Received 21 May 2024; accepted 5 June 2024. Date of publication 11 June 2024; date of current version 15 July 2024.

Digital Object Identifier 10.1109/OJCOMS.2024.3412963

Enhancing Healthcare Systems With Deep Reinforcement Learning: Insights Into D2D Communications and Remote Monitoring

ZINA CHKIRBENE[1], RIDHA HAMILA[1] (Senior Member, IEEE), DEVRIM UNAL[2] (Senior Member, IEEE), MONCEF GABBOUJ[3] (Fellow, IEEE), AND MOUNIR HAMDI[4] (Fellow, IEEE)

[1]Electrical Engineering, Qatar University, Doha, Qatar

[2]KINDI Center for Computing Research, College of Engineering, Qatar University, Doha, Qatar

[3]Faculty of Information Technology and Communication Sciences, Tampere University, 33100 Tampere, Finland

[4]College of Science and Engineering, Hamad Bin Khalifa University, Doha, Qatar

CORRESPONDING AUTHOR: Z. CHKIRBENE (e-mail: zina.chk@gmail.com)

This work was supported in part by the Qatar University Internal under Grant IRCC-2020-001. Open Access funding provided by the Qatar National Library. The statements made herein are solely the responsibility of the author[s].

ABSTRACT The traditional healthcare system is increasingly challenged by its dependence on in-person consultations and manual monitoring, struggling with issues of scalability, the immediacy of care, and efficient resource allocation. As the global population ages and chronic conditions proliferate, the demand for healthcare systems capable of delivering efficient and remote care is becoming more pressing. In this context, Deep Reinforcement Learning (DRL) emerges as a technological advancement that improves the healthcare by enabling smart, adaptive, and real-time decision-making processes. Existing DRL applications in resource allocation, however, face significant challenges. They often lack the adaptability required to respond to the dynamic and complex nature of healthcare environments, struggle with optimizing latency, and fail to address specific node capacity constraints key factors that impacts the effectiveness of healthcare applications. Addressing these challenges, this paper introduces the Deep Reinforcement Learning for Live Video Transmission (DRL-LVT) framework. This new technique optimizes video resource allocation in Device-to-Device (D2D) networks within healthcare settings. By formulating the video resource allocation challenge as a multi-objective optimization problem, the framework aims to minimize network delays while respecting node capacity limitations. The core of DRL-LVT is its novel algorithm that leverages Deep Reinforcement Learning (DRL) to dynamically adapt to changing environmental conditions, facilitating real-time decisions that consider node capacities, latency, and the overall network dynamics. We evaluate the performance of our proposed model and benchmark it against existing state-of-the-art techniques. Our results demonstrate significant improvements in efficiency, reliability, and adaptability, making the DRL-LVT framework a robust solution for real-time remote patient monitoring in smart healthcare systems.

INDEX TERMS Smart healthcare system, RPM, video live streaming, deep reinforcement learning, node capacities.

I. INTRODUCTION

THE ADVENT of advanced video streaming technologies and healthcare informatics has paved the way for innovative solutions in remote patient monitoring (RPM). Traditional RPM systems have primarily focused on data collection and elementary analytics, often overlooking the complexities associated with video streaming. In contrast, Real-Time Remote Patient Monitoring (RL-RPM) integrates real-time video streaming to offer a comprehensive view of a patient's condition. The RL-RPM system can reduce the medical risks, and enable exchanging information between different people in different places, which is extremely important especially in some regions like rural areas that might lack specialized medical personnel. In particular, RL-RPM enables accurate evaluations of conditions like dermatological disorders and high-risk pregnancies, enhances postoperative care, and even holds potential in mental health consultations. By offering immediate visual context, live video streaming significantly improves communication, diagnosis, and treatment in remote healthcare, bridging the gap between remote monitoring and traditional in-person care.

Integrating video streaming with healthcare informatics in RL-RPM systems poses several challenges, particularly in ensuring high-quality video streaming, especially during peak usage times. One major challenge in RL-RPM is maintaining the Quality-of-Experience (QoE), particularly in comparison to video-on-demand (VOD) services. Live streaming QoE generally requires a higher bitrate[1] and experiences less rebuffering[2] compared to VOD, as highlighted in [1]. Additionally, latency and network delay are critically important in a live video environment to facilitate real-time interactions among users and manage delays effectively in dynamic network settings. One of the promising advancements in video streaming is Device-to-Device (D2D) communication. D2D allows for direct data transfer between devices without relying on a central network infrastructure. These results in reduced latency and improved resource utilization, making it particularly beneficial for applications that require real-time data exchange, such as healthcare monitoring systems. Several studies, including [2] and [3], have investigated the use of D2D communication for video streaming. These solutions primarily concentrate on meeting delay requirements but often overlook factors like the status of transmission links and the capacities of participating nodes, which are critical in maintaining network connectivity. The key goal in managing network nodes is to select appropriate nodes for relaying video content, considering their capacity thresholds, and at the same time, aiming to minimize transmission delays [4]. In [2], [3], the use of D2D for video streaming is explored, focusing primarily on delay requirements, they often overlook crucial factors such as the status of transmission links and the capacities of participating nodes. These elements are vital for maintaining network connectivity, highlighting the need to select nodes for video content relay based on capacity thresholds while minimizing transmission delays. Effective control of nodes in a D2D network is essential, as it contributes to the physical layer optimizations that are vital for the success of any advanced RL-RPM system. D2D connectivity reduces latency and increases reliability, essential for real-time medical scenarios like surgery monitoring and emergency care. It efficiently manages bandwidth, allowing for the streaming of high-resolution images and live videos directly between devices. This is crucial for robust network performance, emphasizing the importance of selecting nodes based on capacity and minimizing delays. Effective node management in D2D networks enhances remote patient monitoring systems by ensuring fast and reliable healthcare service delivery, where speed and data integrity are critical.

Another significant development is DRL, a subset of machine learning that excels in making intelligent, real-time decisions based on environmental feed back. DRL algorithms can dynamically allocate resources and adapt to changing conditions, making them ideal for complex systems where multiple variables need to be optimized simultaneously. DRL combines Reinforcement Learning with Deep Learning techniques to solve challenging complex decision-making problems [5]. Within the broad spectrum of DRL applications, numerous studies have aimed to refine network performance, focusing on enhancing throughput, conserving energy, and minimizing latency. In [6], the authors focused on optimizing resource allocation in cloud computing environments using DRL. In [7], the authors used DRL for adaptive resource allocation in IoT network. The paper studied an optimization problem within mixed action spaces, combining both discrete and continuous elements. In [8], DRL is used to allocate spectrum resources in cognitive radio networks. In the domain of dense heterogeneous networks (HetNets) over 5G, research highlighted in [9] employs DRL to navigate the network selection challenge, striving to enhance medical data delivery within smart health systems. This model seeks to bolster energy consumption and latency, and satisfy a spectrum of Quality of Service (QoS) requirements. Similarly, the emergence of a novel Healthcare Internet of Things (H-IoT) system [10], fortified by permissioned blockchain and DRL, targets the acute challenges of security [11] and limited energy capacity, exacerbated by the COVID-19 pandemic. However, the scalability of blockchain implementations and the computational overhead intertwined with DRL and blockchain synergy pose considerable challenges, particularly for resource-constrained IoT devices. Despite the importance of effective node control in DRL and D2D networks for the success of advanced RL-RPM systems, existing solutions may not fully address the dynamic nature of healthcare environments. While DRL offers promising avenues for intelligent, real-time decision-making in complex systems [5], applications in cloud computing [6], IoT networks [7], and cognitive radio networks [8] have highlighted the challenge of balancing computational overhead with the scalability of solutions, especially in resource-constrained settings.

In response to these challenges, this paper proposes a new technique, the Deep Reinforcement Learning for Live Video Transmission (DRL-LVT) framework for the healthcare system. The DRL-LVT framework leverages the strengths of DRL to dynamically adapt resource allocation in real-time, ensuring optimal video streaming quality within

[1]Bitrate is associated with video definition; a higher bitrate indicates that the video can be played in higher definition.

[2]Rebuffering refers to the instances when a video pauses for buffering; less rebuffering allows for smoother video playback.

D2D networks. Unlike previous approaches, the DRL-LVT framework takes into account not only the delay requirements but also the critical aspects of node capacity and link status, ensuring a robust and reliable solution for remote patient monitoring. Furthermore, the DRL-LVT framework incorporates a predictive mechanism that intelligently anticipates network variations, thereby maintaining consistent video quality even in fluctuating network conditions. Additionally, the framework incentivizes network participation by offering benefits to D2D communications, such as accelerated access to localized services and data, alongside reduced service costs, promoting wider technology adoption. The followings are the main contributions proposed in this paper:

- Introducing a novel framework, termed DRL-LVT (Deep Reinforcement Learning for Live Video Transmission), specifically designed to optimize real-time video resource allocation in D2D networks for healthcare applications. The proposed model employs a hierarchical decision-making algorithm that allows for control over resource allocation, thereby enhancing the system's overall performance metrics.
- Formulating the video resources allocation problem as an optimization problem that aims to minimize the network delay while respecting the nodes' capacities constraints.
- Proposing a new algorithm to dynamically adapt to varying network conditions, making real-time decisions based on node capacities, latency, and network dynamics. The algorithm learns to solves the formulated allocation problem while adapting to different environment variations.
- Defining the reward function and the set of states, and actions and integrating the deep reinforcement learning model with hierarchical value decisions. Basically, the DRL agent makes discrete actions, each of them controls a set of sub-actions related to the node's link activation.
- Evaluating the performance of the proposed model with various simulations and comparing it with the state of the art techniques.

