IEEE ComSoc | IEEE COMPUTER SOCIETY | IEEE Signal Processing Society | VTS | IEEE Transactions on Machine Learning in Communications and Networking

Received 26 February 2024; revised 25 July 2024 and 29 August 2024; accepted 29 August 2024.
Date of publication 5 September 2024; date of current version 18 September 2024.

The associate editor coordinating the review of this article and approving it for publication was M. Mozaffari.

Digital Object Identifier 10.1109/TMLCN.2024.3455268

RSMA-Enabled Interference Management for Industrial Internet of Things Networks With Finite Blocklength Coding and Hardware Impairments

NAHED BELHADJ MOHAMED[1], MD. ZOHEB HASSAN[2] (Member, IEEE), AND GEORGES KADDOUM[1,3] (Senior Member, IEEE)

[1]Department of Electrical Engineering, École de technologie supérieure (ÉTS), Université du Québec, Montreal, QC H3C 1K3, Canada
[2]Department of Electrical and Computer Engineering, Université Laval, Québec, QC G1V 0A6, Canada
[3]Cyber Security Systems and Applied AI Research Center, Lebanese American University, Beirut 03797751, Lebanon

Corresponding author: M. Z. HASSAN (md-zoheb.hassan@gel.ulaval.ca)

This work was supported in part by Canada Research Chair Program Tier-II entitled "Toward a Novel and Intelligent Framework for the Next Generations of IoT Networks" and in part by the Discovery Grant from the Natural Science and Engineering Research Council (NSERC) of Canada.

ABSTRACT The increasing proliferation of industrial internet of things (IIoT) devices requires the development of efficient radio resource allocation techniques to optimize spectrum utilization. In densely populated IIoT networks, the interference that results from simultaneously scheduling multiple IIoT devices over the same radio resource blocks (RRBs) severely degrades a network's achievable capacity. This paper investigates an interference management problem for IIoT networks that considers both finite blocklength (FBL)-coded transmission and signal distortions induced by hardware impairments (HWIs) arising from practical, low-complexity radio-frequency front ends. We use the rate-splitting multiple access (RSMA) scheme to effectively schedule multiple IIoT devices in a cluster over the same RRB(s). To enhance the system's achievable capacity, a joint clustering and transmit power allocation (PA) problem is formulated. To tackle the optimization problem's inherent computational intractability due to its non-convex structure, a two-step distributed clustering and power management (DCPM) framework is proposed. First, the DCPM framework obtains a set of clustered devices for each access point by employing a greedy clustering algorithm while maximizing the clustered devices' signal-to-interference-plus-noise ratio. Then, the DCPM framework employs a multi-agent deep reinforcement learning (DRL) framework to optimize transmit PA among the clustered devices. The proposed DRL algorithm learns a suitable transmit PA policy that does not require precise information about instantaneous signal distortions. Our simulation results demonstrate that our proposed DCPM framework adapts seamlessly to varying channel conditions and outperforms several benchmark schemes with and without HWI-induced signal distortions.

INDEX TERMS Deep reinforcement learning (DRL), Industrial Internet of Things (IIoT), finite block length (FBL), hardware impairment (HWI), rate-splitting multiple access (RSMA).

I. INTRODUCTION

THE industrial internet of things (IIoT) has emerged as a transformative force that is propelling industries into a new era of automation, data-driven decision-making, and operational efficiency. IIoT networks incorporating devices, robots, and healthcare systems that require low-latency (around 1 ms) communication are anticipated to grow in scale in future-generation wireless networks [1]. Significant challenges must be overcome to meet such stringent quality-of-service (QoS) requirements, particularly over time-varying fading channels. Finite blocklength (FBL) codes are used to reduce the latency of over-the-air transmission in applications, such as industrial automation where control packets of around 100 bits are transmitted to provide instructions to

robots and autonomous vehicles [2]. This is in stark contrast to the conventional practice of transmitting arbitrary long codes for negligible error. Moreover, next-generation IIoT networks for 6G-based Industry 5.0 are expected to support much higher data rates, traffic volumes, and connection density levels (devices/km^2) than IIoT networks designed for Industry 4.0 [3]. In such large-scale IIoT networks, the fact that spectrum resources are limited means that radio resource blocks (RRBs) must be shared among multiple IIoT devices, which leads to co-channel interference. Additionally, IIoT devices often use low-complexity radio-frequency (RF) front ends that are power-efficient and cost-effective but usually result in hardware impairment (HWI)-induced signal distortions. Co-channel interference and HWI-induced signal distortions make it difficult to achieve the QoS levels that IIoT networks require over time-varying fading channels. Thus, effective radio resource allocation that takes into consideration FBL-coded transmission, co-channel interference, and HWI-induced signal distortions is crucial for improving spectrum utilization in IIoT networks.

Device clustering and transmit power allocation (PA) are critical for managing interference and improving spectrum utilization in wireless networks [4]. More specifically, clustering IIoT devices can improve the utilization of the limited RRBs that are available, and optimizing transmit PA in multiple access (MA) systems can enhance system capacity over the inferring channels. Accordingly, this work aims to develop a device clustering and transmit PA optimization framework to improve resource utilization and manage interference in IIoT networks.

A. RELATED WORKS

1) RELATED WORKS ON RATE-SPLITTING MULTIPLE ACCESS (RSMA)

Designing optimal MA schemes is fundamentally important for interference mitigation and improving resource utilization in multi-user networks. Although orthogonal frequency-division multiple access (OFDMA), which the LTE-A and 5G new radio standards rely on, can avoid multi-user interference by allocating each user dedicated RRBs, it is inherently resource-inefficient for dense networks. Recently, RSMA has emerged as a novel MA scheme to simultaneously schedule multiple users over the same RRB. RSMA manages interference in multi-antenna systems by splitting messages into the power and spatial domains and providing flexibility between complete and partial interference cancellation at user devices [5]. Furthermore, the literature attests that RSMA requires fewer successive interference cancellation (SIC) operations at devices than non-orthogonal multiple access (NOMA) [6], which makes RSMA more suitable for low-complexity IIoT devices. In addition, even in single-antenna multi-user networks, RSMA can achieve a higher sum rate than NOMA in the presence of erroneous channel state information (CSI) at the transmitter or practical SIC constraints [7], [8]. RSMA's effectiveness when it comes to improving the energy efficiency of single-antenna multi-user networks is also demonstrated [9]. While these studies report RSMA's advantages for infinite blocklength-coded systems, RSMA has also been shown to be effective at dealing with interference in FBL-coded systems. For instance, the authors of [10] optimize the RSMA scheme for a two-user FBL-coded uplink communication system, while the authors of [11] propose to optimize the precoding matrix in order to maximize the sum rate in multi-antenna FBL downlink communications that require low latency. However, the HWI-induced signal distortions resulting from the FBL-coded devices' low-complexity RF front ends were ignored in these studies. In addition, both [10] and [11] rely on time-consuming iterative optimization methods to find the near-optimal solution for each channel fading state, which is challenging to implement in delay-constrained networks.

