Chapter 1

Boosting Machine Learning Mechanisms in Wireless Mesh Networks Through Quantum Computing

Francesco Vista [1,2], Vittoria Musa [1,2], Giuseppe Piro [1,2], Luigi Alfredo Grieco [1,2], and Gennaro Boggia [1,2]

1.1 Introduction

Nowadays, the ever-increasing demand of data rate and node density, along with low latency and reliability features, makes the introduction of Wireless Mesh Networks (WMNs) a key solution for future wireless communication networks [1, 2]. WMNs, in fact, are self-organised and self-configured networks, where every node is able to autonomously establish and manage its connection to the network in real-time. In detail, a WMN consists of two different types of nodes, named Mesh Routers (MRs) and Mesh Clients (MCs). MRs, as in traditional wireless communication systems, are usually equipped with multiple interfaces to integrate the WMN with internet and various existing wireless networks (e.g., wireless sensor networks, wireless-fidelity (Wi-Fi), and mobile networks). MCs, instead, correspond to typical wireless devices which, differently from MRs, can be mobile and cannot be used as gateways (e.g., laptops, mobile phones, and tablets) [3]. Based on node functionalities, WMNs can be deployed by following three different network architectures. In backbone WMNs, only MRs build the mesh network by creating an infrastructure for clients and providing access to the backbone by leveraging existing wireless interfaces. In client WMNs, instead, also end-users act as relay nodes forwarding incoming packets through the network. To reduce the overall network cost and complexity of the previous architectures, the hybrid WMN considers that each MC can directly communicate with neighbouring MCs or access the mesh network exploiting MRs.

Despite the manifold advantages introduced by the adoption of WMNs in terms of reliability, network installation costs, long-range communications, and large-coverage connectivity, several critical factors negatively affect WMN performance, including network capacity and management issues, scalability, and mobility [3, 4].

[1]Department of Electrical and Information Engineering (DEI), Politecnico di Bari, Bari, Italy
[2]Consorzio Nazionale Interuniversitario per le Telecomunicazioni (CNIT)

Some of these drawbacks can be partially solved through the introduction of novel enabling technologies already investigated by the scientific community. For instance, the network flexibility and capacity can be strongly enhanced by the introduction of single-user or multi-user Multiple-Input/Multiple-Output (MIMO) systems [5, 6]. Moreover, several works exploit unique capabilities of Software-Defined Networking (SDN) paradigm, such as global visibility, real-time programming, and agility, to guarantee optimal network management and further improve the system performance [7, 8]. The Quality of Service (QoS) of the communication system can be also enhanced by the adoption of Intelligent Reflective Surfaces (IRSs) which improve the Signal to Noise Ratio (SNR) both in Line of Sight (LoS) or Non-Line of Sight (NLoS) scenarios by exploiting the environment as a controllable signal reflector [9]. A further improvement in terms of capacity and QoS, while guaranteeing secure and fault-tolerant communications, is provided by the application of Machine Learning (ML) algorithms to solve design and management tasks in WMNs [10]. In the last years, in fact, the scientific community is promoting the adoption of ML techniques to strongly enhance the network adaptability according to real-time conditions also in highly variable scenarios. However, given the continuous growth of involved devices and, in turn, the amount of data to be exchanged and processed in future WMN applications (e.g., wearable devices [11], Vehicular Ad Hoc Networks (VANET) [12], and smart cities [13]), the computational time required by traditional computers to solve ML algorithms is expected to proportionally increase, demanding much efforts for training and inference procedures [14].

Under these premises, the innovation progress triggered by the emerging of Quantum Computing (QC) can be considered as a turning point to counteract this issue. Indeed, QC investigates quantum mechanics principles to develop new types of algorithms able to solve complex problems faster than classical approaches. Accordingly, the adoption of QC may speed up ML techniques, making them suitable also for extreme scenarios (i.e., computationally heavy and real-time applications). The scientific community already proposed the usage of QC-aided ML (i.e., Quantum Machine Learning (QML)) for several challenging applications, such as 6G networks [14], chemistry, and physics [15]. However, at the time of this writing and to the best of authors' knowledge, the application of QML algorithms for WMNs remains an unexplored research topic. To bridge this gap, this work provides a twofold contribution. First, it defines design strategies and new logical entities useful to exploit the potential of QC for WMN intelligence. Specifically, it proposes a centralised and a distributed network architectures according to the quantum computers location. The centralised architecture is supposed to perform QML tasks by exploiting quantum computers already deployed by Tech Giants in their cloud. Traditional computers, instead, are supposed to be spread at the edge of the network to solve simpler ML problems. In this case, a new node is added to the network in order to efficiently allocate computing resources (i.e., traditional and quantum computers) based on the task complexity and the status of computing resources. The distributed architecture, instead, is expected to be feasible only in very far future where quantum computers, equipped with a lower number of quantum bit (qubit), will placed at the edge of the network and communicate by exploiting the quantum Internet. In this case, a creation of a quantum network allows to distribute computing resources more efficiently and also to solve more complex ML problems. Anyway, this approach requires the introduction of new nodes which setup the quantum network by distributing entangled particles between involved quantum computers. Second, this work presents pros and cons of the proposed WMN architectures, pointing out the main issues to be solved and paving the way for future research activities in this promising topic.

The rest of this book chapter is organised as follows. Section 1.2 presents the role of ML for WMNs, highlighting the usage of well-known algorithms to solve typical issues in this context and discussing their main limitations. Section 1.3 describes QC principles and introduces QML algorithms as a possible solution to overcome ML limits. Section 1.4 proposes two network architectures to integrate quantum computers in WMNs and, finally, Section 1.5 draws the conclusions of this work and faces possible future research directions.

1.2 The Role of Machine Learning in WMNs

ML algorithms are a subset of Artificial Intelligence (AI) techniques aiming at improving the performance of a system starting from data and information collected during previous tasks. Unlike optimisation schemes, they are strongly adaptable to environmental conditions, resulting particularly suitable for time-variable use cases, such as WMN-based applications. Typically, ML techniques are classified as supervised, unsupervised, reinforcement, and deep learning. Given the features of these algorithms, they can be used in WMNs to solve different design and management tasks [16]. Fig. 1.1 summarises the main ML algorithms exploited in WMN scenarios.

1.2.1 Supervised Learning

Supervised learning employs a training dataset, composed of input-output pairs, to develop a model that learns over time and computes the corrected output corresponding to new input data. To this end, supervised learning algorithms aim at minimising a loss function in an iterative manner by modifying the hyperparameters of the model. In this context, the most commonly used algorithms for WMNs are Decision Tree (DT), Support Vector Machines (SVMs), and K-Nearest Neighbors (KNN), typically performing classification or regression tasks.

