab Emirates, and Pakistan. He has published over 125 articles in reputable journals and conference proceedings. His research interests include signal and image processing, robotics, fuzzy computations, interfacing, expert systems, and intelligent system design.

KHALID ABUALSAUD (Senior Member, IEEE) is currently with the Computer Science and Engineering Department, Qatar University, Qatar. He has more than 25 years of professional experience in information technology. He teaches courses in hardware and software systems. His research interests include health systems, wireless sensors for the IoT applications, cybersecurity, cloud computing, and computer network protocols. His research work has been presented in international conferences and journals. He has participated actively in organizing several IEEE international conferences in Qatar, namely, ICIoT2020, IEEE WCNC'2016, PLM'2015, AICCSA'2014, RelMiCS'2011, and AICCSA'2008. He received several awards from different local and international organizations. He is active in getting research funding from different sources, including Qatar National Research Foundation, the Supreme Committee for Delivery and Legacy (FIFA'2022), and some other organizations in Qatar. He is also a LPI of NPRP 10-1205-160012 Research Project which achieved significant outcomes. He has served as a technical program committee (TPC) member and the chair for various reputable IEEE conferences. Recently, he served as a Guest Editor in Connected Healthcare Special Issue for *IEEE Network*. He is an Associate Editor of *IET Quantum Communication* journal.

STANISŁAW RZEWUSKI received the Bachelor of Engineering degree in applied mechanics, the master's degree in applied mechanics, the master's degree in telecommunications, and the Ph.D. degree from Warsaw University of Technology, in 2004, 2006, 2008, and 2017, respectively.

He has been working in several software development positions, since 2004. He worked at RS Technologies, Poland, where he was involved in aviation electronics production for research rockets and aviators, from 2015 to 2017. He currently works with IS-Wireless, Poland. His research interests include passive radar and wireless networks, in addition to software development and testing.

ADAM FLIZIKOWSKI received the M.Sc. degree from the University of Technology and Life Sciences, Bydgoszcz, Poland, in 2000. He is currently a Research and Development Expert/a System Architect at IS-Wireless, Poland. He has around 18 years of professional experience in information and communication technologies (ICT). He is in charge of IS-Wireless activities in the EuWireless and 5G Essence Research and Development projects. His research interests include QoS in heterogeneous networks, RRM in wireless networks (admission/congestion control), video adaptation, drone-based surveillance, and machine learning. He has more than 65 publications and several patents.