2) RELATED WORKS ON PA

Several transmit PA schemes for FBL-coded IIoT systems are proposed in the state-of-the-art literature. In [12], joint power and bandwidth allocation and active base station (BS) antenna selection are proposed to maximize the short blocklength regime's energy efficiency in ultra-reliable low-latency communications (URLLCs). However, the URLLC rate expression that is considered is not appropriate. The authors of [13] propose to use beamforming optimization to maximize the weighted sum rate in multiple-input single-output and multi-user downlink URLLC networks. In [14], joint bandwidth and PA is proposed to ensure max-min fairness in a multi-user URLLC network. In [15], the authors study the joint optimization of FBL and shared-pilot length for a multi-device downlink IIoT network.

While the aforementioned works focus on downlink systems, uplink resource allocation is also important for IIoT networks. The authors of [16] study jointly optimizing the blocklength and a drone-mounted flying BS's position, height, and beamwidth to minimize the power consumption of uplink URLLC devices. The authors of [17] investigate how transmit PA can be utilized to minimize the decoding error of relay-assisted URLLC IoT systems. In [18], a joint pilot and payload PA scheme is proposed for massive multiple-input multiple-output (MIMO)-enabled IIoT networks.

However, the aforementioned studies either ignore interference in the network by using orthogonal RRB allocation [12], [14], [15], [16], [17], [18] or adopt complex iterative optimization methods [13]. Notably, in an FBL-coded system, the achievable rate is a highly evolved function of the transmit power, blocklength, and decoding error probability [19], and leads to computationally challenging non-convex resource allocation problems. The simultaneous presence of multi-user interference and HWI-induced signal distortions makes the resource allocation problems even more complicated. Standard iterative optimization methods, such as successive convex approximation (SCA) and difference-of-convex

(DC) optimization, require many iterations and a significant amount of memory to converge, which makes them inefficient for dynamic resource optimization in large-scale IIoT networks. Recently, deep reinforcement learning (DRL) has emerged as a promising technique for solving complex optimization problems in various fields, including IoT networks [20], heterogeneous cellular networks [21], and cognitive radio networks [22]. More specifically, the fact that DRL is able to learn a suitable policy by adapting to dynamic and complex environments makes it an ideal candidate for PA in IIoT networks, especially in the presence of co-channel interference and unknown HWI-induced signal distortions.

3) RELATED WORKS ON DEVICE CLUSTERING

The clustering of devices in accordance with specific criteria has been studied extensively and in combination with various access technologies, including NOMA [23], [24], OFDMA [25], [26] and RSMA, in the literature. In the context of RSMA technologies, in particular, the existing research delves into employing the RSMA strategy within each cluster. This strategy aims to schedule the devices in an RRB in such a way as to potentially enhance network performance and resource utilization. In [9], the authors propose to use a clustering algorithm to organize devices into non-overlapping clusters according to their respective locations. Additionally, they apply an RSMA strategy to enable devices to receive data from a suitable fog access point (AP) and a cloud BS over the same RRB. Similarly, in [7], the authors extend the application of the RSMA strategy to each cluster. However, the authors employ a distinct approach for clustering user devices (UDs) in this context. They utilize a multi-agent RL technique to maximize the UD's long-term achievable data rate. In [27], the authors propose a low-complexity k-means clustering algorithm that dynamically divides multiple users into clusters based on their respective locations.

The aforementioned works demonstrate that clustering can improve the RSMA strategy's performance. However, how clustering can be exploited to enhance the performance of RSMA-aided networks in the presence of FBL-coded data transmission and HWI-induced signal distortions remains largely unexplored in the existing literature. This research gap is addressed in this work.

B. MOTIVATION

Despite significant advancements in resource optimization having been made, the current literature fails to address some limitations, particularly in terms of clustering IIoT devices and allocating transmit power among device clusters[1] while considering FBL, co-channel interference, and HWI-induced distortions. Most radio resource allocation studies overlook the detrimental impact HWIs have on network resource optimization. Note that, HWIs caused by non-linearity in low-cost and low-power IIoT devices can significantly reduce the average sum rate achieved. Practical radio equipment, such as power amplifiers and filters, often have inherent limitations that cannot be ignored [28]. Additionally, it is impractical to assume Shannon channel capacity in the context of IIoT networks since the transmitted packets' blocklengths are short due to the stringent delay requirements. Consequently, FBL information theory must be employed in place of the classical Shannon rate formula to accurately estimate the rate performance of finite packet transmission. Notably, the optimization of transmit power and device clustering in a downlink FBL-coded IIoT network is a computationally intractable problem when HWI-induced distortions are present. Off-the-shelf optimization tools usually fail to provide scalable solutions for these types of optimization problems. Furthermore, these approaches require accurate HWI models for devices, which are not always available in practice. These challenges motivate our work to develop an optimized device clustering and PA framework in order to improve the system capacity of FBL-coded IIoT networks with non-negligible co-channel interference and HWI-induced distortions.

C. CONTRIBUTIONS AND PAPER ORGANIZATION

This work presents a comprehensive resource optimization framework to enhance the system capacity of RSMA-enabled IIoT networks in the presence of FBL-coded data transmission and HWI-induced signal distortions. This work's specific contributions are summarized below.

1) This work investigates a downlink data transmission scheme for an FBL-coded multi-cell IIoT network. Each AP in the envisioned network utilizes dedicated orthogonal RRBs for data transmission to overcome inter-cell interference. Each AP's devices are grouped into non-overlapping clusters such they are concurrently scheduled over the same RRB. Using the RSMA strategy enables the IIoT devices in each cluster to receive data over the same RRB, which makes the proposed framework particularly suitable for resource-constrained IIoT networks. Note that the network's system capacity is impaired by both intra-cell co-channel interference and HWI-induced signal distortions arising from the practical low-complexity RF front ends. A sum rate maximization problem is formulated to optimize the clusters of IIoT devices and select a suitable transmit PA for the interfering IIoT links in order to mitigate the HWI-induced signal distortions. To the best of the authors' knowledge, this is the first work to develop a device clustering and transmit PA solution for IIoT networks that explicitly considers the detrimental effects of FBL, co-channel interference, and HWI-induced signal distortions.
2) The joint device clustering and transmit PA problem is proved to be NP-hard and computationally intractable. A distributed clustering and power

[1] In this work, an IIoT device cluster implies the set of IIoT devices that are scheduled over the same RRB. We consider a dense IIoT network and assume that IIoT devices are clustered to ensure efficient spectrum use.

TABLE 1. Table of variables and descriptions.