Specifically, DT algorithms exploit a tree-like structure to solve both classification and regression problems. The attributes of the input data are compared with features labelling internal nodes of the tree. Starting from the root node and performing these comparisons, the algorithm traverses the tree until it reaches the leaf nodes which represent the class or the relationship between dependent and independent variables. SVM, instead, is a ML technique commonly used for classification tasks. In this case, the algorithm constructs hyperplanes aiming at maximising the width of the gap between points belonging to different classes in order to increase the classification precision of successive input data. SVM and DT algorithms are employed

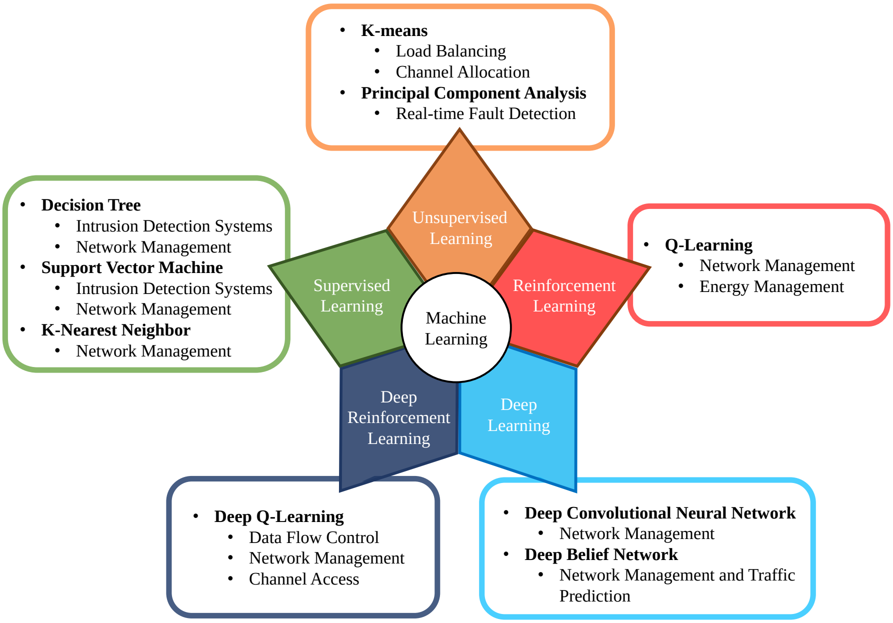

Figure 1.1 Summary of ML techniques and corresponding role in WMNs.

in WMNs to build efficient cross-layer-based [17] and network-layer-based [18] intrusion detection systems. In this case, the model is trained starting from packet delivery ratio, packet arrival interval, and end-to-end delay statistics in order to easily detect anomalous behaviour and remove malicious nodes. These algorithms are also integrated with a threshold that avoids false decisions. An easier algorithm used to perform both classification and regression tasks is the KNN. Here, the input data is classified by considering a similarity concept (i.e., every data point falling near others belongs to the same class). Considering an Internet of Things (IoT) network supported by a software-defined WMN, the presented supervised learning approaches (i.e., DT, SVM, and KNN) are used for optimising the management of the network and perform time granular analysis of the network traffic. The comparison among these learning strategies demonstrated that the KNN algorithm provide the best performance in terms of accuracy [7].

1.2.2 Unsupervised Learning

Unsupervised learning algorithms are used to find unknown patterns from a training dataset containing unlabelled data. In particular, this kind of algorithm analyses the internal structure of the training dataset, thus discovering patterns between data without any external information. The main unsupervised algorithms used in WMNs are K-means and Principal Component Analysis (PCA).

K-means is a clustering algorithm that groups the input unlabelled data in k clusters with an iterative procedure. Specifically, at each iteration, the n observations are grouped in order to minimise the variance intra-cluster and maximise the distance inter-cluster. Here, the distance, usually measured through a Euclidean metric, is computed considering the cluster centres, named centroids. The iterative procedure ends when the algorithm converges. In WMNs, K-means can be used for the load balancing of the network [19] or the channel allocation [10]. In detail, load balancing is performed in order to optimise the resource allocation, increase the overall load of the network, and reduce the congestion at the gateways [19]. Moreover, the K-means clustering can be used to group the MRs efficiently, choosing the cluster head according to the computed centroid [10]. The PCA algorithms, instead, aim at reducing the data dimensionality by describing each data point only with several uncorrelated principal components, while maintaining the highest training-data variance in the first component. Given that the PCA algorithm allows handling high-dimensionality application scenarios, it is particularly suitable for real-time fault detection in high-interference environments, such as WMNs [20].

1.2.3 Reinforcement Learning

The training of Reinforcement Learning (RL) models is an iterative process aiming at choosing the optimal decisions among a set of available actions to maximise cumulative feedback received from the environment, named cumulative reward.

The most known RL algorithm is the Q-learning. Its main feature is the capability to train a model without the knowledge of the environment. Q-learning algorithms, in fact, are based on a Q-value that is updated at each iteration: the optimal action corresponds to the largest cumulative Q-value. In WMNs, RL strategies can be exploited for routing purposes in order to decide the optimal route, among many possible paths, to take from source to destination node. RL fits very well with this kind of problem: the next MR could be chosen, at each iteration, from a set of possible actions in that state. Moreover, Q-learning can be used to avoid critical problems, such as the congestion at the gateway, by dynamically learning an optimal routing scheme that considers several metrics (e.g., loss-ratio, interference, and load at the gateways) [21]. Since classical routing protocols may suffer from excessive energy consumption and do not consider past experience, Q-learning can also optimally enhance the energy balance of the network [22].

1.2.4 Deep Learning

Deep Learning (DL) is a sub-field of ML which involves multiple layers for the processing of input raw data in order to progressively extract higher-level features. It commonly uses an artificial neural network composed of many perceptrons organised in multiple dense hidden layers. To properly train a model, it also needs an initial step useful to tune the hyperparameters starting from a huge amount of data. Specifically, DL algorithms train the model by minimising the loss function over the training dataset and extracting the weights of the final model. The main DL architectures used for WMNs are Deep Convolutional Neural Network (DCNN) and Deep Belief Network (DBN).

DCNN is an example of DL architectures, mostly used in computer vision. In this case, the classification task is performed by filtering the input data using convolution layers in order to extract low-level information. Then, the size of the extracted features is reduced by pooling layers, thus obtaining the output of the fully connected layer (i.e., a vector which contains the result of the classification process). In WMNs, gateways receive traffic information from both MRs and MCs, leading to a higher probability that several nodes become congested. To overcome this issue, DCNN can be used to periodically train a model in order to make optimal routing decisions based on past events [23]. On the other hand, DBN is a class of deep neural network defined as a stack of Restricted Boltzmann Machines (RBMs), which is a two-layer undirected graphical model. Each RBM layer is connected with both the previous and next layers and the nodes alongside any layer are not connected with each other. Since RBMs training process is unsupervised, a DBN ending with a Softmax layer can be used both for classification and clustering of unlabelled data. This makes the DBNs algorithms particularly suitable in WMNs to improve the network management operations in terms of network traffic prediction [24].

1.2.5 Deep Reinforcement Learning

Deep Reinforcement Learning (DRL) combines two sub-fields of ML: RL and DL. To efficiently use RL, in fact, agents must infer a good representation of the environment, thus choosing the action which maximises the reward by following a trial and error strategy. However, if the state spaces or action spaces are too large, this decision can be a complex task that requires more computational time. In this context, DL can help agents to make decisions by learning policies directly from high-dimensional and unstructured input data [25]. The most promising example of DRL for WMNs is the Deep Q-Learning Network (DQN).