The rest of this paper is organized as follows.: Section II delves into related work in the field. Section III introduces the proposed system model and the formulation of the problem. Detailed explanations of the proposed scheme's functionality are provided in Sections IV and V. Section VI is dedicated to evaluating the performance of the DRL model in comparison to contemporary techniques. The significance and implications of the proposed model are discussed in Section VII. The paper concludes with Section VIII, summarizing the key findings and contributions.

II. RELATED WORKS

In this section, various solutions for RPM and video streaming techniques are presented.

A. REMOTE MONITORING OF PATIENTS

In recent years, the concept of smart healthcare systems has gained considerable attention, leading to a variety of studies that focus on real-time monitoring for chronic diseases. For instance, the work presented in [12] explores a real-time monitoring system specifically designed for chronic disease management. This system employs body sensor networks to monitor blood pressure, aiming to provide timely interventions and prevent complications. Another study [13] introduces a wireless ECG monitoring system with the objective of reducing data transmission time, thereby enhancing the efficiency of remote healthcare services. Furthermore, research in [14] discusses the use of smartphones for diabetes management, while [15] presents a real-time online assessment and mobility monitoring framework that employs sensor-based infrastructure. In the realm of IoT, the study in [16] proposes an IoT-enabled framework that collects various medical data, including electrocardiograms (ECG), through mobile devices and sensors. This data is then securely transmitted to the cloud, allowing healthcare professionals seamless access for timely interventions. Another noteworthy contribution focuses on hospital resource management by monitoring patients remotely at home, thereby automating the data collection and storage process [17].

While these studies have significantly advanced the field of remote patient monitoring, they often do not address the unique challenges posed by real-time video streaming. This proves the need for integrated solutions that can adapt to the dynamic requirements of real-time video-based monitoring.

B. D2D TECHNIQUE FOR VIDEO TRANSMISSION

D2D communication has emerged as a promising technique for efficient video transmission, particularly in real-time applications. Various models have been proposed to leverage D2D for enhancing video quality and reducing latency. For instance, edge caching is utilized to mitigate congestion and delay in video content transmission [18]. While these models are effective for stored video content, they often fall short in the context of live streaming, introducing issues such as higher throughput, delay, and jitter. The study in [18] focuses on resource utilization in 5G networks but limits its scope to single-user, single-video scenarios, neglecting multi-user, multi-video contexts. Another work [19] prioritizes links for video transmission but fails to consider video quality-aware mechanisms, leaving the suitability of such techniques for on-demand D2D video streaming an open question. In [20], D2D multicast communications for live streaming video are explored. The authors employ frame priority (FP) encoding to improve user Quality of Service (QoS). This ensures that valuable frames for decoding are re-transmitted, enhancing the overall video quality. In [21], an innovative method is introduced, focusing on a blockchain-based model for video streaming. This approach enhances collaboration between content creators and transcoders by incorporating a block size adaptation strategy. Additionally, the well-known Dijkstra's algorithm, as referenced in [22], addresses the shortest-path problem in directed graphs with positive weights.

Despite these advancements, defining an optimal solution for video transmission remains a challenging problem. Inappropriate resource allocation can significantly impact network spectral efficiency, highlighting the need for more robust and adaptable solutions.

C. SMART ALLOCATION TECHNIQUES FOR VIDEO TRANSMISSION

DRL has been widely applied in resource management tasks like power allocation, channel selection, and spectrum access, as highlighted in various studies [23]. Its effectiveness in video streaming and network choice has also been documented in numerous research works [24], [25], [26]. For instance, in [24], the authors proposed an Adaptive Bitrate (ABR) system using Reinforcement Learning (RL), where a neural network model is trained to adjust video chunk bitrates based on data observed on the client side. Similarly, another research [25] introduces a Video Quality Aware Rate Control (QARC) system. This system uses a neural network to enhance transmission quality by adjusting the sending rate and reducing latency, based on historical network conditions. In research [26], a DRL-based approach is used to predict user viewing directions in panoramic video streams, utilizing Long Short-Term Memory (LSTM) algorithms. While these models primarily focus on predicting video resolution and bitrate, they often do not address other Quality of Service (QoS) aspects, such as block error rate, jitter, delay, and latency, influenced by the physical layer. In [27], a comprehensive strategy is presented for enhancing federated learning (FL) within the context of the Industrial Internet of Things (IIoT). This approach focuses on strategic device selection and resource allocation to foster an efficient collaborative FL architecture. In [28], the authors addressed the challenge of real-time deep neural network (DNN) inference within the constrained resources of IIoT networks by introducing an end-edge-cloud orchestration architecture. This setup enables dynamic placement of pre-trained DNN models from the cloud to the edge and end devices for efficient inference. DetFed [29] is applied to federated learning to the industrial IoT, with the help of 6G and Time-sensitive Networks (TSN). Specifically, the complexity of real-time problem-solving, exacerbated by the implicit nature of the optimization's objective function and the temporal correlation of available time slots, presents significant challenges. This complexity can lead to difficulties in achieving the optimal balance between learning accuracy and network performance constraints, such as delay, jitter, and packet loss, especially in highly dynamic or unpredictable network environments.

The exploration into a continuous bitrate and latency control model for live video streaming, utilizing Deep Deterministic Policy Gradient (DDPG) for nuanced control, marks another notable advancement [30]. This model's ability to enhance the Quality of Experience (QoE) across diverse network conditions underscores DRL's potential. Nevertheless, the need for broader validation, concerns over scalability and computational efficiency, and the intricacies of practical deployment within existing infrastructures suggest avenues for further development to elevate the model's robustness and efficacy in streaming applications. DeepPower [31] introduces a DRL-based power management solution for latency-critical (LC) applications in data centers, employing a hierarchical control mechanism for adaptive power management. This novel approach promises substantial power savings and reduced request timeouts, yet it confronts challenges in real-world implementation, including the complexity of the hierarchical control structure and the responsiveness of the system to sudden workload changes. These issues underscore the necessity for further optimization to enhance DeepPower's efficiency and applicability across data center environments. These studies, typically operate within relatively stable or controlled network settings and employ DRL specifically designed for these targeted aims. While successful for their specified purposes, these methods often do not fully address the complex and dynamic challenges prevalent in healthcare settings, where the need for dependable, low-latency communication is critical, especially for live video streams in remote patient monitoring. Another study [32] presents a DRL-based model for scheduling links in wireless networks to address interference, employing Deep Neural Networks (DNN) to understand and adapt to network dynamics. Additionally, research [33] applies DRL in Cognitive Radio (CR) networks for network selection. The DRL agent in this case evaluates performance metrics like collision probability, blocking, dropout rates, and throughput to make decisions. However, the challenge in CR networks is managing the limited and fluctuating resources for various video applications. In contrast to existing works, this paper introduces a novel DRL framework specifically designed for real-time remote patient monitoring. Our centralized DRL agent in the core network has access to all node data and information, which empowers it to perform optimized network performance through smart resource allocation. This unique setup enables the agent to efficiently manage network resources, ensuring that each node operates at optimal parameters and maintains consistent communication standards critical for healthcare monitoring.

III. SYSTEM MODEL AND PROBLEM FORMULATION

This section delineates the proposed system model followed by the formulation of the optimization problem aimed at enhancing video streaming for real-time remote monitoring in healthcare systems.

A. SYSTEM MODEL

Consider a network denoted by $\mathfrak{N}$, comprising N participants, which includes healthcare providers such as doctors and nurses, as well as patients, all interconnected via a D2D communication system. This network is specifically optimized for the unique requirements of medical consultations

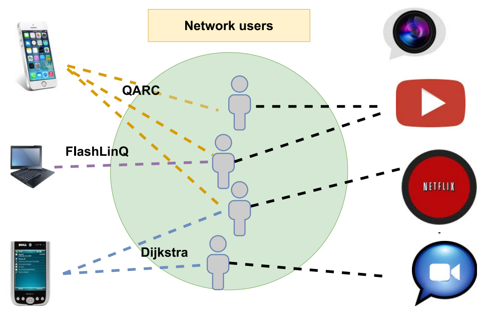

FIGURE 1. Video live streaming system under study.

and remote patient monitoring. Participants within $\mathfrak{N}$ are equipped with devices that possess defined download and upload capacities. These capacities are critical for the efficient transmission and reception of medical data. Utilizing advanced 5G technologies, the network ensures robust and high-speed data communication. Additionally, the network's dynamic topology is intelligently designed to adapt to the varying conditions and locations of the participants.