Variable	Description	Variable	Description
M	Number of APs	θ_c	SIC error coefficient
K	Number of devices per AP	n_b	Blocklength
N	Number of clusters	ϵ	Block error probability
K_D	Number of devices per cluster	$R_{C_n}^{(m)}$	Achievable sum-rate of the n-th device-cluster
$h_{k,n}^{(m)}$	Small-scale block Rayleigh fading	$R_{\text{total}}^{(m)}$	Total achievable sum-rate of the m-th AP
$\beta_{k,n}^{(m)}$	Large-scale fading	B	Total system bandwidth
$g_{k,n}^{(m)}$	Channel gain	$\mathcal{U}_{MU}$	Set of unassigned IIoT devices
ρ	Correlation coefficient	$\mathcal{U}_{MR}$	Set of RRBs that can accommodate at least one IIoT device
f_d	Doppler Frequency	D_n	Number of IIoT devices assigned to the n-th RRB
T_s	Time interval	R	Cell radius
$Pr_{\text{LOS}}, Pr_{\text{NLOS}}$	LoS and NLoS probabilities	r_s	Small region radius
$PL_{\text{LOS}}, PL_{\text{NLOS}}$	LoS and NLoS path losses	P_t, $P_{total,n}$	Total power, total power provided by the AP to the n-th cluster
$\eta_{\text{LOS}}, \eta_{\text{NLOS}}$	LoS and NLoS additional attenuation factors	$P_{\max}$	Maximum power
$d_{k,AP}$	3D distance between the AP and the k-th IIoT device	$P_{\min}$	Minimum power
f_c	Carrier frequency	s_t, a_t	State and action at t
H	Height of the AP	r_{t+1}	Reward at $t+1$
h_k	Height of the k-th IIoT device	R_t	The discounted accumulated reward
d_{clutter}	Typical clutter size	$Q(\dots)$	Q-function
h_c	Effective clutter height	$s_{n,t}^{(m)}$	State space
r	Clutter density	$\mathcal{A}_u^{(n)}$	Action space
s_c	Common stream	α	Learning rate
$s_{p,k}$	Private stream of the k-th IIoT device	γ	Discount factor
$P_{c,n}^{(m)}$, $P_{p,k,n}^{(m)}$	Powers assigned for the common and private messages	Z	Episode number
x_n	The signal transmitted by the m-th AP to the n-th device cluster	D	Replay memory buffer size
$\mathcal{C}_n^{(m)}$	The set of devices in the n-th device cluster	D_b	Mini-batch size
y_k	The received signal at the k-th IIoT device	$\pi(\cdots)$	Policy of the agent
n_a	Additive white Gaussian noise (AWGN)	θ, θ^-	Weights of the trained and target network
σ^2	The AWGN power spectral density	$\mathcal{L}oss$	Loss function
σ_t, σ_r	Level of HWI at the transmitter and receiver	T_{step}	Time step
$\gamma_{c,k}^{(n)}$, $\gamma_{p,k}^{(n)}$	SINRs for the common and private streams for the k-th device	$\mathcal{Z}^{(m)}$	The set of learning agents associated with the m-th AP
$R_{c,k}^{(n)}$, $R_{p,k}^{(n)}$	Achievable rates for the common and private streams for the k-th device	T	Duration of each RL episode

management (DCPM) framework is proposed to efficiently solve the joint problem. The framework decomposes the joint problem into multiple device clustering and transmit PA sub-problems (one for each AP) and solves them sequentially utilizing only local CSI.

3) We propose a greedy clustering algorithm to solve the device clustering sub-problem for a given transmit PA. In the algorithm, each AP's devices are grouped into multiple clusters to enhance their signal-to-interference-plus-noise ratios (SINRs) in an RSMA setup. We also devise a multi-agent DRL-empowered PA algorithm to solve the PA sub-problem. The proposed algorithm efficiently distributes the AP's transmit power among the IIoT devices in each cluster. Our proposed PA algorithm is innovative because it learns a suitable transmit PA policy without requiring precise and instantaneous knowledge of HWI. This learned policy facilitates the cooperative allocation of the most suitable transmit power across all APs to support dynamic adaptation to changing channel conditions and network variables.
4) Extensive simulations are conducted to verify the influence of several system parameters, including HWI, FBL, block error probability, total available power at the AP, and number of UDs, on the proposed DCPM framework's system capacity. The provided simulation results confirm the framework is able to adapt and effectively learn an appropriate PA strategy for a range of IIoT networking scenarios. The simulation results also validate that the DCPM framework consistently outperforms state-of-the-art transmit PA algorithms and interference management schemes, which attests that our proposed DCPM is able to enhance the IIoT network's resilience to co-channel interference.

The rest of the paper is organized as follows. Section II provides an overview of the system and the problem formulation. The sub-problem solutions are detailed in Sections III and IV. Sections V and VI present the overall algorithm and the simulation results, respectively. Finally, Section VII contains the conclusion. A list of symbols and notations used in this paper is given in Table 1.

II. SYSTEM MODEL AND PROBLEM FORMULATION

A. SYSTEM OVERVIEW

We consider a multi-device downlink IIoT network of M cells each equipped with one AP and K devices randomly distributed within their coverage areas (Fig. 1). For simplicity, let $\mathcal{M} = \{1, 2, \cdots, M\}$ be the set of all APs,

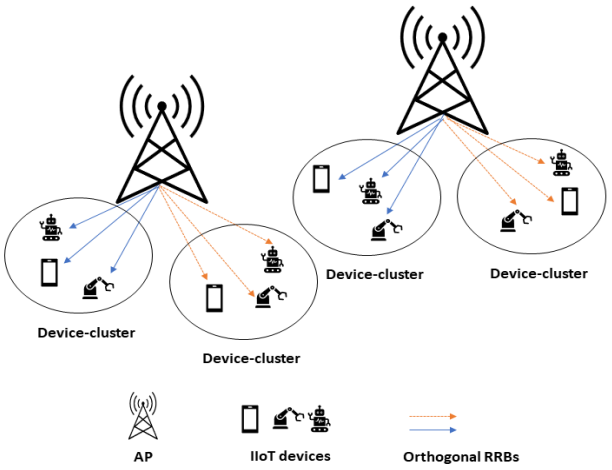


FIGURE 1. An RSMA-enabled downlink IIoT network with two APs, and multiple IIoT devices in each cell.

$\mathcal{K} = \{1, 2, \cdots, K\}$ be the set of IIoT devices in each cell, and $\mathcal{N} = \{1, 2, \cdots, N\}$ be the set of RRBs in each cell. We ignore the inter-AP interference by assigning the neighboring APs orthogonal RRBs and reusing the same RRBs at the far APs. This consideration can be explained by assuming that a total of $S > N$ RRBs are available in the system, and the RRBs are divided into a total of L non-overlapping RRB groups (RRBGs) (each containing N RRBs), which are denoted by $\mathcal{G}_1 = \{1, 2, \cdots, N\}$, $\mathcal{G}_2 = \{N+1, N+2, \cdots, 2N\}$, $\mathcal{G}_3 = \{2N+1, 2N+2, \cdots, 3N\}$, $\cdots$, $\mathcal{G}_L = \{S-N+1, S-N+2, \cdots, S\}$. Each AP is assigned an RRBG in such a way that its L nearest APs are assigned orthogonal RRBGs and its $(L+1)$-th nearest AP reuses its RRBG. Also, the AP's down-tilt angles can be optimized as very little power (interference) radiates from the neighboring cell. As a result, inter-cell interference from distant APs can be ignored due to path loss and shadowing without any performance degradation. In this work, we assume that both RRBG assignments and the down-tilt angles have been pre-optimized to keep inter-AP interference at a minimum.[2] However, in the IIoT network considered, $K \gg N$ (corresponding to a dense network) leads to there being fewer resources available than are needed to accommodate the vast number of devices. We address this challenge by concurrently scheduling multiple IIoT devices over the same RRB. We mitigate the resultant interference and improve system capacity by incorporating a one layer RSMA strategy in each cluster so that multiple devices can be scheduled over the same RRB simultaneously. A two-step procedure is then followed to optimize the RSMA framework's ability to manage interference. In the first step, we divide the K IIoT devices that are in each cell into a total of N clusters, and each cluster is assigned to one orthogonal RRB. In the second step, we execute the DRL algorithm to optimize PA and maximize each cluster's capacity.