DQN combines deep neural networks and Q-learning in order to estimate and maximise the Q-values by considering both states and rewards. It can be employed in WMNs to control the data flow and enhance the throughput. In fact, classical control flow methods suffer from the continuous growth of the number of mesh nodes and the complexity of data applications which make these kinds of scenarios strongly dynamic. DQN, instead, intrinsically has the capability to manage and optimise complex traffic communication flows [26]. DRL algorithms can also be used for optimally planning the network in real-time, thus optimally deploying gateways in the WMN and choosing the network topology [27]. Moreover, DRL can manage the channel access in dynamic spectrum scenarios, where multiple discrete channels are shared by different types of nodes without any a priori knowledge [28].

1.2.6 Open issues in the Application of ML for WMNs

The network management improvement reached with the introduction of aforementioned ML techniques in WMNs is motivating the scientific community to implement these intelligence strategies in real-time applications. However, features of WMNs make them particularly suitable for the management of high dynamic scenarios. In this context, the training dataset must be continuously updated in order to accurately describe the behaviour of considered networks, thus requiring a periodical re-trained of ML models [7]. Moreover, the number and heterogeneity of devices involved in future networks are expected to grow exponentially, thus increasing the number and dimensionality of data to be managed. At the same time, the training time of traditional ML methods strongly depends on the data-space dimension [14]. Accordingly, the benefits obtained by the application of ML algorithms in WMNs may be invalidated by delays related to real-time training procedures, thus requiring the introduction of novel intelligence strategies.

1.3 Quantum Computing: Background and Quantum Machine Learning

QC, a subfield of Quantum Information Science, is a well-known paradigm that is gaining momentum in the last years for its particular features. In fact, it harnesses quantum mechanics principles, completely transforming traditional computing approaches and providing performance enhancements in terms of tasks execution time, accuracy, and computational complexity [29]. For instance, superposition and entanglement principles can be exploited to perform complex tasks that in classic realm would be very challenging. Accordingly, QC emerges as a promising solution to overcome the aforementioned issues for the application of ML in WMNs.

1.3.1 Superposition Principle

The quantum computation is based on the concept of qubit. A qubit is a mathematical representation of a discrete two-level quantum system, where the two computational basis states are commonly denoted as $|0\rangle$ and $|1\rangle$. Physically, for instance, a qubit can be described as the polarisation of a photon, where the two orthogonal basis states are the horizontal and the vertical polarisation of the photon. Differently from classical computers, where bits can assume exactly one binary value at any time (i.e., either 0 or 1), in quantum computers qubits can be in a superposition of two simultaneous values until it is observed. According to the superposition principle, hence, n qubits can encode all the 2^n possible states at once. As a consequence, the power of QC, as well as the information intrinsically kept, grows exponentially with the number of involved qubits [30].

1.3.2 Quantum measurement

In quantum mechanics, it is not possible to establish the state of a qubit by directly observing the quantum state. However, it is allowed to observe the results of the measurements. According to the quantum measurement postulate, after the measurement of the original quantum state, a qubit collapses in either the zero state or the one state (or in one state of the basis states in case of multi-qubit systems). The result of the quantum measurement depends on the amplitude probability associated with each state and any further measurement will give the same result. This deeply impacts the design of the quantum network: a quantum state cannot be transmitted by simply measuring the qubit and sending the result [30].

1.3.3 No-cloning theorem

The no-cloning theorem states that it is not allowed to make a copy of an unknown quantum state. This is a fundamental concept of the Quantum Key Distribution (QKD), since if an eavesdropper tries to *read* the state of a photon, which travels along the path from a sender to a receiver, it will destroy the state of the photon. Furthermore, since it is not possible to store redundant copies of the qubit, new strategies are needed to send a qubit among remote quantum devices, such as quantum entanglement and teleportation [30].

1.3.4 Entanglement

Entanglement is a quantum phenomenon with no equivalent example in the classic world, where two (or more) distant particles share a quantum state. In this case, a measurement performed on one particle affects the outcome of the entangled one. The maximally entangled quantum states are the well-known Bell states [30]:

$$
\begin{aligned}
|\Phi^{\pm}\rangle &= \frac{1}{\sqrt{2}}(|00\rangle \pm |11\rangle) \\
|\Psi^{\pm}\rangle &= \frac{1}{\sqrt{2}}(|01\rangle \pm |10\rangle)
\end{aligned}
$$

The entanglement is a key resource to enable teleportation in quantum networks. Accordingly, two remote quantum computers which want to communicate must share entangled particles. In this context, three different methods can be employed to generate and distribute entanglements [31].

The first method, known as Spontaneus Parametric Down-Conversion (SPDC), generates a pair of entangled photons using their polarisation. A non-linear crystal is hit by a laser beam generating two photons with vertical and horizontal polarisation. These photons, called flying qubit, reach the interested nodes through a quantum channel. Then, at each side, flying qubits are transferred to a computation qubit by using a transducer device to execute quantum operations.

The second method uses optical fiber to connect optical cavities to each side. In particular, entanglement is generated at the sender by exciting the atom with a laser beam, which causes the emission of a photon entangled with the atom. This photon reaches the other node passing through the optical fiber and it is absorbed by the optical cavity. At this point, the atom-photon entanglement is mapped to an atom-atom entanglement.

The latter method also uses optical cavities, but it generates the entanglement at both remote quantum computers by exciting both the atom at the same time. The emitted photons reach a particular device, which is in charge of performing a Bell state measurement in order to map the atom-photon entanglement on both sides into an atom-atom entanglement between the remote quantum computers.

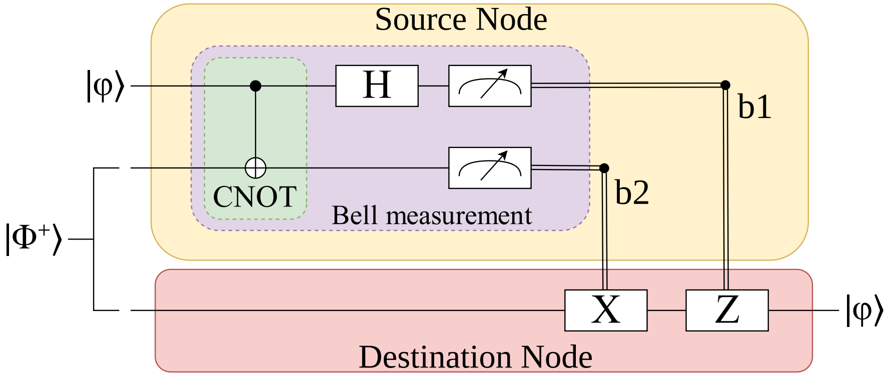

Figure 1.2 Quantum teleportation circuit.

1.3.5 Teleportation

Quantum teleportation allows sending qubits without the transmission of the physical particle that stores the quantum state. In fact, although a qubit can be encoded by the photon polarisation, if it is lost due to attenuation or altered by the environment while it is transferred to a remote quantum computer, the original quantum state cannot be recovered. Hence, quantum teleportation is a workaround procedure for transferring quantum information leveraging classical communication media.