The transmission delay between any two users, U_i and U_j, in $\mathfrak{N}$ is represented by $D_{i,j}$. For participants not directly connected, $D_{i,j}$ is considered infinite, highlighting the importance of establishing efficient relay paths:

$$D_{i,j} = \infty, \ \forall \ U_i, U_j; \ \ U_i \nleftrightarrow U_j. \tag{1}$$

In this cooperative streaming model, each participant U_i may request and forward video chunks, aiming to minimize the overall streaming delay of the network. The status of the current video chunk v at time t at user U_i is denoted by $H_{U_i}^{v,t}$. If user U_1 successfully forwards the latest chunk of video v at time t to U_2, the decision variable $X_{U_1,U_2}^{v,t}$ is set to one; otherwise, it remains zero:

$$H_{U_i}^{v,t} = t, \ \forall \ U_i \in \mathfrak{S}_v. \tag{2}$$

For any user U_i not in $\mathfrak{S}_v$, the latest chunk of video v at time t is the most recent one received, determined by:

$$H_{U_i}^{v,t} = \max_{U_j \in \mathfrak{N}} \left(X_{U_j,U_i}^{v,t-1} \times H_{U_j}^{v,t-1} \right), \ \forall \ U_i \notin \mathfrak{S}_v. \tag{3}$$

The delay in transmitting a chunk from U_i to U_j is crucial, especially in healthcare settings where information timeliness can directly impact patient outcomes. Thus, the cumulative delay to U_j for video v at time t is computed as:

$$D_j^{v,t} = \sum_{U_i \in \mathfrak{N}} X_{i,j}^{v,t} \times (D_i^{v,t-1} + D_{i,j}). \tag{4}$$

with the conditions for $D_j^{v,t}$ further detailed by:

$$D_j^{v,t} = \begin{cases} 0 & \text{if } U_j \in \mathfrak{D}_v, \\ +\infty & \text{if } U_j \notin \mathfrak{D}_v \\ \text{and } \left(\sum_{U_i \in \mathfrak{N}} X_{i,j}^{v,t-1} = 0 \right), & \\ \sum_{U_i \in \mathfrak{N}} X_{i,j}^{v,t-1}.(D_i^{v,t-1} + D_{i,j}) & \text{if } U_j \notin \mathfrak{D}_v \\ \text{and } \left(\sum_{U_i \in \mathfrak{N}} X_{i,j}^{v,t-1} \neq 0 \right). & \end{cases} \tag{5}$$

Similarly, jitter[3] measures the delay variability in transmitting a single chunk from a user U_i to another user U_j at time t, for a specific video v. The jitter, denoted as $J_j^{v,t}$, is crucial for ensuring smooth and uninterrupted video streams essential for remote patient monitoring and consultations. This is represented as follows [18]:

$$J_j^{v,t} = \sum_{U_i \in \mathfrak{N}} X_{i,j}^{v,t} \cdot J_{i,j}. \tag{6}$$

B. PROBLEM FORMULATION

Responding to the critical need for real-time remote monitoring in healthcare systems, our optimization framework is designed with the specific requirements and challenges of healthcare live video environments in mind. The primary objective remains to minimize the delivery time for video v to requesting users, crucial for ensuring timely medical consultations and patient monitoring, while adhering to the constraints of node resources. The optimization problem to reduce transmission time within network limitations is defined as:

$$\min D_j^{v,t} + J_j^{v,t} - H_j^{v,t}. \tag{7}$$

Thus, to enhance delivery efficiency for all videos to all requesting users at time t, especially under the stringent conditions of healthcare applications where delays can significantly impact patient outcomes, the problem defined in Equation (7) is adapted as follows:

$$\min \sum_{v=1}^{V} \sum_{U \in \mathfrak{D}_v} \left(D_j^{v,t} + J_j^{v,t} - H_j^{v,t} \right) \tag{8}$$

$$\text{s.t.} \sum_{v=1}^{V} \sum_{U_j \in \mathfrak{N}} X_{j,i}^{v,t} \leq CD_i, \forall U_i \in \mathfrak{N} \tag{9}$$

$$\sum_{v=1}^{V} \sum_{U_j \in \mathfrak{N}} X_{i,j}^{v,t} \leq CU_i, \forall U_i \in \mathfrak{N} \tag{10}$$

$$\sum_{U_j \in \mathfrak{D}_v} D_j^{v,t} \leq T, \forall v \in [1..V] \tag{11}$$

In this formulation, CD_i and CU_i represent the download and upload capacities for node i, respectively. Equations (9) and (10) are designed to guarantee that all participating nodes i adhere to the constraints regarding their available upload and download capacities. Additionally, the constraint delineated in Equation (12) mandates that the delay experienced by a receiving node U_j for a video v at any given time t must not surpass the predefined threshold T. These constraints are meticulously formulated to ensure the system's performance aligns with the critical healthcare context, where the efficiency of video delivery directly correlates with the quality of patient care. However, optimizing each time

[3]In healthcare scenarios, jitter represents the variation in delay times for received video chunks, which is crucial for maintaining the quality of live telehealth sessions.

slot independently might not incentivize users to download videos they do not need. Specifically, a user U is likely to download a video v, where $U \notin \mathfrak{D}_v$, only if there is an intention to forward it to another user in subsequent time steps. Consequently, when this optimization extends over all time steps, covering $t = 1, \ldots, N_t$, the problem formulation becomes more complex:

$$\min \sum_{t=1}^{N_t} \sum_{v=1}^{V} \sum_{U \in \mathfrak{D}_v} \left(D_j^{v,t} + J_j^{v,t} - H_j^{v,t}\right) \tag{12}$$

$$\text{s.t.} \sum_{v=1}^{V} \sum_{U_j \in \mathfrak{N}} X_{j,i}^{v,t} \leq CD_i, \forall U_i \in \mathfrak{N} \forall t \in [1..N_t] \tag{13}$$

$$\sum_{v=1}^{V} \sum_{U_j \in \mathfrak{N}} X_{i,j}^{v,t} \leq CU_i, \forall U_i \in \mathfrak{N}, \forall t \in [1..N_t] \tag{14}$$

$$\sum_{U_j \in \mathfrak{D}_v} D_j^{v,t} \leq T, \forall v \in [1..V], \forall t \in [1..N_t] \tag{15}$$

$$\underset{U_j \in \mathfrak{N}}{max} \left(X_{U_j,U_i}^{v,t-1} H_{U_j}^{v,t-1}\right), \forall v \in [1..V], \forall t \in [1..N_t] \; \forall U_i \notin \mathfrak{S}_v \tag{16}$$

$$H_{U_i}^{v,t} = t, \forall \; U_i \in \mathfrak{S}_v. \tag{17}$$

Furthermore, with the revised expressions for $D_j^{v,t}$, $J_j^{v,t}$, and $H_j^{v,t}$, the optimization problem becomes:

$$\min \sum_{t=1}^{N_t} \sum_{v=1}^{V} \sum_{U \in \mathfrak{D}_v} \Big(\sum_{U_i \in \mathfrak{N}} X_{i,j}^{v,t} D_{i,j} + \sum_{U_i \in \mathfrak{N}} X_{i,j}^{v,t} J_{i,j} - \underset{U_i \in \mathfrak{N}}{max} \left(X_{U_i,U_j}^{v,t-1} H_{U_i}^{v,t-1}\right)\Big) \tag{18}$$

$$\text{s.t.} \sum_{v=1}^{V} \sum_{U_j \in \mathfrak{N}} X_{j,i}^{v,t} \leq CD_i, \forall U_i \in \mathfrak{N} \forall t \in [1..N_t] \tag{19}$$

$$\sum_{v=1}^{V} \sum_{U_j \in \mathfrak{N}} X_{i,j}^{v,t} \leq CU_i, \forall U_i \in \mathfrak{N}, \forall t \in [1..N_t] \tag{20}$$

$$\sum_{U_j \in \mathfrak{D}_v} \sum_{U_i \in \mathfrak{N}} X_{i,j}^{v,t} D_{i,j} \leq T, \forall v \in [1..V], \forall t \in [1..N_t] \tag{21}$$

$$\underset{U_j \in \mathfrak{N}}{max} \left(X_{U_j,U_i}^{v,t-1} \underset{U_i \in \mathfrak{N}}{max} \left(X_{U_i,U_j}^{v,t-1} H_{U_i}^{v,t-1}\right)\right), \forall v \in [1..V], \forall t \in [1..N_t] \; \forall U_i \notin \mathfrak{S}_v \tag{22}$$

$$H_{U_i}^{v,t} = t, \forall \; U_i \in \mathfrak{S}_v \tag{23}$$

The constraints of the problem define the allocation of resources and limitations on delay propagation. Consequently, the minimization of the objective function, as shown in (18), is achievable when all the specified constraints in (19) and (20) are met. The following section will introduce the proposed model that dynamically addresses this optimization challenge.

IV. DRL-BASED FRAMEWORK FOR VIDEO STREAMING

In this section, we detail the set of states $\mathbb{S}$, the reward function and the actions $\mathbb{A}$. Also, the DRL algorithm for the dynamic video resources allocations is discussed.

A. DQN ALGORITHM IN THE DRL-LVT FRAMEWORK

To effectively meet the demanding requirements of healthcare networks, which necessitate both low latency and high reliability, we have integrated the DQN algorithm into our DRL-LVT framework. DQN is particularly suited for environments with discrete action spaces, making it suited for our system where node selection for video transmission is linked to capacity constraints. In our system, the decision on whether a user should forward a video segment to another user is represented by binary actions, forming a comprehensive set of actions each corresponding to a possible node-to-node transmission at any given time. Directed by the DRL agent, these actions aim to optimize resource allocation, ensuring that each decision adheres to network constraints while minimizing an objective function that reflects the priorities of real-time remote monitoring.

The choice of DQN over other reinforcement learning algorithms (such as Policy Gradient or Actor-Critic) is based on its superiority in managing binary decision-making processes that are vital in our application. Unlike these other algorithms, which are more suited to continuous action spaces, DQN excels in scenarios requiring precise, binary decisions. Its robust features, such as experience replay and fixed Q-targets, furnish the system with the agility necessary for rapid adaptation to network condition fluctuations, crucial for the efficient distribution of video resources and maintaining stable communication channels. Moreover, DQN's capability to handle complex decision-making processes allows our system to dynamically manage node capacities and navigate high-dimensional state spaces, facilitating informed and efficient decision-making. Leveraging DQN, our system dynamically adjusts strategies based on observed changes, thereby ensuring that decisions are informed by both historical and current data. This adaptability allows for continuous refinement of strategies, thereby enhancing operational efficiency and reliability. These timely adjustments are essential for maintaining the responsiveness of our remote healthcare monitoring system to the fast-evolving nature of wireless networks, consistently meeting the demands of healthcare delivery.