[2]The mitigation of both inter-cell and intra-cell interference in an IIoT network is left for future work.

For analytical tractability, the following assumptions are made. **A1:** Each AP is aware of the CSI of the devices in its cell. However, APs do not have any information about the presence of instantaneous HWI-induced signal distortions at the devices. **A2:** The clusters are non-overlapping to ensure that inter-cluster interference is negligible. **A3:** Each device can be associated with a maximum of one AP and one cluster. **A4:** The time horizon is decomposed into multiple non-overlapping time slots (TSs). We consider all communication links to exhibit quasi-static channel fading, i.e., the CSI of the links remains constant in each TS and can change independently from one TS to the next.

B. CHANNEL MODEL

We model the downlink channel gain from the m-th AP to the k-th IIoT device in the n-th cluster the same way as it is modeled in [29], where each channel is subjected to large-scale fading $\beta_{k,n}^{(m)}$ and small-scale block Rayleigh fading $h_{k,n}^{(m)}$. In the t-th TS, the channel gain is expressed as:

$$g_{k,n}^{(m)}(t) = |h_{k,n}^{(m)}(t)|^2 \beta_{k,n}^{(m)}. \tag{1}$$

According to the Jakes fading model [30], $h_{k,n}^{(m)}(t)$ can be expressed as a first-order complex Gauss-Markov process:

$$h_{k,n}^{(m)}(t) = \rho h_{k,n}^{(m)}(t-1) + \sigma', \tag{2}$$

where $\rho = J_0(2\pi f_d T_s)$ is the correlation coefficient between two TSs, with $J_0(.)$ being the zeroth-order Bessel function, f_d being the Doppler frequency, and T_s being the duration of each TS, and σ' is a random variable with a distribution $\sigma' \sim \mathcal{CN}(0, 1-\rho^2)$. The large-scale fading includes both path loss and shadowing. We consider the path loss model proposed by 3GPP [31] for a scenario involving an indoor factory with space clutter and a high BS height (InF-SH) [32]. The average path loss (in dB) between the AP and the k-th IIoT device is given by:

$$PL = Pr_{LoS} PL_{LoS} + Pr_{NLoS} PL_{NLoS}, \tag{3}$$

where Pr_{LoS} and Pr_{NLoS} are the probability of having a line-of-sight (LoS) and a non-line-of-sight (NLoS) link, respectively, between the AP and the k-th IIoT device. PL_{LoS} and PL_{NLoS} represent the path loss (in dB) between the AP and the k-th IIoT device for the LoS and NLoS links, respectively. They are expressed as [31, Table 7.4.1-1]:

$$PL_{LoS} = 31.84 + 21.5\, log_{10}(d_{k,AP}) + 19\, log_{10}(f_c) + \eta_{LoS} \tag{4}$$

and

$$PL_{NLoS} = \max(32.4 + 23\, log_{10}(d_{k,AP}) + 20\, log_{10}(f_c) + \eta_{NLoS}, PL_{LoS}), \tag{5}$$

where $d_{k,AP} = \sqrt{(dx_k - X)^2 + (dy_k - Y)^2 + (h_k - H)^2}$ represents the 3D distance between the AP and the k-th IIoT device, with (dx_k, dy_k) being the position of the k-th IIoT device, (X, Y) being the position of the AP, H denoting the

height of the AP, and h_k denoting the height of the k-th IIoT device, f_c denotes the carrier frequency and η_{LoS} and η_{NLoS} represent additional attenuation factors due to the LoS and NLoS connections, respectively.

The probability values can be obtained from [31, Table 7.4.2-1]:

$$Pr_{LoS} = e^{(\frac{-d}{k_{subsce}})} \tag{6}$$

and

$$Pr_{NLoS} = 1 - Pr_{LoS}, \tag{7}$$

where

$$k_{subsce} = -\frac{d_{clutter}}{ln(1-r)} \frac{H - h_k}{h_c - h_k}, \tag{8}$$

with $d_{clutter}$ being the typical clutter size, h_c representing the effective clutter height, and r representing the clutter density.

C. RSMA STRATEGY

We adopt the well-known one-layer RSMA[3] strategy that splits the received data of the devices in a cluster into common and private parts. All the receivers' common parts are combined into a common message, which is encoded in a common stream s_c, while the private parts are encoded in a set of private streams $\{s_{p,k}\}$, one per device [5]. The s_c and $\{s_{p,k}\}$ streams are then linearly precoded and transmitted together over the same RRB. The signal transmitted by the m-th AP to the n-th device cluster is expressed as:

$$x_n = \sqrt{P_{c,n}^{(m)}}\, s_c + \sum_{n \in \mathcal{C}_n^{(m)}} \sqrt{P_{p,k,n}^{(m)}}\, s_{p,k}, \tag{9}$$

where $\mathcal{C}_n^{(m)}$ is the set of devices in the n-th device cluster and $|\mathcal{C}_n^{(m)}|$ is the cardinality of the $\mathcal{C}_n^{(m)}$ set, $P_{c,n}^{(m)}$ and $\{P_{p,k,n}^{(m)}\}$ denote the transmission power assigned to the common and private messages, respectively, and $P_n = [P_{c,n}^{(m)}, P_{p,1,n}^{(m)}, .., P_{p,|\mathcal{C}_n^{(m)}|,n}^{(m)}]$ denotes the power profile. The signal received at the k-th IIoT device in the t-th TS is given by:

$$y_k(t) = \sqrt{g_{k,n}^{(m)}(t)}(\, x_n + z) + n_a, \tag{10}$$

where z represents the signal distortion caused by HWI and n_a is AWGN with variance σ^2. In this article, we consider the linear distortion introduced by the imperfect power amplifier. According to [33], $z \sim \mathcal{CN}\left(0, (\sigma_t^2 + \sigma_r^2)P_{total,n}\right)$, where $P_{total,n}$ is the total power provided by the AP to the n-th cluster, and σ_t and σ_r denote the level of HWI at the transmitter and receiver, respectively.