At the basis of quantum teleportation, it is assumed that a specific node in a quantum network can generate and distribute entanglement between the source and destination node. Given the quantum measurement postulate and the no-cloning theorem, the source node has to send the quantum state $|\varphi\rangle$ to a destination node without any a priori knowledge about its state. To this end, the source node must perform some operations, depicted in Fig. 1.2, between the qubit to be sent and its owned part of the entangled particle, i.e., $|\Phi^+\rangle$. In particular, the sender performs a Bell state measurement which consists of a Controlled NOT (CNOT) operation followed by a Hadamard operation and two measurements. The CNOT operation acts on two qubits and performs a bit-flip (i.e., a NOT operation) on the target qubit when the control qubit is $|1\rangle$. Otherwise, the target qubit remains unchanged. The Hadamard operation is applied on the first qubit and creates a superposition of the two basis states. Both the CNOT and Hadamard operations are used to rotate the Bell basis into the computational basis of the two qubits. Finally, the outcomes of the measurement on the two qubits, i.e., $b1$ and $b2$, are sent to the destination node through a classical channel.

At this point, based on the measurement outcomes, the receiver can recover the original quantum state from its entangled particle by either applying X or Z operations, both or none. The X and Z operations correspond to a bit-flip or phase-flip on the qubit, respectively.

It should be noted that the teleportation of a quantum state does not violate the relativity principle, since it requires classical communication. Furthermore, quantum teleportation guarantees a safe state transferring, since even though an attacker can intercept the classical bits, it does not have the destination's entangled particle and thus it cannot recover the original state. It also does not violate the no-cloning theorem, because the Bell state measurement destroys the original qubit as well as the entangled particle held by the sender. Clearly, to perform another qubit teleportation, it is necessary to generate a new entanglement and distribute it between the two communicating nodes.

1.3.6 Quantum Machine Learning

The combination of QC and ML is emerging as a new powerful technique to improve learning algorithms [15]. Specifically, depending on whether the input data and the information processing system are quantum or classical, there are four different approaches to merge QC and ML [32, 33]:

- *Classical-classical approach.* It implements quantum-inspired classical algorithms on classical computers. Here, classical data are processed by classical computers, by employing traditional ML algorithms based on quantum principles theory.
- *Quantum-classical approach.* It consists in employing ML techniques in a QC system. In particular, ML can help quantum computers to learn from data. For instance, ML can be used to analyse measurement data, thus reducing the number of measurements of a quantum state.
- *Classical-quantum approach.* It is commonly known as QML. This approach aims at translating classical ML algorithms into a quantum-compliant language to take advantage of quantum mechanics by running it on quantum computers. The adoption of this approach requires a pre-processing step to convert the classical input data into suitable data for quantum computers. Nowadays, the research community proposes several encoding methods, such as basis encoding and amplitude encoding [32, 34].
- *Quantum-quantum approach.* It aims to develop quantum algorithms to manipulate quantum data. In this approach, it is not required to encode data, as the input is directly the quantum state of the system.

In particular, this work considers the third approach, as in the real world scenario most of the input data are classical. Moreover, since quantum-inspired algorithms are executed on classical computers, the achievable speed-up is not comparable with running it on quantum computers [15].

Nevertheless, QC and, in turn, the application of QML have to face some hardware problems. Quantum states, in fact, are very fragile (i.e., decoherence principle) and suffer from every gate operation, thus inevitably altering computation tasks and limiting quantum computers capabilities. To avoid these problems, the scientific literature proposes two main strategies: first, quantum circuits must be embedded in a specialised large infrastructure with cooling systems able to maintain a near absolute zero temperature [35]; second, the state of qubits must be preserved through the introduction of quantum error correction schemes which spread the information originally belonging to one logical qubit into several physical qubits [29].

However, due to the continuous growth of the number of devices involved in the WMN and, consequently, the amount of exchanged information, QML can help to speed up algorithms used in WMN. In fact, it can improve the computational time, thus getting results faster and also in real-time, as well as increasing the learning capacity and efficiency by discovering more intricate patterns from the input data [36, 37]. In detail, preliminary studies on the performance comparison between QML and ML algorithms demonstrated that the QML is convenient in the case of high-dimensionality input data [38]. Hence, future wireless networks must take into account the possibility to jointly use traditional ML and QML capabilities by supporting the integration of quantum computers.

1.4 Introduction of QML in WMNs: Design principles and Research Challenges

The application of QML methodologies in WMNs can be achieved only with the definition of novel network architectures. In fact, the integration of quantum and traditional computers performing QML and ML algorithms, respectively, requires the introduction of new logical entities embedded with new functionalities. To this end, this Section presents design principles for the realisation of two innovative network architectures, denoted by centralised and distributed approaches, able to combine the benefits provided by traditional and quantum computers deployed either in the cloud or at the edge of the network.

1.4.1 Centralised Architecture

The integration of QML functionalities in future WMNs first requires the introduction of quantum computers in their architectures. Nowadays, some Tech Giants, such as IBM, Google, and Microsoft, have already developed quantum computers with up to a hundred qubits, also envisioning strong improvements in this direction for the next years [35]. Accordingly, a first suitable approach for integrating quantum computers in WMNs is the centralised architecture depicted in Fig. 1.3, where quantum computers deployed by Tech Giants in their cloud are supposed to perform QML algorithms.

The proposed architecture is composed of the access network, the wireless mesh backbone, and the remote cloud:

- The access network includes all the application scenarios sustained by the WMN (e.g., mobile networks, wireless sensor networks, and vehicular networks) and the related network attachment points which provide the connection to the mesh network (e.g., Base Station (BS) and sink node).
- The wireless mesh backbone hosts MRs (with or without gateway capabilities), a Data Aggregator node which stores and transmits dataset for intelligence operations, and traditional computers solving simple and low-dimensionality ML problems. Given the heterogeneity and complexity of the wireless mesh backbone, the traffic flow is managed by an SDN controller, thus avoiding network congestion issues.

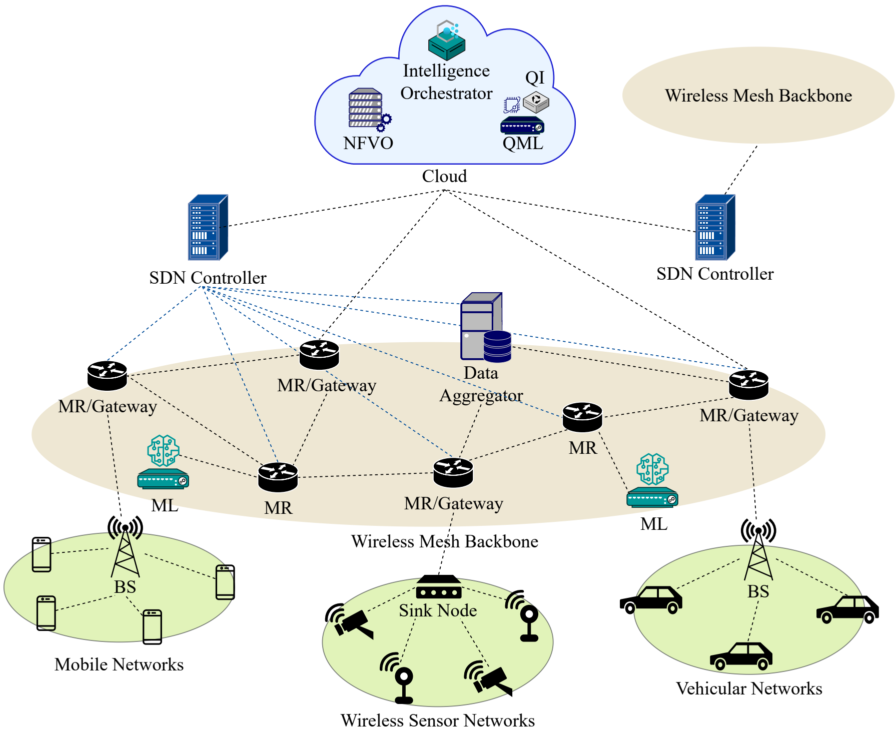

Figure 1.3 QC-aided network intelligence: centralised deployment.