B. DRL STATES

In our adapted model, the DRL agent is aware of the environment's states, which, in a healthcare context, include the urgent need for reliable and uninterrupted video communication. Based on these states, the agent orchestrates actions to ensure optimal resource allocation for video streaming. A crucial aspect of this process involves nodes communicating their available resources and capacities. This communication enables the DRL agent to intelligently select nodes with sufficient remaining capacities for packet transfer. Resource allocation is conducted iteratively for each node, emphasizing the healthcare requirement for continuous and efficient video stream handling. For instance, after node$_1$ allocates a portion of its resources to forward packets, it updates its remaining resources accordingly. This procedure is replicated across all nodes, ensuring the network's capacity is maximized for the demands of healthcare video streaming.

At the beginning of each episode, the DRL agent initializes the values of the download and upload capacities CD_i and CU_i respectively for each node i. Then, the remaining capacities RCU_i and RCD_i will be updated as:

$$RCD_i^t = RCD_i^{t-1} - \sum_{U_j \in \mathfrak{N}} X_{j,i}^t \tag{24}$$

$$RCU_i^t = RCU_i^{t-1} - \sum_{U_i \in \mathfrak{N}} X_{i,j}^t \tag{25}$$

For each video v and corresponding time stamp t, the DRL agent strategically selects the optimal transmission path based on the available remaining capacities. Therefore, the set of states includes not only the investigated video but also a detailed account of the available remaining resource for each node. i.e., $S = \{t, v, RCD_1, RCD_2, \ldots, RCD_{\mathfrak{N}}, RCU_1, RCU_2, .., RCU_{\mathfrak{N}}\}$. Note that the RCD and RCU will be reinitialized at the end of the last video. These updates ensure that the system dynamically adapts to the evolving demands of healthcare video streaming, addressing both the immediate needs for patient care and the overarching goal of enhancing healthcare delivery through technology.

C. DRL ACTIONS

Upon analyzing the current state of the network, the DRL agent strategically selects nodes for the transmission of video chunks, critical for maintaining the continuity and quality of healthcare services through live video feeds. As depicted in Figure 2, the agent's decision-making process begins with the selection of a primary action. This initial decision is then refined into specific sub-actions that involve activating particular communication links within the network, ensuring optimal path selection for video chunk transmission in real-time. For example, if it is determined that user U_i should forward the most recent chunk of video v at time t to user U_j, the decision variable $X_{i,j}^{v,t}$ is assigned a value of one. If not, $X_{i,j}^{v,t}$ is set to zero. Therefore, the action set can be represented as $A = \{X_{i,1}^{v,t}, X_{i,2}^{v,t}, \ldots, X_{i,N}^{v,t}\}$. The actions generated by the DRL agent aim to minimize the objective function detailed in (18) while adhering to all the constraints specified in (19) and (20) offering a robust solution for real-time remote monitoring challenges.

D. DRL REWARD FUNCTION

The optimization problem objective is to transfer the videos to their destinations while satisfying the resource constraints and minimizing:

$$Obj = \sum_{t=1}^{N_t} \sum_{v=1}^{V} \sum_{U \in \mathfrak{D}_v} \Big(\sum_{U_i \in \mathfrak{N}} X_{i,j}^{v,t} D_{i,j} + \sum_{U_i \in \mathfrak{N}} X_{i,j}^{v,t} J_{i,j} - \max_{U_i \in \mathfrak{N}} \big(X_{U_i,U_j}^{v,t-1} H_{U_i}^{v,t-1} \big) \Big). \tag{26}$$

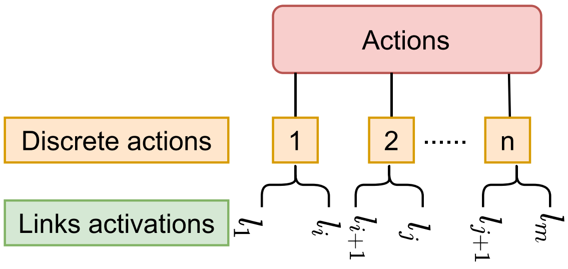

FIGURE 2. DQN actions.

At each time stamp, the problem will be solved for one video in one time stamp. The cumulative reward is computed as:

$$Reward_C = C_1 + C_2 + C_3 - \alpha Obj. \tag{27}$$

where, α represents a constant less than 1. This constant α ensures that the primary focus of the system is on meeting the constraints, followed by the minimization of the objective function. The constraint $C_1(i)$ is related to the delay minimization that should be less than the required time T:

$$C_1(i) = \begin{cases} 1 \text{ if } \sum_{U_j \in \mathfrak{D}_v} X_{i,j}^{v,t} D_{i,j} \le T, \\ 0 \text{ if } \sum_{U_j \in \mathfrak{D}_v} X_{i,j}^{v,t} D_{i,j} > T. \end{cases}$$

For the network with D_v nodes, C_1 becomes:

$$C_1 = 1\Big(\sum_{U_j \in \mathfrak{D}_v} X_{i,j}^{v,t} D_{i,j} \le T \Big). \tag{28}$$

The constraint $C_2(i)$ is to ensure that the node i should not exceed its download capacities while forwarding the data to their destinations:

$$C_2(i) = \begin{cases} 1 \text{ if } \sum_{U_j \in \mathfrak{N}} X_{j,i}^{v,t} \le RCD_i, \\ 0 \text{ if } \sum_{U_j \in \mathfrak{N}} X_{j,i}^{v,t} > RCD_i \end{cases}$$

$$= 1(\sum_{U_j \in \mathfrak{N}} X_{j,i}^{v,t} \le RCD_i)$$

C_2 becomes:

$$C_2 = \sum_{U_i \in \mathfrak{N}} C_2(i) \tag{29}$$

$$= \sum_{U_i \in \mathfrak{N}} 1(\sum_{U_j \in \mathfrak{N}} X_{j,i}^{v,t} \le RCD_i)$$

The constraint $C_3(i)$ is to ensure that the node i should not exceed its upload capacities while forwarding the data to their destinations:

$$C_3(i) = \begin{cases} 1 \text{ if } \sum_{U_j \in \mathfrak{N}} X_{j,i}^{v,t} \le RCU_i, \\ 0 \text{ if } \sum_{U_j \in \mathfrak{N}} X_{j,i}^{v,t} > RCU_i \end{cases}$$

C_3 becomes:

$$C_3(i) = 1\left(\sum_{U_j \in \mathfrak{N}} X_{j,i}^{v,t} \le RCU_i \right)$$

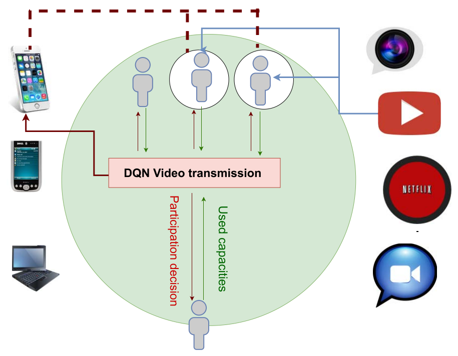

FIGURE 3. DQN predictive model.

The commutative reward is maximized when the objective function is minimized and all the constraints are satisfied. Note that before computing the reward, all the last chunk indexes for all the users must be updated according to Eq. (20).

E. VIDEO TRANSMISSION ALGORITHM

We propose a new algorithm entitled DQN for video live transmission (Algorithm 1) that receives the set of states from the network and decides about the set of actions. The algorithm takes into consideration the conditions in Eq. (18) while optimizing the performance of the DRL agent.

The input comprises the system's state S, which includes the number of nodes $\mathfrak{N}$, the number of videos V, and the available upload and download capacities for each node $(RCD_1, RCD_2, \ldots, RCD_{\mathfrak{N}}, RCU_1, \ldots, RCU_{\mathfrak{N}})$. The iteration rounds are represented by each episode in the algorithm, where actions for video transmission are decided per node. The action set $[X^{v,t}i, 1, X^{v,t}i, 2], \ldots, X^{v,t}i, N$, determined either randomly or through the trained actor based on the exploration rate ϵ, forms the core output mechanism for decision-making. The step size is managed through the update of the exploration rate ϵ_{decay}. After initializing the algorithm (lines 1-4) with the network's nodes $(\mathfrak{N})$, video count (V), and activation functions, we focus on the system state initialization, defined as $S = t, v, RCD_1, RCD_2, \ldots, RCD_{\mathfrak{N}}, RCU_1, \ldots, RCU_{\mathfrak{N}}$, representing time (t), video (v), download capacities (RCD), and upload capacities (RCU) for each node (line 5). The exploration phase begins with $\epsilon = 1$, promoting full action space exploration (line 6). Throughout each episode, the algorithm sequentially processes nodes, deploying a uniformly distributed random variable r to choose between random action selection and trained model predictions (Lines 9-14), effectively balancing exploration with exploitation. This mechanism ensures that actions are not just randomly chosen but are informed by past learning, especially as ϵ gradually decreases. Actions, represented as $[X^{v,t}i, 1, X^{v,t}i, 2], \ldots, X^{v,t}_{i,N}$, are derived either from direct model output or random selection, depending on r's relation to ϵ. Upon selecting an action, the algorithm updates the network's state by activating links and adjusting capacities