In the RSMA strategy considered, each receiver first decodes the common messages by treating the interference from all the private streams as noise. Each receiver then removes the interference from the decoded common stream by applying SIC and decodes its private message while treating the other devices' private messages as noise. Thus, the SINRs of the common and private streams for the k-th IIoT device, $\forall k \in \mathcal{C}_n^{(m)}$, are expressed as:

$$\gamma_{c,k}^{(n)} = \frac{P_{c,n}^{(m)}\, g_{k,n}^{(m)}(t)}{\sum_{k=1}^{|\mathcal{C}_n^{(m)}|} P_{p,k,n}^{(m)}\, g_{k,n}^{(m)}(t) + I_{HWI} + \sigma^2} \tag{11}$$

and

$$\gamma_{p,k}^{(n)} = \frac{P_{p,k,n}^{(m)}\, g_{k,n}^{(m)}(t)}{\sum_{\substack{k'=1 \\ k' \neq k}}^{|\mathcal{C}_n^{(m)}|} P_{p,k',n}^{(m)}\, g_{k,n}^{(m)}(t) + I_{SIC} + I_{HWI} + \sigma^2}, \tag{12}$$

respectively, where $I_{HWI} = g_{k,n}^{(m)}(t)(\sigma_t^2 + \sigma_r^2)P_{total,n}$ represents the interference caused by HWI-induced signal distortions and $I_{SIC} = \theta_c g_{k,m,n}(t) P_{c,n}^{(m)}$ is the interference caused by applying SIC to decode the common message, with $\theta_c \in [0, 1]$ representing the SIC error coefficient. $\theta_c = 0$ and $\theta_c = 1$ represent the scenarios with ideal SIC and no SIC, respectively.

The common stream's achievable rate for the k-th IIoT device when considering FBL is expressed as [34]:

$$\mathrm{R}_{c,k}^{(n)} = \log_2(1 + \gamma_{c,k}^{(n)}) - \frac{\mathrm{Q}^{-1}(\epsilon)}{\sqrt{n_b}} \sqrt{\mathrm{V}(\gamma_{c,k}^{(n)})}, \tag{13}$$

where $\mathrm{V}(\gamma_{c,k}^{(n)})$ is defined as:

$$\mathrm{V}(\gamma_{c,k}) = 1 - \frac{1}{(1 + \gamma_{c,k}^{(n)})^2}, \tag{14}$$

and n_b and ϵ represent the blocklength and block error probability, respectively. Moreover, $Q^{-1}(.)$ is the inverse of the Gaussian Q-function $Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-\frac{t^2}{2}} dt$. It should be noted that the common message needs to be decoded by each device, and, therefore, the common stream's achievable rate is equal to the lowest of all the receivers' common rates:

$$\mathrm{R}_c^{(n)} = \min[\, \mathrm{R}_{c,1}^{(n)}, \cdots, \mathrm{R}_{c,|\mathcal{C}_n^{(m)}|}^{(n)}]. \tag{15}$$

Once the interference has been removed from the common message using SIC, each device decodes its private message by considering the interference from the other devices in the device cluster as noise. The private stream's achievable rate for the k-th IIoT device, $\forall k \in \mathcal{C}_n^{(m)}$, is obtained as:

$$\mathrm{R}_{k,p}^{(n)} = \log_2(1 + \gamma_{p,k}^{(n)}) - \frac{\mathrm{Q}^{-1}(\epsilon)}{\sqrt{n_b}} \sqrt{\mathrm{V}(\gamma_{p,k}^{(n)})}. \tag{16}$$

We consider that each AP has a total of N RRBs, and consequently, each AP has a maximum of N IIoT device clusters. We denote the m-th AP's device clusters by the sets $\mathcal{C}_1^{(m)}, \mathcal{C}_2^{(m)}, \cdots, \mathcal{C}_N^{(m)}$. The achievable sum rate of the n-th device cluster in the m-th AP is obtained as

$$\mathrm{R}_{C_n}^{(m)} = \frac{B}{MN} \left(\mathrm{R}_c^{(n)} + \sum_{k \in \mathcal{C}_n^{(m)}} \mathrm{R}_{k,p}^{(n)} \right), \tag{17}$$

[3]It is noteworthy that our DCPM framework is generic and can be applied to IIoT networks utilizing the NOMA scheme. While NOMA-based DCPM has the potential to achieve higher system capacity compared to RSMA-based DCPM, its implementation is challenging due to the need for multiple SIC operations at user devices.

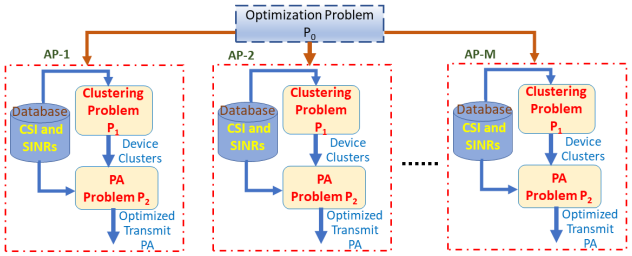


FIGURE 2. **Proposed DCPM framework.**

where B is the total system bandwidth, which is divided equally among all the APs and their device clusters. The total achievable sum rate of the m-th AP is obtained as:

$$\mathrm{R}_{total}^{(m)} = \sum_{n=1}^{N} \mathrm{R}_{C_n}^{(m)}. \tag{18}$$

D. PROBLEM FORMULATION

The joint clustering and transmit PA optimization problem is formulated as (19), shown at the bottom of the next page. Constraints (C1) and (C2) imply that the transmit power of the common and private messages must be bounded by P_{min} and P_{max}, which represent the minimum and maximum transmit power limits of an IIoT device, respectively. Constraint (C3) implies that the total power provided by an AP must be less than the total power an AP can achieve. Constraint (C4) implies that each device cluster can support a maximum of K_D IIoT devices. Constraint (C5) ensures that the device clusters belonging to the same AP do not overlap.

Proposition 1: P0 is an NP-hard optimization problem.

Proof: The proof is provided in Appendix.

Since P0 is an NP-hard and mixed-integer non-linear programming problem, it is computationally intractable to obtain its global optimal solution using standard optimization techniques. Therefore, we propose to mitigate this computational intractability with a sequential optimization framework. The framework involves fixing one block of optimization variables so that P0 can be decomposed into two sub-problems, namely, a device clustering one and a PA one. If we assume the PA is fixed, i.e., $P_{c,n}^{(m)} = P_0$ and $P_{p,k,n}^{(m)} = P_0$, the device clustering sub-problem can be formulated as:

$$\begin{aligned} \text{P1:} \quad & \max_{\{\mathcal{C}_n^{(m)}\}} \sum_{m=1}^{M} R_{total}^{(m)} \\ & \text{s.t. C4, C5.} \end{aligned} \tag{20}$$

Meanwhile, if we consider that each AP has a given set of device clusters, the PA sub-problem can be formulated as:

$$\begin{aligned} \text{P2:} \quad & \max_{\mathbf{P}} \sum_{m=1}^{M} R_{total}^{(m)} \\ & \text{s.t. C1, C2, C3.} \end{aligned} \tag{21}$$

We propose to use a distributed optimization framework entitled the DCPM framework, which is illustrated in Fig. 2, to sequentially solve P1 and P2. Each AP consists of a database that stores the instantaneous local CSI and past SINRs of its associated devices. Each AP obtains its set of device clusters by solving P1 using the local CSI extracted from its database. Then, each AP obtains its device clusters' transmit PA by solving P2. We built this optimization framework by first developing a greedy clustering algorithm to solve P1 (see Section III). We then developed a multi-agent DRL approach to solve P2 (see Section IV). The overall DCPM algorithm leverages the greedy clustering and DRL approaches to solve P0 and is presented in Section V.