- The remote cloud provides orchestration and high-dimensionality computational capabilities to the overall network. In detail, the Intelligence Orchestrator performs the allocation of computing resources among ML and QML tasks according to their data-dimensionality, while the Network Function Virtualization Orchestrator (NFVO) provides service management functionalities. It is worthwhile to note that quantum computers are equipped with a Quantum Interface (QI) useful for data pre-processing.

1.4.1.1 The Information Exchange in the Centralised Architecture

As illustrated in Fig. 1.4, the information exchange in the centralised architecture can be summarised as in what follows.

- *Phase 1: Dataset Creation.* Each node belonging to the network generates information data to be processed by traditional or quantum computers for the purposes listed in Section 1.2. The collected data strongly depends on the considered node. While end-users (e.g., mobile phones, sensors, and vehicles) acquire data from the surrounding environment, such as channel quality indicators and performance levels of high-level applications, network equipment (e.g., MRs, BSs, sink nodes, SDN controllers, and NFVO) provide information related to

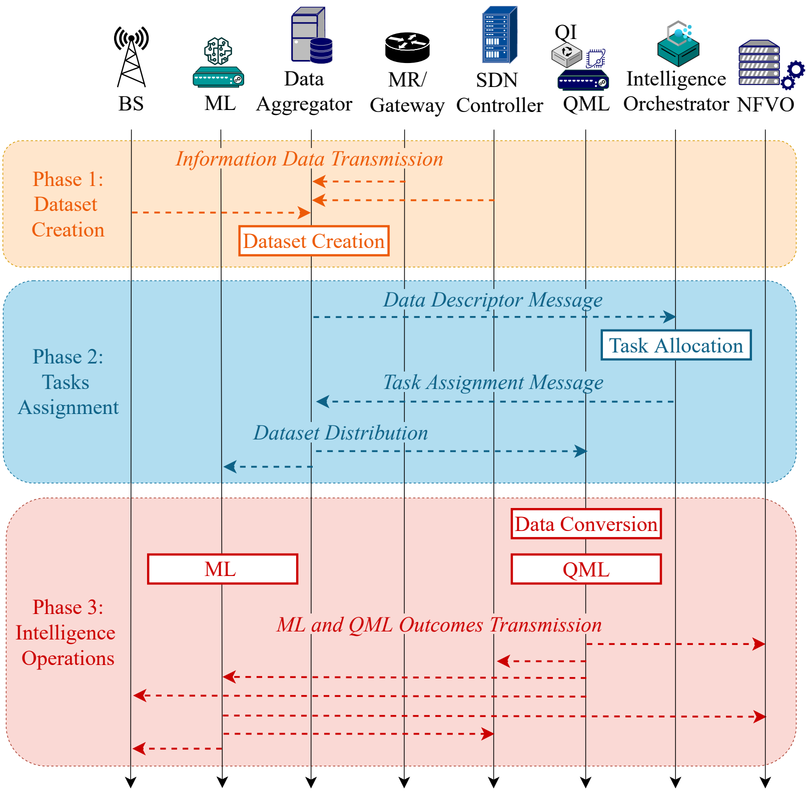

Figure 1.4 Message sequence chart of the centralised architecture.

the network functionalities, such as bandwidth and energy consumption. All the collected data are transmitted to Data Aggregators by means of REST or RESTful communication protocols in order to increase the performance, scalability, simplicity, and reliability of the network. Then, Data Aggregators pre-process incoming data and compare them with existing network information in order to create and/or update datasets useful for intelligence operations.

- *Phase 2: Tasks Assignment.* The Intelligence Orchestrator must assign the generated datasets to computing resources (e.g., traditional or quantum computers). To this end, given the huge amount of data to be exchanged, the Data Aggregators periodically create and transmit to the Intelligence Orchestrator a data descriptor message containing high-level information about the available datasets, such as data format, data size, and statistical variability with respect to previous updates. Starting from this information and considering the status of computing resources, the Intelligence Orchestrator performs the task allocation and sends a task assignment message to the Data Aggregators in order to efficiently transmit the datasets to designed traditional or quantum computers.

- *Phase 3: Intelligence Operations.* Quantum computers in the cloud offer a suitable environment where implementing QML techniques. However, classical data cannot be directly used as input of quantum computers. Accordingly, when QML capabilities are required, a Quantum Interface (QI) logical entity is first used for converting classical data into quantum data, and vice-versa. Without loss of generality, this work considers that these logical entities are directly equipped in the quantum computer. After the data pre-processing, ML and QML operations are performed by traditional and quantum computers, respectively, thus obtaining the corresponding outcomes (e.g., the hyperparameters of the model in case of learning procedures; classification, prediction, or specific actions in other cases). These results are, finally, transmitted to different network equipment for specific purposes, ranging from service management to network optimisation. For instance, the NFVO can exploit these outcomes for optimally managing upper layer services and allocating virtual resources among active applications. SDN controllers, instead, dynamically configure network functionalities (e.g., flow forwarding and load balancing) and solve complex routing problems based on users' mobility and traffic dynamics. Finally, edge nodes and BSs use ML and QML outcomes to update ML models or perform resource scheduling and allocation.

1.4.1.2 Benefits and Research Challenges

The implementation of the proposed centralised architecture in WMNs presents both advantages and disadvantages, thus providing several research directions to the scientific community. On the one hand, at the time of this writing, the number of available quantum computers and qubits is strongly limited by physical and economical constraints, making the centralised approach the first feasible solution to integrate QML in WMNs. Moreover, the centralised architecture presents only one point of failure (i.e., the central nodes in the cloud). As a consequence, it is the easiest architecture to maintain and control. The simplicity of this approach, in fact, allows to efficiently control the computing resources by directly managing the few quantum computers placed in the cloud. On the other hand, the centralised approach is very unstable, since any issue affecting the central nodes in the cloud inevitably causes impairment throughout the network. Furthermore, when QML operations are required, the network must sustain the transmission of a huge amount of data from the Data Aggregator to the remote cloud, thus producing very high bandwidth and energy consumption, along with possible congestion episodes. In addition, the centralised approach presents high communication latency, significantly impairing the benefits provided by the introduction of quantum computers. These open issues introduced several possible research challenges. The transmission of data from Data Aggregators and QIs, in fact, requires the introduction of optimised and energy-aware routing strategies. At the same time, the network scalability can be improved through the optimisation of the number and the position of Data Aggregators based on services details and users' statistics. These problems can be solved by the introduction of the same QC in WMNs. In fact, QML can be used for complex optimisation tasks, like routing problems (i.e., select the optimal path of data-packets) or location strategies.