Algorithm 1 DQN for Video Transmission, Specifying Input and Output Details

```
1:  Initialize number of nodes $\mathfrak{N}$.
2:  Initialize the number of videos $V$.
3:  Initialize the number of layers and neurons.
4:  Initialize the actor activation functions.
5:  Initialize the system state: the available upload and download capacities for each node, i.e., $S = \{t, v, RCD_1, RCD_2, ..., RCD_{\mathfrak{N}}, RCU_1, \ldots, RCU_{\mathfrak{N}}\}$.
6:  Initialize a full exploration of the model (the exploration rate $\epsilon = 1$).
7:  Initialize exploration decay rate $\epsilon_{\text{decay}}$ to control exploration reduction over time.
8:  for each episode do
9:      for each node do
10:         Generate a random variable $r$.
11:         if $r \leq \epsilon$ then
12:             Generate random action set $[X^{v,t}_{i,1}, X^{v,t}_{i,2}], \ldots, X^{v,t}_{i,N}$.
13:         else
14:             Generate action set $[X^{v,t}_{i,1}, X^{v,t}_{i,2}], \ldots, X^{v,t}_{i,N}$ with the trained actor.
15:         end if
16:         Integrate hierarchical action-value functions.
17:         Activate the links interconnected to $[X^{v,t}_{i,1}, X^{v,t}_{i,2}], \ldots, X^{v,t}_{i,N}$.
18:         Update the cumulative reward.
19:         Update remaining capacities $RC_{\text{remain}} = \{t, v, RCD_1, RCD_2, \ldots, RCD_{\mathfrak{N}}, RCU_1, \ldots, RCU_{\mathfrak{N}}\}$.
20:         Store transition $(s^t_i, a^t_i, r^t_i, s^{t+1}_i)$ in replay buffer.
21:     end for
22:     Update the actor model.
23:     Update the exploration rate to $\epsilon_{\text{decay}}$ so that $\epsilon = \epsilon \times \epsilon_{\text{decay}}$.
24: end for
```

(Lines 15-18), a crucial step for managing the dynamic conditions of live video streaming, particularly in healthcare applications where the stability and reliability of video data are paramount. Rewards received update the cumulative metric, which then influences future decisions and optimizes the network performance for video transmission. Finally, the algorithm's continuous learning loop is maintained through the updates of the actor model and the adjustment of the exploration rate (ϵ), with ϵ experiencing exponential decay to shift the focus from exploration to exploitation as the model's performance improves (Lines 20-21). This gradual transition ensures the model becomes increasingly efficient at predicting optimal actions for video transmission, effectively managing the network's resources to enhance the quality and reliability of live video streams.

V. PERFORMANCE EVALUATION

This section delves into the performance evaluation of the proposed model. Initially, the network topology and

TABLE 1. Comprehensive simulation parameters for DRL-LVT model.

Parameter	Value	Description
Number of Nodes	12	Total number of nodes within the D2D network.
Maximum Upload Capacity	2 Mbps	Maximum upload capacity for each node.
Maximum Download Capacity	2 Mbps	Maximum download capacity for each node.
Exploration Rate (ϵ)	1 decreasing to 0	Initial exploration rate in DQN, adjusting to balance exploration and exploitation.
Discount Factor (λ)	0.99	Factor determining the importance of future rewards in the DQN algorithm.
Learning Rate (α)	0.0001	Step size in the DQN algorithm's optimization process.
Batch Size	10	Number of experiences sampled from the replay buffer for training the DQN model.
Replay Buffer Size	100	Capacity of the replay buffer to store experiences for the DQN model.
Number of Episodes	1000	The total number of episodes for training the DRL model.

the parameters of the DRL algorithm are outlined in Section V-A. The effectiveness of the DRL algorithm is then examined, focusing on the convergence of the objective function and rewards, as detailed in Section V-B and V-C. Additionally, Section V-D explores a dynamic scenario where node capacities vary over time, assessing the DRL algorithm's response in terms of allocation efficiency.

A. SIMULATION ENVIRONMENT

In our simulation setup, each node is assigned fixed capacities for both downloading and uploading, which are pivotal for transmitting videos to their destinations (as defined in Equations (9) and (10)). To evaluate the performance of these methods in adhering to resource constraints, we introduce the concept of **Average Upload/ Download**. This metric represents the mean number of videos uploaded or downloaded by each node. Table 1 presents the simulation and the DRL parameters respectively [9], [34].

B. SIMULATION RESULTS

Figure 4 depicts the DRL cumulative reward. For the first 1000 episodes, the system starts with a full exploration with an exploration rate ϵ to 1. Thus, initially the system estimates a random policy based on random actions, which will form the basis to improve the policies. Then at each episode, the algorithm updates the exponential rate to ϵ_{decay} to get better policies and better interaction between the states and the made actions in time. The better the actions are, the better the performance will be, and the more the cumulative reward is. After 1500 episodes, the DRL agent learns better how to select the users with the adequate capacities to forward the video chunks. Hence, the DRL reward increases fast and the proposed system converges. Figure 5 shows the objective

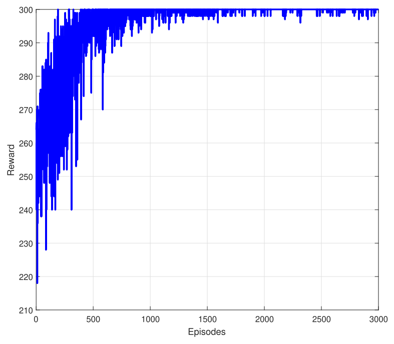

FIGURE 4. The DRL cumulative reward with a varied number of episodes.

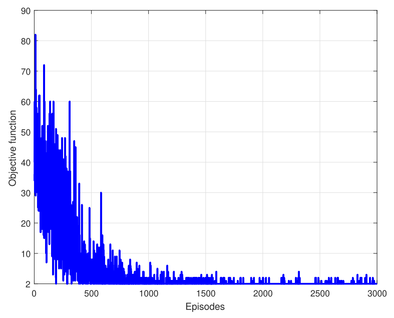

FIGURE 5. The DRL objective function.

function of the DRL model for a varied number of episodes. First, we remark the variation of the objective function since the system is not able yet to make good decisions. Then, we increase the number of episodes, the DRL agent knows better about the environment interaction and is capable to make good actions for each received state. Hence, the objective function and the reward are increased (Figure 4).

The performance of our Deep Reinforcement Learning for Live Video Transmission (DRL-LVT) algorithm is rigorously evaluated against four distinct methodologies: the Flexible Video Transmission Scheme (Flexi) [20], the NoD2D approach [35], the Distributed Random (DR) strategy [35], and the classical Dijkstra algorithm [22]. The NoD2D method is characterized by its reliance on direct node-to-base station connections, offering a centralized but less flexible communication model. In contrast, Flexi employs a frame priority metric for video transmission decisions, based on user feedback on quality and requests for non-received frames, aiming to enhance viewing experiences. The DR approach randomly selects the transmission routes without considering the network's topology or conditions, embodying a truly distributed approach to video streaming. Distinguished by its efficiency in identifying the shortest paths between network nodes, the Dijkstra algorithm serves as a benchmark for optimizing network routing to minimize propagation delays. Further comparisons extend to the Distributed Resource Allocation (CSM) [21] and the Scheduling Algorithm (Sch) [18], both designed with live video streaming's quality of service in mind, focusing on

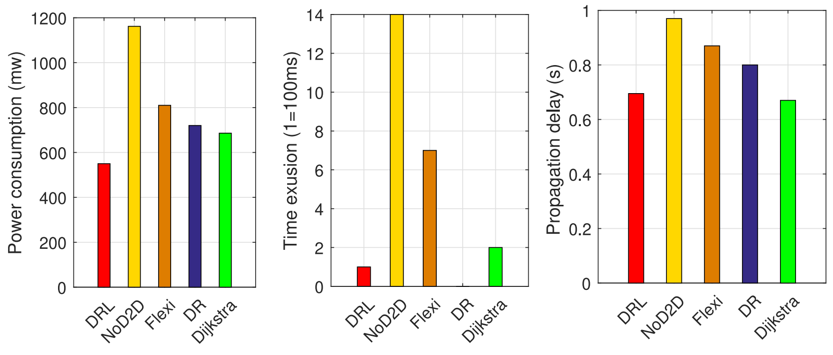

FIGURE 6. Comparison of DRL-LVT against NoD2D, Flexi, DR and Dijkstra.

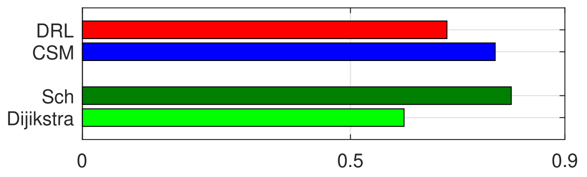

FIGURE 7. Comparison of the propagation delay under CSM, Sch, Dijkstra and DRL.

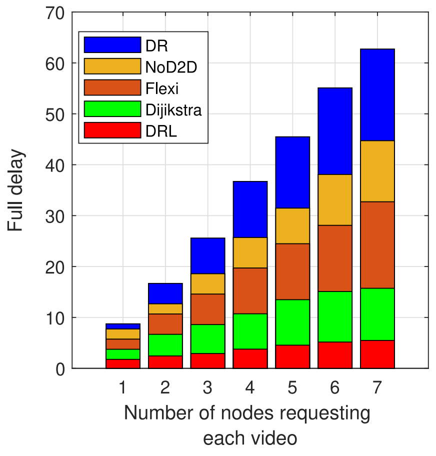

FIGURE 8. Comparison of the full delay with a varied number of nodes requesting each video under DR, NoD2D, Flexi, Dijkstra and DRL.

delivery ratio as a critical performance metric. These collective methodologies provide a comprehensive evaluation for the DRL-LVT framework's performance including the power consumption, execution time (for a 100-node scenario), and propagation delay. Our comparative analysis, illustrated in Figure 6, reveals that while the NoD2D approach leads to increased power usage and propagation delays due to its reliance on direct base station communications, Flexi incurs additional delays and power consumption through its feedback and re-transmission process. In contrast, the DR model, lacking network awareness, achieves lower execution times at the cost of higher propagation delays. The Dijkstra and DRL-LVT algorithms demonstrate superior performance, underscoring the efficacy and innovation of the proposed framework. This evaluation not only highlights the DRL-LVT algorithm's advancements but also broadens the discourse on network management strategies for efficient live video streaming.