Remark 1: The motivation for using the DRL approach to solve P2 is as follows. P2 is a non-convex and computationally intractable optimization problem that becomes maximizing a sum-of-function-ratios, which is NP-complete. Solving P2 using a conventional optimization technique that relies on satisfying the Karush-Khun-Tucker (KKT) conditions is practically challenging. Conventional optimization techniques require precise knowledge of devices' instantaneous SINRs to derive the KKT conditions. However, it is practically unfeasible to know the exact value of HWI-induced signal distortions at the devices, as these distortions vary randomly from device to device. We specifically consider a practical setting in which the devices' exact non-linearity models are unknown to the network controller. The SIC error coefficients and the interference caused by imperfect SIC at the receivers are also unknown to the network controller. Therefore, even though the instantaneous CSI is assumed to be available, the devices' *exact instantaneous* SINR values are not available to the network controller in practice. Such a fact renders conventional optimization techniques ineffective at solving P2. In contrast, the DRL approach learns a suitable policy for allocating transmit power based on the devices' historical SINRs and current CSI.[4] The DRL approach neither requires precise knowledge of the devices' instantaneous SINRs nor the exact value of HWI-induced signal distortions to do this. Moreover, once the DRL agent is fully trained, it can also quickly solve P2 and thus converges in much less time than the conventional optimization method considered, which requires several iterations to converge. This efficiency is particularly advantageous in dynamic environments where quick decision-making is crucial. Therefore, the DRL approach is a robust and scalable means of solving P2.

Remark 2: The optimality of the overall solution can be improved by alternately optimizing P1 and P2 instead of solving them only once sequentially. However, this iterative optimization approach increases the computational complexity, signaling overhead, and solution time. In this approach, device clusters need to be updated at each iteration based on the transmit powers obtained in the previous iteration. Afterward, the SINRs (with HWI-induced distortions) must

[4]We assume that the IIoT devices provide accurate received SINR feedback at the end of each TS. Hence, the AP knows the SINRs of the common and private streams of the associated IIoT devices from the previous TS. AP needs to compute PA decisions for the current TS using that information.

be estimated for the newly formed device clusters. Finally, the PA needs to be updated based on the new SINR values. It is practically non-trivial to perform these three steps iteratively several times within the channel coherence time. A sequential optimization approach is therefore adopted to solve P0 in a computationally efficient manner.

III. SOLUTION TO P1: DEVICE CLUSTERING ALGORITHM

A. ALGORITHM DEVELOPMENT

Since there is no interference among the APs, P1 can be decomposed into a total of M independent optimization problems, one for each AP. The device clustering sub-problem for the m-th AP, $\forall m \in \mathcal{M}$, is formulated as:

$$\text{P1.1:} \quad \max_{\{\mathcal{C}_n^{(m)}\}} \sum_{n=1}^{N} R_{\mathcal{C}_n}^{(m)} \quad \text{s.t. C4, C5.} \tag{22}$$

The computational complexity involved in solving P1.1 optimally can be obtained from the Stirling number of the second kind, which is defined as the number of ways K different objects (i.e., IIoT devices) can be partitioned into N non-overlapping device clusters. The Stirling number of the second kind is expressed as:

$$\left\{ {K \atop N} \right\} \triangleq \frac{1}{N!} \sum_{j=0}^{N} (-1)^{N-j} \binom{N}{j} j^K. \tag{23}$$

For $K \gg N$, the Stirling number of the second kind is approximated as $\mathcal{O}(N^{K-N})$ [35]. Clearly, the computational complexity involved in solving P1.1 optimally increases exponentially with the number of IIoT devices, and, as a result, arriving at a theoretically optimal solution to P1.1 is computationally intractable for practical IIoT networks. However, it is possible to develop a sub-optimal yet efficient algorithm to solve P1.1 by exploiting the problem's characteristics. To this end, we assume that certain clusters of $K' \geq 1$ devices are known.[5] Doing so reduces P1.1 to an optimization problem that involves assigning the remaining $(K - K')$ devices to N RRBs in such a way as to maximize the total sum rate. The optimal device–RRB assignment needs to maximize (a) the device's SINR and (b) the minimum channel gain of the devices in the cluster. An efficient device–RRB assignment can be obtained by selecting the remaining $(K - K')$ devices in a suitable order and sequentially assigning them to their most appropriate device cluster.[6]

In light of the above discussion, P1.1 can be alternatively solved by finding the clusters of IIoT devices that maximize the clustered devices' total SINRs, which are the sum of the SINRs of their common and private messages. Without loss of generality, we assume that the k-th IIoT device is part of the n-th cluster. The total SINR of the k-th IIoT device is obtained as $\overline{\gamma_k^{(n)}} = \gamma_{c,k}^{(n)} + \gamma_{p,k}^{(n)}$, where

$$\gamma_{c,k}^{(n)} = \begin{cases} \dfrac{P_o}{|\mathcal{C}_n^{(m)}| P_o + \dfrac{\sigma^2}{\min\{\min_{k' \in \mathcal{C}_n^{(m)}} g_{k',n}^{(m)}, g_{k,n}^{(m)}\}}} & \text{if } |\mathcal{C}_n^{(m)}| \geq 1 \\ \dfrac{P_o}{\dfrac{\sigma^2}{g_{k,n}^{(m)}}} & \text{if } |\mathcal{C}_n^{(m)}| = 0 \end{cases} \tag{24}$$

and

$$\gamma_{p,k}^{(n)} = \begin{cases} \dfrac{P_o}{\left(|\mathcal{C}_n^{(m)}| - 1 + \theta_c\right) P_o + \dfrac{\sigma^2}{g_{k,n}^{(m)}}} & \text{if } |\mathcal{C}_n^{(m)}| \geq 1 \\ \dfrac{P_o}{\dfrac{\sigma^2}{g_{k,n}^{(m)}}} & \text{if } |\mathcal{C}_n^{(m)}| = 0, \end{cases} \tag{25}$$

where P_o is the fixed transmit power of both the device's common and all its private messages.[7] Let us introduce the variable $z_{k,n}^{(m)}$, which equals 1 if $k \in \mathcal{C}_n^{(m)}$ and 0 otherwise. The modified device clustering sub-problem can be formulated as:

$$\text{P1.2:} \quad \max_{\{z_{k,n}^{(m)} \in \{0,1\}} \sum_{n=1}^{N} \sum_{k=1}^{K} z_{k,n}^{(m)} \left(\gamma_{c,k}^{(n)} + \gamma_{p,k}^{(n)}\right) \quad \text{s.t.} \begin{cases} \sum_{k=1}^{K} z_{k,n}^{(m)} \leq K_D, & \forall n \in \mathcal{N} \\ \sum_{n=1}^{N} z_{k,n}^{(m)} = 1, & \forall k \in \mathcal{K}. \end{cases} \tag{26}$$

[5] One can obtain initial set device clusters by assigning K' devices to their most suitable RRBs such that each device is associated with only one RRB and each RRB is associated with only one device.