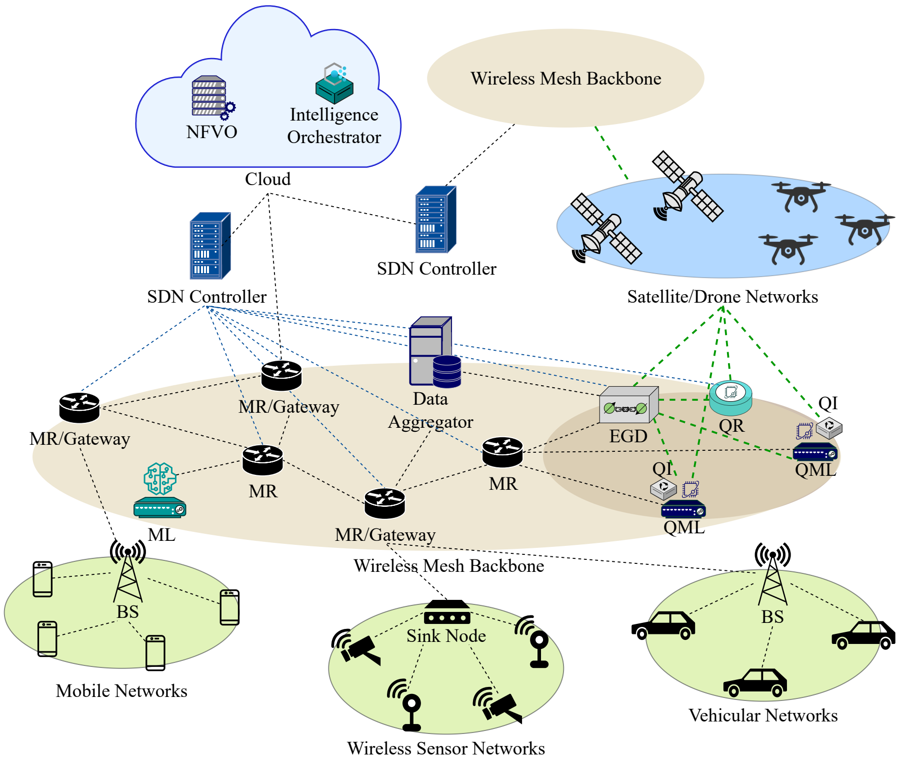

Figure 1.5 QC-aided network intelligence: distributed deployment.

1.4.2 Distributed Architecture

Despite the current limited number of quantum computers and qubits, the growing interest of the scientific community in QC supports the idea of developing also a distributed architecture in a very far future. The decentralisation of QC capabilities entails a higher number of quantum computers, geographically distributed at the edge of the network and equipped with few qubits to mitigate the deployment cost. However, to scale up the number of qubits, it is possible to consider a network of quantum nodes that communicate through the quantum Internet paradigm in order to solve also complex QML tasks [39].

The proposed distributed architecture is illustrated in Fig. 1.5. It requires the introduction of novel logical and physical nodes, along with those presented for the centralised architecture, in order to efficiently sustain the deployment of distributed quantum computers:

- The wireless mesh backbone contains two new nodes, named Entanglement Generator and Distributor (EGD) and Quantum Repeater (QR). The former represents a third party used to generate and distribute entangled particles following, for example, the SPDC method. The latter, instead, allows to distribute

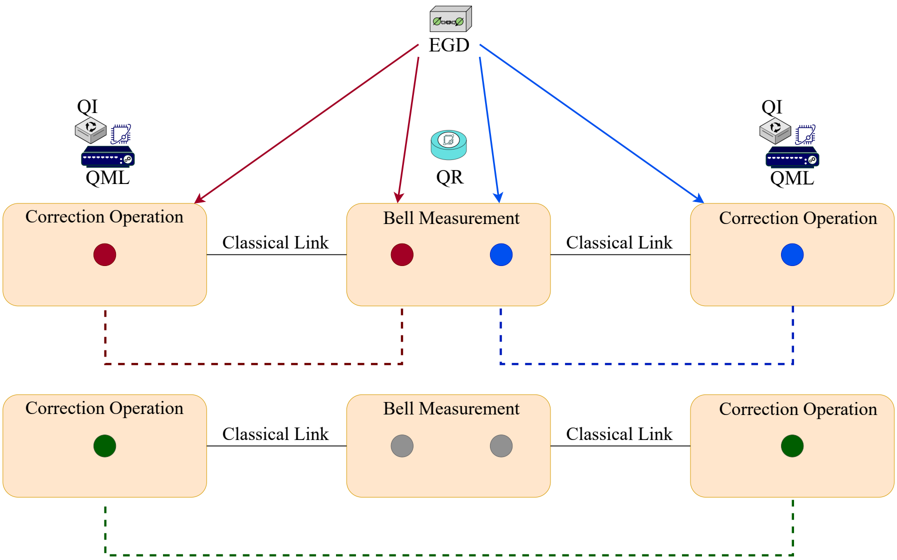

Figure 1.6 Representation of the entanglement swapping procedure.

entangled particles between remote quantum computers. In fact, the reliability of the entanglement distribution is strongly affected by attenuation effects due to the distance between quantum computers [39]. However, since the no-cloning theorem does not allow to simply read and copy qubits, traditional repeaters must be replaced by QRs, which perform the entanglement swapping to establish longer-distance end-to-end entanglements. Fig. 1.6 depicts an example of this procedure involving two distant quantum computers and a QR. In particular, the EGD first creates and delivers a pair of entanglement particles to the involved quantum computers and the QR. Then, the QR performs a Bell state measurement on its particles, thus causing their collapse. The resulting measurement outcomes are transferred to both quantum computers by means of classical channels. Finally, each quantum computer carries out correction operations in order to reconstruct the end-to-end entanglement. Without loss of generality, this procedure can be extended for multiple quantum repeaters scenarios where each QR must be able to receive, process, and transmit both classical and quantum data.

- The entangled particle distribution can be also supported by a satellite or a drone network. Satellites and drones, in fact, can act as QRs for the entanglement distribution of two distant quantum remote computers, completely substituting ground QRs or simply supporting them. On the one hand, the main benefit of using satellite is that photons loss takes place at low levels of the troposphere and the transmission path has no photon absorption [40]. On the other hand, since low-orbital satellites serve specific ground quantum computers only for a limited time, drones can be used as QR, receiving a photon and retransmitting it