In Figure 7, the propagation delays of various algorithms, including CSM, Sch, Dijkstra, and the proposed DRL model, are compared. Notably, the DRL and Dijkstra models demonstrate lower delay times relative to the others. The Sch algorithm necessitates that devices exchange data, such as the most recent video chunk and average playback delay. This process obliges each device to compute its propagation delay based on the information received from neighboring devices. Consequently, this requirement for additional computational work extends the time it takes for video to travel from its source to its destination, contributing to the Sch algorithm's relatively longer delays. In a different approach, CSM, when not paired with blockchain technology, mandates that nodes fetch video chunks from the cloud and send back the processed version through backhaul links. This approach inherently introduces more extended delays compared to other methodologies. On the other hand, the Dijkstra algorithm is adept at finding the shortest routes between nodes within a cluster, effectively cutting down on transmission times. The DRL model, on the other hand, focuses on identifying the optimal path that maximizes rewards while minimizing the objective function, contributing to its efficiency in reducing delays. Figure 8 presents a comparison of the full delay for a varying number of nodes requesting each video using Distributed Random (DR), NoD2D, Flexi, Dijkstra, and DRL-LVT algorithms. Given the constraint that both upload and download capacities are fixed at 2, each video transmission algorithm shows significant shifts in performance. Dijkstra's algorithm, typically adept at identifying the shortest paths, experiences quick saturation of network paths due to these strict capacity limits, resulting in escalated delays as the network load increases. Flexi, dependent on feedback and prioritization, encounters severe compounded delays as the capacity cap significantly restricts its ability to effectively manage re-transmissions. Likewise, the NoD2D and DR strategies face considerable difficulties, their less efficient routing methods culminating in steep increases in delays. In contrast, the DRL-LVT algorithm continues to perform

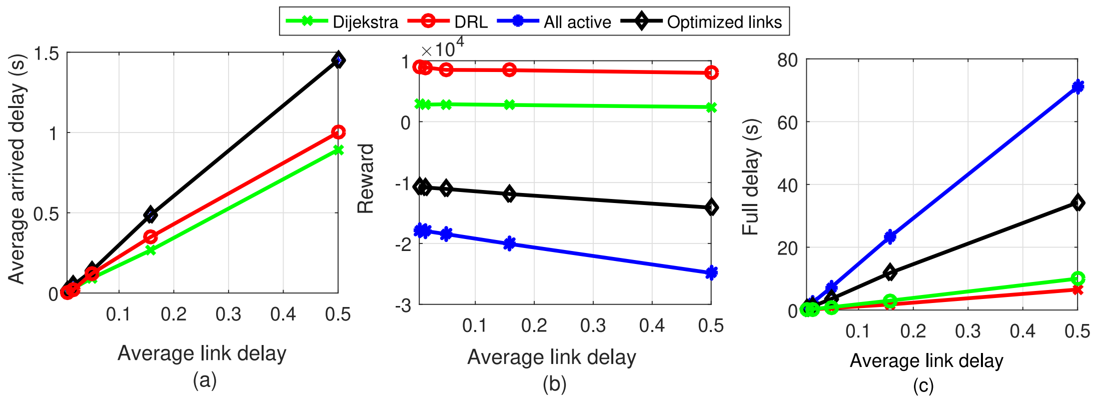

FIGURE 9. **The average arrived delay, the reward, and the full delay of the DRl model compared to Dijkstra, while varying the average link delay.**

robustly, dynamically optimizing transmissions within the capacity constraints, though its efficiency is challenged as the network approaches these limits. Figure 9 shows that the Dijkstra algorithm achieves the lowest propagation delay, which is pivotal for video streaming systems. Consequently, the subsequent sections will compare the proposed DRL-LVT model with the Dijkstra algorithm to further evaluate the efficacy of the DRL approach.

C. DRL FRAMEWORK AGAINST DIJKSTRA

This section focuses on contrasting the performance of the DRL model with the Dijkstra algorithm within a 5-node network. Both models operate under the condition that the upload and download capacities are fixed at 2. The key metric for comparison is the average link delay, defined as the time taken from capturing a video chunk to its display to the viewer. To ensure a fair comparison, limitations regarding download and upload capacities are also applied to the Dijkstra algorithm, assessing its efficiency in resource management. Additionally, the DRL model's performance is evaluated against two other strategies:

- *Active Links Strategy:* This strategy maintains all network links in an active state, enabling users within a cluster to utilize any available link for forwarding packets. This approach ensures maximum connectivity and flexibility in packet routing but may not be the most efficient in terms of resource utilization.
- *Optimized Links Strategy:* By selectively deactivating links that are not currently handling packet transfers from users, this strategy aims to streamline network resource consumption. It focuses on maintaining only those connections that contribute to active data transmission, potentially enhancing the network's overall efficiency and reducing unnecessary bandwidth usage.

Figure 9 depicts a comparison of the average arrived delay, the full delay, and the reward delay for the DRL, Dijkstra, all active, and optimized links approaches. The average link delay varied between 0.005 to 0.5. First, we can see that when the average link delay is increasing, the video chunks take more time to reach their destination (Figure 9 (a)). The active and optimized links have a week performance since both of them are activating an important number of links in the network to forward the video chunks. Dijkstra, instead, has the smallest average delay (Figure 9 (a)) compared to the other techniques. However, in terms of reward (Figure 9 (b)) and full delay (Figure 9 (c)), Dijkstra has a lower performance compared to the DRL. The DRL has dynamically shared the available capacities of nodes according to the network needs. Thus, the system distributes the load on the nodes having remaining capacities and uses the links connected to them to satisfy the constraints while minimizing the delay. However, contrary to the DRL, Dijkstra forwards the data without taking into consideration the nodes' capacities. The only objective is the arrival delay.

As the average delay in the network increases, traditional algorithms like Dijkstra face challenges in managing network constraints effectively. These algorithms often select paths that surpass the upload and download capacities of participant nodes, leading to diminished rewards and compromised performance. This issue becomes particularly critical in healthcare applications, where delays can directly impact patient care and outcomes. In contrast, our proposed DRL model excels by strategically choosing links that distribute the network load more evenly, significantly alleviating bottlenecks in high-delay connections. This approach not only reduces the overall network delay but also ensures that the stringent requirements for real-time video streaming in healthcare contexts are met. The DRL model's superior performance is demonstrated through its ability to maintain high rewards and lower overall delays across various connectivity scenarios, as highlighted in Figure 9 (c). Moreover, the reduction in average arrival delay, crucial for timely patient monitoring, is clearly illustrated in Figure 9 (a). A notable example of the DRL model's effectiveness is presented in Figure 10, depicting a scenario with increasing video demand while node capacities remain constant. Unlike the Dijkstra algorithm, which struggles when requests exceed a certain threshold, the DRL model adeptly manages even large numbers of requests. This is paramount in healthcare settings where the volume of data and the need for its rapid analysis

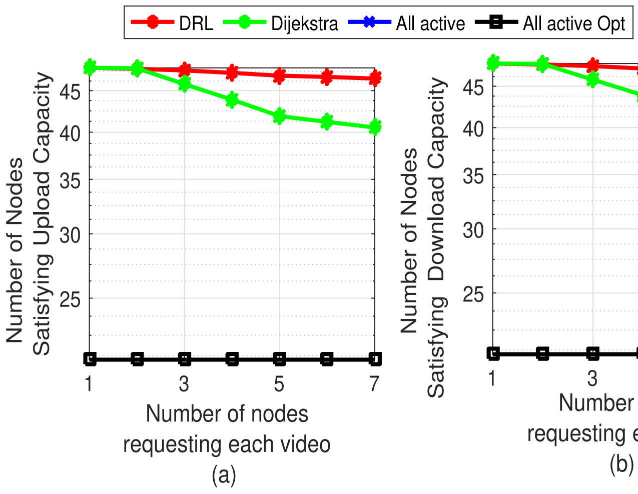

FIGURE 10. The number of nodes satisfying the upload and download capacities for the DRL compared with Dijkstra technique for one chunk transmission.

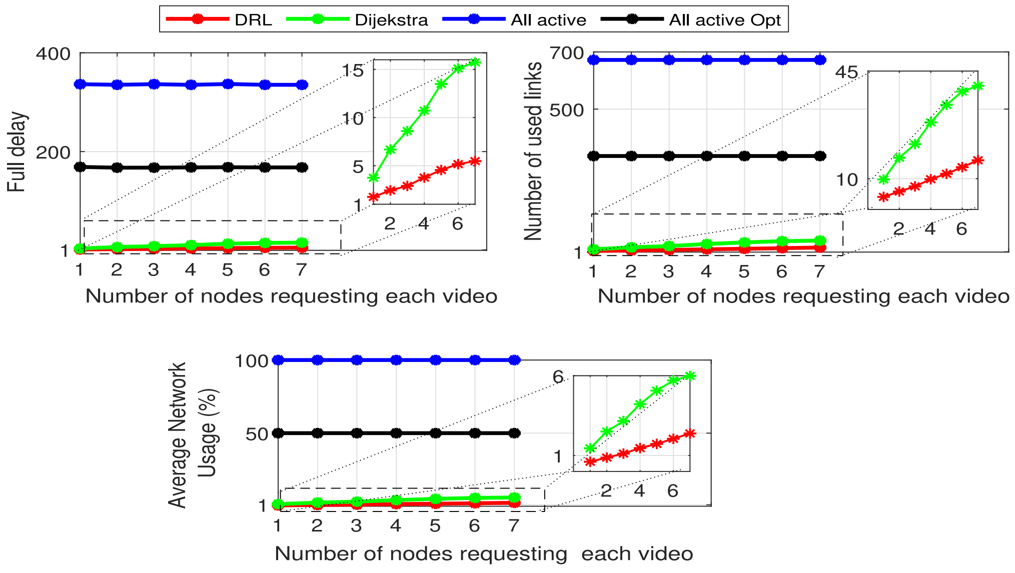

FIGURE 11. Comparison of the full delay, the number of used links and the average network usage for the DRL compared with Dijkstra, all active and optimized links technique.

and transmission can be substantial. Our model ensures that vital video packets are delivered within the nodes' capacity constraints to enhance real-time remote patient monitoring services in healthcare systems.