[6] This idea is inspired by the greedy approach used to solve the generalized assignment problem (GAP) and exploits the similarity that exists between P1.1 and a GAP [36].

[7] Recall that instantaneous information about HWI-induced signal distortions and SIC error coefficients is not available at the devices. Consequently, interference caused by HWI-induced signal distortions and imperfect SIC is ignored in (24) and (25), i.e., during the device clustering stage.

$$\text{P0:} \quad \max_{\boldsymbol{P}, \{\mathcal{C}_n^{(m)}\}} \sum_{m=1}^{M} R_{total}^{(m)}$$

$$\text{s.t.} \begin{cases} \text{C1: } P_{min} \leq P_{c,n}^{(m)} \leq P_{max}, & \forall n \in \mathcal{N}, m \in \mathcal{M} \\ \text{C2: } P_{min} \leq P_{p,k,n}^{(m)} \leq P_{max}, & \forall k \in \mathcal{C}_n^{(m)}, n \in \mathcal{N}, m \in \mathcal{M} \\ \text{C3: } \sum_{n=1}^{N} \sum_{k \in \mathcal{C}_n^{(m)}} (P_{p,k,n}^{(m)} + P_{c,n}^{(m)}) \leq P_t, & \forall k \in \mathcal{C}_n^{(m)}, n \in \mathcal{N}, m \in \mathcal{M} \\ \text{C4: } |\mathcal{C}_n^{(m)}| \leq K_D, & \forall n \in \mathcal{N}, m \in \mathcal{M} \\ \text{C5: } \mathcal{C}_n^{(m)} \cap \mathcal{C}_t^{(m)} = \varnothing, & \forall n, t \in \mathcal{N}, n \neq t, m \in \mathcal{M} \end{cases} \tag{19}$$

We develop a greedy clustering algorithm, which is summarized in Algorithm 1, to solve P1.2. Algorithm 1 iteratively selects a device and assigns it to an RRB (i.e., device cluster). It maintains two sets – (i) $\mathbf{UM}_U$, the set of unassigned IIoT devices; and (ii) $\mathbf{UM}_R$, the set of RRBs that can accommodate at least one IIoT device. Furthermore, it keeps track of the number of IIoT devices that are assigned to the n-th RRB, which is denoted by D_n. When a device is associated with an RRB, it simultaneously obtains a profit by increasing its own SINR gain and incurs a cost by either reducing the minimum channel gain of its cluster (and thereby, reducing the common message rate) or reducing the SINRs of other devices' private messages. Hence, attention must be paid to the order in which devices are selected and assigned to their most suitable clusters, as it can enhance the net profit. To select the most suitable device cluster for a given IIoT device at each iteration of Algorithm 1, we consider the following score

$$\Delta_k = \overline{\gamma_k^{(n_k^*)}} - \max_{\substack{n\in\mathbf{UM}_R \\ n\neq n_k^*}} \overline{\gamma_k^{(n)}}, \tag{27}$$

where n_k^* denotes the index of the k-th IIoT device's most suitable cluster, and the first and second terms of (27) provide the k-th IIoT device's highest and second-highest SINRs, respectively, obtained from the available device clusters in $\mathbf{UM}_R$. This score captures the net profit that would be obtained from assigning the k-th IIoT device to its most suitable device cluster. In other words, it provides a measure of the importance of including the IIoT device in its most suitable cluster.

The key steps in Algorithm 1 are summarized as follows. Step 2 initializes the set $\mathbf{UM}_U$, the variable D_n and the device cluster sets $\{\mathcal{C}_n\}$, $\forall n \in \mathcal{N}$. Steps 4-11 iteratively compute the score defined in (27) for each unassigned device in $\mathbf{UM}_U$ when it is paired with each available device cluster in $\mathbf{UM}_R$. Step 12 selects the device that achieves the highest score. Step 13 assigns the selected device to its most suitable device cluster in $\mathbf{UM}_R$, and updates $\mathbf{UM}_U$ and $\{D_n\}$. The aforementioned steps continue until either $\mathbf{UM}_U$ or $\mathbf{UM}_R$ is empty.

B. COMPUTATIONAL COMPLEXITY

We first determine the computational complexity of executing a single loop iteration of Algorithm 1, i.e., Steps 3-13. The worst-case computational complexity of executing Steps 5-11 is $\mathcal{O}(|\mathbf{UM}_U|N)$, where $|\mathbf{UM}_U|$ denotes the cardinality of $\mathbf{UM}_U$. The computational complexity of executing Step 12 is $\mathcal{O}(|\mathbf{UM}_U|)$, and that of the remaining steps is $\mathcal{O}(1)$. The total computational complexity of a single loop iteration of Algorithm 1 is therefore $\mathcal{O}(|\mathbf{UM}_U|N + |\mathbf{UM}_U|)$. Note that since Algorithm 1 sequentially clusters a device and removes it from $\mathbf{UM}_U$, the variable $|\mathbf{UM}_U|$ is reduced by 1 at each iteration. In the end, the overall computational complexity of Algorithm 1 is obtained as $\mathcal{O}\left(\sum_{l=1}^{K}(Nl + l)\right) \approx \mathcal{O}\left(K^2N\right)$.