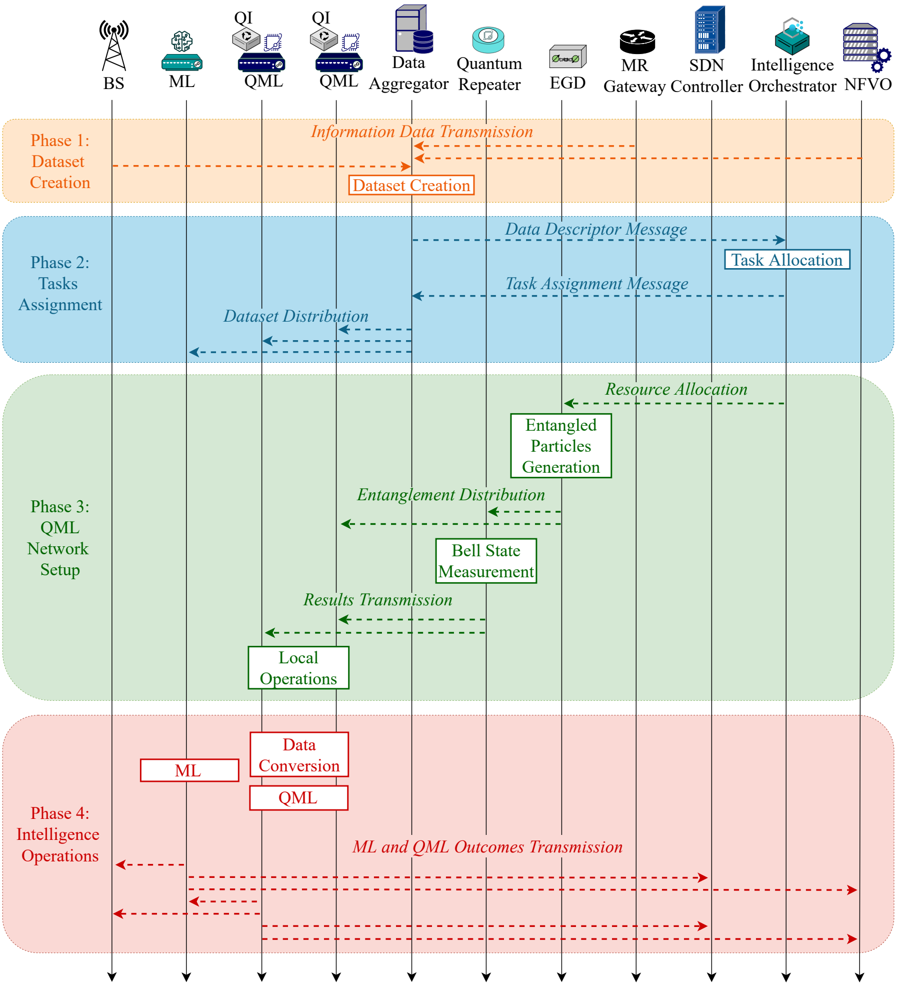

Figure 1.7 Message sequence chart of the distributed architecture.

to the involved quantum remote computer, the next drones, or the next ground QR [41].

It is important to note that the EGD can be either a separated physical node of the network or simply a logical entity equipped by involved QRs (e.g., ground QRs, satellites, or drones). Without loss of generality, this work considers the EGD as a physical ground entity placed in the wireless mesh backbone.

1.4.2.1 The Information Exchange in the Distributed Architecture

The distributed architecture, shown in Fig. 1.5, works as in what follows.

- *Phase 1: Dataset Creation.* As for the centralised architecture, in the first phase of the information exchange procedure, end nodes belonging to different use cases and network equipment deployed in the wireless mesh backbone and in the cloud generate a huge amount of data. This information is, then, transmitted to the Data Aggregator which creates new datasets or updates existing ones.
- *Phase 2: Task Assignment.* Also the second phase, aiming at allocating tasks among computing resources (i.e., traditional and quantum computers), is equivalent to the corresponding phase in the centralised architecture. Here, Data Aggregators transmit metadata of the generated datasets (such as the format or size of data) to the Intelligence Orchestrator, thus avoiding the exchange of an excessive amount of information and, in turn, the congestion of the network. The Intelligence Orchestrator, starting from the aforementioned metadata and from the status of the intelligence network, assigns specific tasks to computing resources and sends a task assignment message to the Data Aggregator. Involved datasets are, finally, delivered to traditional or quantum computers.
- *Phase 3: Network Setup.* Differently from the centralised architecture, in the distributed approach quantum nodes are deployed at the network edge and dynamically grouped in clusters in order to scale up the number of qubits and efficiently solve more complex QML problems. In this case, the Intelligence Orchestrator creates quantum computer networks aiming at grouping computing resources based on the number of available qubits and the distance between them in order to reduce attenuation effects. Since quantum computers belonging to the same cluster share quantum states through the aforementioned teleportation protocol, the third phase of the information exchange envisages setting up the QML network by generating and transmitting entangled particles among involved quantum nodes. They are, then, able to establish a long-distance end-to-end entanglement through the entanglement swapping procedure.
- *Phase 4: Intelligence Operations.* Again, when quantum computers are involved in the computing operation, the received dataset is converted by the QI devices before executing QML algorithms. The outcomes of ML and QML operations are, finally, transmitted to the nodes of the network for different purposes (e.g., the SDN controllers for optimal routing procedures, the NFVO for allocating virtual resources, BSs for optimal resource scheduling).

1.4.2.2 Benefits and Research Challenges

The issues pointed out for the centralised architecture in WMNs can be easily overcome by employing the proposed distributed architecture. In fact, the deployment of multiple quantum computers into the wireless mesh backbone mitigates possible network congestion episodes, improves scalability, and reduces communication latency. On the other side, since quantum Internet is in its fancy, the entanglement distribution and heterogeneity of qubit may represent a first hindrance for the physical implementation of the distributed architecture [39]. In fact, quantum information will be initially transferred to quantum computers belonging to the same cluster and equipped with homogeneous qubits. Then, according to the scientific community long-term vision, the proposed architecture will be practicable when hardware heterogeneity of different quantum computers will be taken into account and entangled particles will be distributed between distant quantum computers. Another issue is related to the simplicity of involved quantum computers in terms of the number of qubits. In this case, complex QML tasks require a high cluster size which drastically increases the number of transferred quantum states and, in turn, the delays and error rates due to the quantum teleportation procedure. Furthermore, the introduction of new physical and logical entities in the distributed architecture entails a more complex system with respect to the centralised approach, thus making the Intelligence Orchestrator a possible point of congestion for the load balancing of computing services. Here, since many quantum computers are involved, it is important to efficiently distribute quantum algorithms among them by jointly minimising the size of the cluster and the user-quantum computer distance. Another research challenge to take into account is the optimal allocation of quantum computers where more computational tasks is expected.

1.5 Conclusions

The need of combining quantum computing and machine learning to fulfil the requirements of future wireless mesh networks envisages the definition of proper network architectures with the introduction of new logical entities. After describing the role of different machine learning algorithms for wireless mesh networks and the main properties of quantum computing, this work introduced the application of quantum machine learning for wireless mesh networks. Specifically, it proposed a centralised and a distributed architecture where quantum machine learning capabilities are placed in the cloud or at the edge, respectively. Design principles and information exchange procedures are deeply discussed for both these architectures, also highlighting their advantages, disadvantages, and possible future research directions.

References

[1] Hossain E, Leung KK. Wireless mesh networks: architectures and protocols. Springer; 2007.

[2] Akyildiz IF, Wang X, Wang W. Wireless mesh networks: a survey. Computer networks. 2005;47(4):445–487.

[3] Akyildiz IF, Wang X. A survey on wireless mesh networks. IEEE Communications magazine. 2005;43(9):S23–S30.

[4] Bano M, Qayyum A, Rais RNB, et al. Soft-Mesh: A Robust Routing Architecture for Hybrid SDN and Wireless Mesh Networks. IEEE Access. 2021;9:87715–87730.