Figure 11 shows a comparison of the full delay, the number of used links, and the average network usage for the DRL compared with the Dijkstra technique. We remark that all active links have a high rate of network usage (100%) (Figure 11 (c)) since all the links in the network are activated. Optimized links improve the system performance compared to the full active technique, however, the network resource usage still very high (50%). Dijkstra is better than optimized and full links techniques with an average network usage rate equal to 6%. Importantly, unlike Dijkstra, the DRL agent optimizes the set of nodes to be used according to each node's capacities. The use of adequate nodes leads to a lower network resource (Figure 11 (a) and (b)). Figure 12 illustrates a scenario of chunk transmission using both the DRL and Dijkstra algorithms. This example features two nodes competing for resources while requesting videos V_1

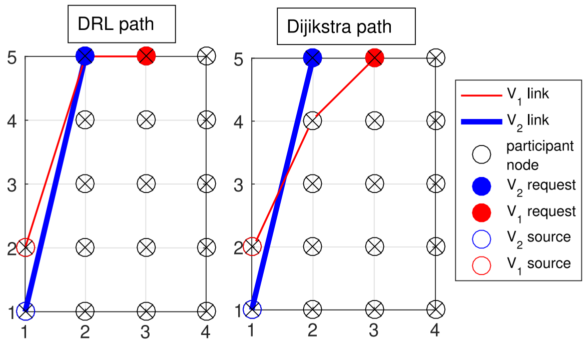

FIGURE 12. Transmission path.

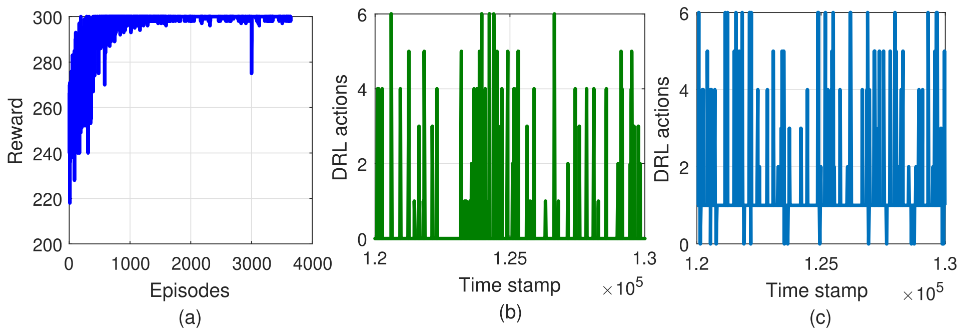

FIGURE 13. The effects of the system changes on the DRL model.

and V_2. It is observed that Dijkstra selects distinct paths for transmitting each video. Conversely, the DRL approach utilizes the same link for forwarding the video V_1 packets. This pattern suggests that DRL effectively reduces the usage of network resources during the transmission of video chunks.

D. THE EFFECTS OF SYSTEM CHANGES

In this section, the effects of the system changes of the proposed DRL model to the video transmission demand over time is investigated. In particular, we assume that each participant node has initially a download and upload capacities equal to 5. Then, after 3000 episodes, the nodes reduce their capacities to 2.

Figure 13 (a) depicts the reward function with a varied number of episodes. The node capacities have been changed after the system convergence. The DRL agent reacts rapidly and redistributes the resources according to the new requests and capacities. In fact, the DRL allocates the links according to the nodes' capacities. In particular, if the maximum download and upload capacities of a node is high, then the system can use the same link that has the lowest delay to froward all the data (Figure 13 (b)). However, if the available capacities is reduced the network has to search for alternative path to satisfied the capacities constraints without losing a lot in terms of delay (Figure 13 (c)).

VI. DISCUSSION

In the experimental setup, each node is precisely allocated fixed capacities for both uploading and downloading, a critical consideration for assessing performance in healthcare remote monitoring scenarios where data integrity and timely delivery are paramount. We evaluate effectiveness through the Average Upload/Download metric, capturing the mean video transactions per node to reflect operational efficiency under the stringent conditions detailed in Table 1, which are particularly relevant to healthcare applications.

In our comprehensive comparative analysis of the DRL-LVT algorithm against established methods such as the Flexi, NoD2D, DR strategies, and the Dijkstra algorithm, we specifically address the unique requirements of healthcare systems. These include ensuring high reliability, minimal delays, and secure transmission key factors in patient monitoring and emergency communication scenarios. Our analysis includes critical metrics such as power consumption, execution time, and propagation delay, which are pivotal for supporting healthcare applications where delays can impact patient outcomes.

Preliminary findings highlight the DRL-LVT model's capability to adeptly navigate the complexities of network environments typical in healthcare settings, ensuring efficient video transmission by adapting dynamically to node-specific characteristics. This adaptability results in enhanced performance metrics, significantly reducing delays and optimizing resource use, factors that are critical in healthcare where data must be both timely and accurate. Notably, while the Dijkstra algorithm focuses on minimizing propagation delays through shortest path calculations. Furthermore, contrasting with algorithms like the Sch scheme that suffer from long delays due to their dependency on continuous updates, our model excels in real-time video forwarding without such constraints. The DRL-LVT approach not only demonstrates its superiority in managing propagation delays but also showcases its robustness in scenarios characterized by high demand and limited resources, typical of remote healthcare monitoring. Thus, our comparative analysis establishes the DRL-LVT algorithm as a superior choice for live video transmission optimization in demanding healthcare environments. It affirms our model's capability to enhance network efficiency and quality of service, making it particularly suitable for healthcare applications where these aspects are critical. This addresses the unique challenges faced in healthcare remote monitoring, providing a robust solution that caters specifically to the needs of this sector.

VII. CONCLUSION

This paper introduces the innovative Deep Reinforcement Learning for Live Video Transmission (DRL-LVT) model, specifically designed for Real-Time Remote Patient Monitoring in healthcare settings. The model effectively merges the capabilities of real-time video streaming with strategic resource management. Early evaluations of DRL-LVT show a promising balance in reducing network latency and optimizing resource use. Utilizing a DRL agent, the model skillfully navigates node selection for video transmission, considering their capacity limits to ensure minimal delay and adherence to network constraints. Our comparative studies highlight DRL-LVT's superiority over traditional methods like the Dijkstra algorithm, particularly in situations with high resource demand and limited availability. This advancement positions DRL-LVT as a pivotal development in the field of real-time patient monitoring. Future research efforts will focus on refining the DRL-LVT framework, increasing network incentives, and conducting extensive field tests to confirm its effectiveness and adaptability in real-world scenarios. Collaborations with healthcare stakeholders will be crucial for tailoring the framework to specific needs and conducting pilot studies in controlled environments. These studies will assess key performance metrics, integration with existing infrastructures, and compliance with privacy standards. This strategic approach will validate the framework's practical efficacy in healthcare settings. Additionally, recognizing the critical importance of data security, future versions of the DRL-LVT framework will enhance security measures to protect sensitive patient data from emerging cyber threats, ensuring a system that is not only effective and adaptive but also secure and trustworthy.

ACKNOWLEDGMENT

The statements made herein are solely the responsibility of the authors.

REFERENCES

[1] Z. Tian, L. Zhao, L. Nie, P. Chen, and S. Chen, "Deeplive: QoE optimization for live video streaming through deep reinforcement learning," in *Proc. IEEE 25th Int. Conf. Parallel Distrib. Syst. (ICPADS)*, 2019, pp. 827–831.

[2] D. Wu, J. Yan, H. Wang, D. Wu, and R. Wang, "Social attribute aware incentive mechanism for device-to-device video distribution," *IEEE Trans. Multimedia*, vol. 19, no. 8, pp. 1908–1920, Aug. 2017.

[3] X. Zhang, M. Yang, Y. Zhao, J. Zhang, and J. Ge, "An SDN-based video multicast orchestration scheme for 5G ultra-dense networks," *IEEE Commun. Mag.*, vol. 55, no. 12, pp. 77–83, Dec. 2017.

[4] L. Rui, S. Yang, Z. Gao, W. Li, X. Qiu, and L. Meng, "Smart network maintenance in edge cloud computing environment: An allocation mechanism based on comprehensive reputation and regional prediction model," *J. Netw. Comput. Appl.*, vol. 198, Feb. 2022, Art. no. 103298. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1084804521002915

[5] Z. Chkirbene, A. Mohamed, A. Erbad, and M. Guizani, "Energy-efficient networks selection based deep reinforcement learning for heterogeneous health systems," in *Proc. IEEE Int. Conf. E-Health Netw., Appl. Services (HEALTHCOM)*, 2021, pp. 1–6.

[6] G. Wu, Y. Zhao, Y. Shen, H. Zhang, S. Shen, and S. Yu, "DRL-based resource allocation optimization for computation offloading in mobile edge computing," in *Proc. IEEE INFOCOM Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, 2022, pp. 1–6.

[7] P. Cheng, Y. Chen, M. Ding, Z. Chen, S. Liu, and Y.-P. P. Chen, "Deep reinforcement learning for online resource allocation in IoT networks: Technology, development, and future challenges," *IEEE Commun. Mag.*, vol. 61, no. 6, pp. 111–117, Jun. 2023.

[8] Y. Du, Y. Xu, L. Xue, L. Wang, and F. Zhang, "An energy-efficient cross-layer routing protocol for cognitive radio networks using apprenticeship deep reinforcement learning," *Energies*, vol. 12, no. 14, p. 2829, 2019.