[5] Kusumoto H, Okada H, Kobayashi K, et al. Performance comparison between single-user MIMO and multi-user MIMO in wireless mesh networks. In: The 15th International Symposium on Wireless Personal Multimedia Communications. IEEE; 2012. p. 202–206.

[6] Jaafar W, Ajib W, Tabbane S. The capacity of MIMO-based wireless mesh networks. In: 2007 15th IEEE International Conference on Networks. IEEE; 2007. p. 259–264.

[7] Kumar R, Venkanna U, Tiwari V. A Time Granular Analysis of Software Defined Wireless Mesh Based IoT (SDWM-IoT) Network Traffic Using Supervised Learning. Wireless Personal Communications. 2021;116(3):2083–2109.

[8] Elzain H, Yang W. QoS-aware topology discovery in decentralized software defined wireless mesh network (D-SDWMN) architecture. In: Proceedings of the 2018 2nd international conference on computer science and artificial intelligence; 2018. p. 158–162.

[9] Al-Jarrah M, Alsusa E, Al-Dweik A, et al. Performance Analysis of Wireless Mesh Backhauling Using Intelligent Reflecting Surfaces. IEEE Transactions on Wireless Communications. 2021;20(6):3597–3610.

[10] Karunaratne S, Gacanin H. An overview of machine learning approaches in wireless mesh networks. IEEE Communications Magazine. 2019;57(4):102–108.

[11] Zornoza J, Mujica G, Portilla J, et al. Merging smart wearable devices and wireless mesh networks for collaborative sensing. In: 2017 32nd Conference on Design of Circuits and Integrated Systems (DCIS). IEEE; 2017. p. 1–6.

[12] Nam J, Kim SM, Min SG. Extended wireless mesh network for VANET with geographical routing protocol. In: 11th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM 2015); 2015. p. 1–6.

[13] Lee HC, Ke KH. Monitoring of large-area IoT sensors using a LoRa wireless mesh network system: Design and evaluation. IEEE Transactions on Instrumentation and Measurement. 2018;67(9):2177–2187.

[14] Nawaz SJ, Sharma SK, Wyne S, et al. Quantum Machine Learning for 6G Communication Networks: State-of-the-Art and Vision for the Future. IEEE Access. 2019;7:46317–46350.

[15] Khan TM, Robles-Kelly A. Machine Learning: Quantum vs Classical. IEEE Access. 2020;8:219275–219294.

[16] Zhang C, Patras P, Haddadi H. Deep Learning in Mobile and Wireless Networking: A Survey. IEEE Communications Surveys & Tutorials. 2019;21(3):2224–2287.

[17] Wang X, Wong JS, Stanley F, et al. Cross-layer based anomaly detection in wireless mesh networks. In: 2009 ninth annual international symposium on applications and the internet. IEEE; 2009. p. 9–15.

[18] Shams EA, Rizaner A. A novel support vector machine based intrusion detection system for mobile ad hoc networks. Wireless Networks. 2018;24(5):1821–1829.

[19] Das B, Roy AK, Khan AK, et al. A new approach for gateway-level load balancing of wmns through k-means clustering. In: 2014 International Conference on Computational Intelligence and Communication Networks. IEEE; 2014. p. 515–519.

[20] Zaidi ZR, Hakami S, Landfeldt B, et al. Real-time detection of traffic anomalies in wireless mesh networks. Wireless Networks. 2010;16(6):1675–1689.

[21] Boushaba M, Hafid A, Belbekkouche A, et al. Reinforcement learning based routing in wireless mesh networks. Wireless Networks. 2013;19(8):2079–2091.

[22] Yin M, Chen J, Duan X, et al. Qebr: Q-learning based routing protocol for energy balance in wireless mesh networks. In: 2018 IEEE 4th International Conference on Computer and Communications (ICCC). IEEE; 2018. p. 280–284.

[23] Tang F, Mao B, Fadlullah ZM, et al. On removing routing protocol from future wireless networks: A real-time deep learning approach for intelligent traffic control. IEEE Wireless Communications. 2017;25(1):154–160.

[24] Nie L, Jiang D, Yu S, et al. Network traffic prediction based on deep belief network in wireless mesh backbone networks. In: 2017 IEEE Wireless Communications and Networking Conference (WCNC). IEEE; 2017. p. 1–5.

[25] Mnih V, Kavukcuoglu K, Silver D, et al. Human-level control through deep reinforcement learning. nature. 2015;518(7540):529–533.

[26] Liu Q, Cheng L, Jia AL, et al. Deep Reinforcement Learning for Communication Flow Control in Wireless Mesh Networks. IEEE Network. 2021;35(2):112–119.

[27] Yin C, Yang R, Zou X, et al. Research on topology planning for wireless mesh networks based on deep reinforcement learning. In: 2020 2nd International Conference on Computer Communication and the Internet (ICCCI). IEEE; 2020. p. 6–11.

[28] Xu Y, Yu J, Headley WC, et al. Deep reinforcement learning for dynamic spectrum access in wireless networks. In: MILCOM 2018-2018 IEEE Military Communications Conference (MILCOM). IEEE; 2018. p. 207–212.

[29] Gyongyosi L, Imre S. A Survey on quantum computing technology. Computer Science Review. 2019;31:51 – 71.

[30] Nielsen MA, Chuang IL. Quantum Computation and Quantum Information: 10th Anniversary Edition. Cambridge University Press; 2010.

[31] Cacciapuoti AS, Caleffi M, Van Meter R, et al. When entanglement meets classical communications: Quantum teleportation for the quantum internet. IEEE Transactions on Communications. 2020;68(6):3808–3833.

[32] Sergioli G. Quantum and Quantum-Like Machine Learning: A Note on Differences and Similarities. Soft Computing. 2020;24(14):10247–10255.

[33] Schuld M, Petruccione F. Supervised learning with quantum computers. vol. 17. Springer; 2018.

[34] Sierra-Sosa D, Telahun M, Elmaghraby A. TensorFlow Quantum: Impacts of Quantum State Preparation on Quantum Machine Learning Performance. IEEE Access. 2020;8:215246–215255.

[35] O'Quinn W, Mao S. Quantum Machine Learning: Recent Advances and Outlook. IEEE Wireless Commununications. 2020;27(3):126–131.

[36] Phillipson F. Quantum Machine Learning: Benefits and Practical Examples. In: QANSWER; 2020. p. 51–56.

[37] Bhat JR, AlQahtani SA. 6G Ecosystem: Current Status and Future Perspective. IEEE Access. 2021;9:43134–43167.
[38] Fastovets D, Bogdanov YI, Bantysh B, et al. Machine learning methods in quantum computing theory. In: International Conference on Micro-and Nano-Electronics 2018. vol. 11022. International Society for Optics and Photonics; 2019. p. 110222S.
[39] Caleffi M, Chandra D, Cuomo D, et al. The Rise of the Quantum Internet. Computer. 2020;53(6):67–72.
[40] Yin J, Cao Y, Li YH, et al. Satellite-based entanglement distribution over 1200 kilometers. Science. 2017;356(6343):1140–1144.
[41] Liu HY, Tian XH, Gu C, et al. Drone-based entanglement distribution towards mobile quantum networks. National Science Review. 2020;7(5):921–928.