[9] Z. Chkirbene, A. A. Abdellatif, A. Mohamed, A. Erbad, and M. Guizani, "Deep reinforcement learning for network selection over heterogeneous health systems," *IEEE Trans. Netw. Sci. Eng.*, vol. 9, no. 1, pp. 258–270, Feb. 2022.

[10] L. Liu and Z. Li, "Permissioned blockchain and deep reinforcement learning enabled security and energy efficient healthcare Internet of Things," *IEEE Access*, vol. 10, pp. 53640–53651, 2022.

[11] Z. Chkirbene, R. Hamila, and A. Erbad, "Secure medical data sharing for healthcare system," in *Proc. IEEE 33rd Annu. Int. Symp. Pers., Indoor Mobile Radio Commun. (PIMRC)*, 2022, pp. 641–647.

[12] H. Lin, W. Xu, N. Guan, D. Ji, Y. Wei, and W. Yi, "Noninvasive and continuous blood pressure monitoring using wearable body sensor networks," *IEEE Intell. Syst.*, vol. 30, no. 6, pp. 38–48, Dec. 2015.

[13] E. Spano, S. Di Pascoli, and G. Iannaccone, "Low-power wearable ECG monitoring system for multiple-patient remote monitoring," *IEEE Sensors J.*, vol. 16, no. 13, p. 1, Jul. 2016.

[14] N. Bui, N. Bressan, and M. Zorzi, "Interconnection of body area networks to a communications infrastructure: An architectural study," in *Proc. 18th Eur. Wireless Conf.*, 2012, pp. 1–8.

[15] S. Friedman, B. Munoz, S. West, G. Rubin, and L. Fried, "Falls and fear of falling: Which comes first? A longitudinal prediction model suggests strategies for primary and secondary prevention," *J. Am. Geriatr. Soc.*, vol. 50, no. 8, pp. 1329–1335, Aug. 2002.

[16] R. G. Cumming, G. Salkeld, M. Thomas, and G. Szonyi, "Prospective study of the impact of fear of falling on activities of daily living, SF-36 scores, and nursing home admission," *J. Gerontol., Ser. A*, vol. 55, no. 5, pp. M299–M305, May 2000.

[17] Y. Suryandari, "Survei IoT healthcare device," *Jurnal Sistem Cerdas*, vol. 3, no. 2, pp. 153–164, 2020.

[18] U. Abbasi and H. Elbiaze, "Multimedia streaming using D2D in 5G ultra dense networks," in *Proc. 15th IEEE Annu. Consum. Commun. Netw. Conf. (CCNC)*, 2018, pp. 1–6.

[19] X. Wu et al., "FlashLinQ: A synchronous distributed scheduler for peer-to-peer ad hoc networks," *IEEE/ACM Trans. Netw.*, vol. 21, no. 4, pp. 1215–1228, Aug. 2013.

[20] Q. Ren, J. Chen, B. Chen, and L. Jin, "A video streaming transmission scheme based on frame priority in device-to-device multicast networks," *IEEE Access*, vol. 7, pp. 20187–20198, 2019.

[21] N. C. Luong, Z. Xiong, P. Wang, and D. Niyato, "Optimal auction for edge computing resource management in mobile blockchain networks: A deep learning approach," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2018, pp. 1–6.

[22] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, no. 1, pp. 269–271, 1959.

[23] S. Levine, P. Pastor, A. Krizhevsky, and D. Quillen, "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection," 2016, *arXiv:1603.02199*.

[24] H. Mao, R. Netravali, and M. Alizadeh, "Neural adaptive video streaming with Pensieve," in *Proc. Conf. ACM Spec. Interest Group Data Commun.*, 2017, pp. 197–210.

[25] T. Huang, R.-X. Zhang, C. Zhou, and L. Sun, "QARC: Video quality aware rate control for real-time video streaming based on deep reinforcement learning," in *Proc. 26th ACM Int. Conf. Multimedia*, 2018, pp. 1208–1216.

[26] G. Xiao, M. Wu, Q. Shi, Z. Zhou, and X. Chen, "DeepVR: Deep reinforcement learning for predictive panoramic video streaming," *IEEE Trans. Cogn. Commun. Netw.*, vol. 5, no. 4, pp. 1167–1177, Dec. 2019.

[27] W. Zhang et al., "Optimizing federated learning in distributed industrial IoT: A multi-agent approach," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 12, pp. 3688–3703, Dec. 2021.

[28] W. Zhang et al., "Deep reinforcement learning based resource management for DNN inference in industrial IoT," *IEEE Trans. Veh. Technol.*, vol. 70, no. 8, pp. 7605–7618, Aug. 2021.

[29] D. Yang et al., "DetFed: Dynamic resource scheduling for deterministic federated learning over time-sensitive networks," *IEEE Trans. Mobile Comput.*, vol. 23, no. 5, pp. 5162–5178, May 2024.

[30] R. Hong, Q. Shen, L. Zhang, and J. Wang, "Continuous bitrate & latency control with deep reinforcement learning for live video streaming," in *Proc. 27th ACM Int. Conf. Multimedia*, 2019, pp. 2637–2641.

[31] J. Zhang, G. Yu, Z. He, L. Ai, and P. Chen, "DeepPower: Deep reinforcement learning based power management for latency critical applications in multi-core systems," in *Proc. 52nd Int. Conf. Parallel Process.*, 2023, pp. 327–336.

[32] Y. He et al., "Deep-reinforcement-learning-based optimization for cache-enabled opportunistic interference alignment wireless networks," *IEEE Trans. Veh. Technol.*, vol. 66, no. 11, pp. 10433–10445, Nov. 2017.

[33] Y. Yang, Y. Wang, K. Liu, N. Zhang, S. Gu, and Q. Zhang, "Deep reinforcement learning based online network selection in CRNs with multiple primary networks," *IEEE Trans. Ind. Informat.*, vol. 16, no. 12, pp. 7691–7699, Dec. 2020.

[34] Z. U. A. Tariq, E. Baccour, A. Erbad, and M. Hamdi, "Reinforcement learning for resilient aerial-IRS assisted wireless communications networks in the presence of multiple Jammers," *IEEE Open J. Commun. Soc.*, vol. 5, pp. 15–37, 2024.

[35] I. Ioannou, V. Vassiliou, C. Christophorou, and A. Pitsillides, "Distributed artificial intelligence solution for D2D communication in 5G networks," *IEEE Syst. J.*, vol. 14, no. 3, pp. 4232–4241, Sep. 2020.

ZINA CHKIRBENE received the bachelor's degree in computer science networks and telecommunications from the National Institute of Applied Science and Technology in 2011, the master's degree in electronic systems and communication networks from Tunisia Polytechnic School in 2012, and the Ph.D. degree in computer science from the University of Burgundy, Dijon, France, in 2017. She was a Research Assistant with Qatar University on a project covering the interconnection networks for massive data centers, where she currently holds a Postdoctoral Position. Her research interests include data center networks, edge computing, green computing, and machine learning as well as deep reinforcement learning techniques.

RIDHA HAMILA (Senior Member, IEEE) received the M.Sc. degree, the Licentiate of Technology degree (Hons.), and the Doctor of Technology degree from the Department of Information Technology, Tampere University of Technology (TUT), Tampere, Finland, in 1996, 1999, and 2002, respectively, where he held various Research and Teaching Positions with the Department of Information Technology, from 1994 to 2002. From 2002 to 2003, he was a System Specialist with the Nokia Research Center and Nokia Networks, Helsinki. From 2004 to 2009, he was with the Etisalat University College, Emirates Telecommunications Corporation, UAE. He served as a Supervisor for a large number of the Bachelor's/Master's Students and the Postdoctoral Fellows. He is currently an Associate Professor with the Department of Electrical Engineering, Qatar University, Qatar. He was an Adjunct Professor with the Department of Communications Engineering, TUT. He is currently a Full Professor with the Department of Electrical Engineering, Qatar University, Qatar. His current research interests include mobile and broadband wireless communication systems, mobile-edge computing, Internet of Everything, and machine learning.

DEVRIM UNAL (Senior Member, IEEE) received the M.Sc. degree in telematics from Sheffield University, U.K., and the Ph.D. degree in computer engineering from Bogazici University, Türkiye, in 1998 and 2011, respectively. He is currently a Research Assistant Professor of cyber security with the KINDI Center for Computing Research, College of Engineering, Qatar University. His research interests include cyber-physical systems and IoT security, wireless security, artificial intelligence, and next generation networks.

MONCEF GABBOUJ (Fellow, IEEE) received the B.S. degree in electrical engineering from Oklahoma State University, Stillwater, OK, USA, in 1985 and the M.S. and Ph.D. degrees in electrical engineering from Purdue University, West Lafayette, IN, USA, in 1986 and 1989, respectively. He is currently a Professor of signal processing with the Department of Computing Sciences, Tampere University, Tampere, Finland. His research interests include big data analytics, multimedia content-based analysis, indexing and retrieval, artificial intelligence, machine learning, pattern recognition, nonlinear signal and image processing and analysis, voice conversion, and video processing and coding.

MOUNIR HAMDI (Fellow, IEEE) received the B.S. degree from the University of Louisiana in 1985, and the M.S. and Ph.D. degrees in electrical engineering from the University of Pittsburgh in 1987 and 1991, respectively. He was a Chair Professor and a Founding Member of the Hong Kong University of Science and Technology, where he was the Head of the Department of Computer Science and Engineering. From 1999 to 2000, he was a Visiting Professor with Stanford University and the Swiss Federal Institute of Technology. He is the Founding Dean of the College of Science and Engineering, Hamad Bin Khalifa University, Qatar.