Algorithm 1 Proposed IIoT Device Clustering Algorithm for the m-th AP

1: **Input:** Number of IIoT devices K, number of RRBs N, channel gains $\{g_{k,n}^{(m)}\}$, fixed PAs $P_{c,n}^{(m)} = P_0$ and $P_{p,k,n}^{(m)} = P_0$, $\forall k \in \mathcal{C}_n^{(m)} n \in \mathcal{N}, m \in \mathcal{M}$.
2: **Initialize:** $\mathbf{UM}_U = \{1, 2, \cdots, K\}$, $D_n = 0$, $\forall n \in \{1, 2, \cdots, N\}$; IIoT device clusters $\mathcal{C}_n = \varnothing$, $\forall n \in \{1, 2, \cdots, N\}$.
3: **repeat**
4: Find $\mathbf{UM}_R = \{n \in \{1, 2, \cdots, N\} | D_n < K_D\}$.
5: **for** $k = 1 : |\mathbf{UM}_U|$ **do**
6: **for** $n = 1 : |\mathbf{UM}_R|$ **do**
7: Calculate $\overline{\gamma_k^{(n)}} = \gamma_{c,k}^{(n)} + \gamma_{p,k}^{(n)}$ using (24) and (25).
8: **end for**
9: Find $n_k^* = \arg\max_{n\in\mathbf{UM}_R} \overline{\gamma_k^{(n)}}$.
10: Calculate $\Delta_k = \overline{\gamma_k^{(n_k^*)}} - \max_{\substack{n\in\mathbf{UM}_R \\ n\neq n_k^*}} \overline{\gamma_k^{(n)}}$.
11: **end for**
12: Find $k^* = \arg\max_{k\in\mathbf{UM}_U} \Delta_k$.
13: Assign $\mathcal{C}_{n_k^*} \rightarrow \mathcal{C}_{n_k^*} \cup \{k^*\}$; update $D_{n_k^*} = D_{n_k^*} + 1$ and $\mathbf{UM}_U \rightarrow \mathbf{UM}_U \setminus \{k^*\}$.
14: **until** $\mathbf{UM}_U = \varnothing$ or $\mathbf{UM}_R = \varnothing$
15: **Output:** IIoT device clusters $\mathcal{C}_1, \mathcal{C}_2, \cdots, \mathcal{C}_N$.

IV. SOLUTION TO P2: DRL-EMPOWERED PA ALGORITHM

A. DEEP Q-NETWORK OVERVIEW

We develop a DRL-empowered PA algorithm by utilizing the deep Q-network (DQN), which is a single-agent RL algorithm that performs Q-learning in high-dimensional state spaces and complex environments. The fundamental building block of DQN is a deep neural network (DNN) that serves to approximate the agent's optimal action value function in a given environment. DQN learns to make decisions by iteratively updating the Q-values of state–action pairs based on the rewards that are received from the environment [37].

We first provide a brief overview of the Q-learning scheme. Let S denote a set of possible states and A denote a set of discrete actions. At TS t, the RL agent observes the state of its environment ($s_t \in S$) and takes an action ($a_t \in A$) based on a certain policy $\pi(s/a)$. Once the agent takes an action, the environment moves to the next state, s_{t+1}. The agent then receives a reward r_{t+1} that describes how beneficial the action taken at state s_t was. The tuple $(s_t, a_t, r_{t+1}, s_{t+1})$ forms an experience that describes the agent's interaction with the environment. The agent's main goal is to maximize the discounted accumulated reward, which is expressed as $R_t = \sum_{\tau=1}^{\infty} \gamma^\tau r_{t+\tau}$, where γ is the discount factor, and $r_{t+\tau}$ is the value of the reward at TS $t + \tau$. The agent needs to learn an optimal policy, $\pi^*(s, a)$, that maps the state space to the action space. Q-learning is based on the action value function, which represents the return that is expected for taking action a in state s using policy π. It is defined as $Q_\pi(s, a) = \mathbb{E}(R_t | s_t = s, a_t = a, \pi)$. The optimal action value function satisfies the Bellman optimality equation as follows:

$$Q^*(s, a) = \mathbb{E}(r_{t+1} + \gamma \max_{a'} Q^*(s_{t+1}, a') | s_t = s, a_t = a), \tag{28}$$

where $Q^*(s,a) = \max_\pi Q_\pi(s,a)$ represents the maximum action value that can be obtained by following any policy. Clearly, the optimal action selection policy in a given state s is defined as:

$$\pi^*(s,a) = \begin{cases} 1, & a = \arg\max_{a \in A} Q^*(s,a) \\ 0, & \text{otherwise.} \end{cases} \tag{29}$$

Q-learning based on (30) is guaranteed to converge as long as the Markov state transition property holds. However, for a complex environment with a continuous state space, Q-learning requires a large amount of storage capacity to store the action value functions of a large number of states and an (infinitely) long time to visit all the states. As a result, despite the fact that it converges, Q-learning is impractical for complex environments.

DQN overcomes Q-learning's limitations by employing a DNN, which inputs the state and outputs the Q-function $q(s,a;\theta)$ for each possible action, where θ denotes the weight of the DNN. Essentially, DQN learns to predict the action values of an unseen state using the DNN's inference capability. When training a DQN, the agent takes actions according to the ϵ-greedy policy for dealing with the exploration-exploitation dilemma [38]. Exploration refers to trying out different actions to learn more about the environment. In contrast, exploitation refers to using the learned policy to make decisions that maximize the reward. Like the Q-learning algorithm, the Q-function's values are obtained through trial and error and are updated as:

$$q(s_t,a_t;\theta) \leftarrow (1-\alpha) q(s_t,a_t;\theta) + \alpha[r_{t+1} + \gamma \max_{a'} q(s_{t+1},a';\theta)], \tag{30}$$

where α represents the learning rate. DQN is an off-policy RL algorithm that stores the previous experience in a replay memory D. Then, a mini-batch from this memory is sampled to train the DQN by minimizing the mean square loss function. If the same DQN network is used to provide the ground truth for the loss function and update parameters, then the training of DQN becomes highly unstable. We address this by using a quasi-static target Q-network with parameter θ^- to predict the Q-function's target value (i.e., the ground truth). The loss function for training the DQN is constructed as:

$$\mathcal{L}oss = \frac{1}{2} \sum_{(s,a,r,s') \in D} (r' - q(s,a;\theta))^2, \tag{31}$$

where $r' = r + \gamma \max_{a'} q(s',a';\theta^-)$ is the Q-function's target value. To minimize (31), we iteratively update θ by sampling a mini-batch of experiences from D and using stochastic gradient descent as follows:

$$\theta \leftarrow \theta - [r' - q(s,a;\theta)] \bigtriangledown q(s,a;\theta). \tag{32}$$

The target Q-network is periodically updated by copying parameters from the trained DQN. The DQN rapidly converges to a suitable parameterized policy for a Markov decision process with a continuous state space [29].

B. THE PROPOSED SOLUTION

In this work, we develop a DQN-based multi-agent DRL algorithm to optimize PA at each AP. The main constituents of our proposed DRL scheme are defined as follows:

1) **Agents:** Each device cluster is a learning agent. The set of learning agents associated with the m-th AP is denoted by $\mathcal{Z}^{(m)} = \{1, 2, \cdots, n\}$, and thus $\mathcal{Z} = \cup_{m=1}^{M} \mathcal{Z}^{(m)}$ represents an overall set of learning agents.
2) **States:** The state space refers to the set of all possible states that can apply to the environment during the learning process. In our context, the state space of the n-th agent in the m-th AP comprises three key components:
   (i) The set of local CSI at the t-th TS $(g_{1,n}^{(m)}, \ldots, g_{|\mathcal{C}_n^{(m)}|,n}^{(m)})$,
   (ii) The set of SINRs of the common streams at the $(t-1)$-th TS $(\gamma_{c,1}^{(n)}, \gamma_{c,2}^{(n)}, \ldots, \gamma_{c,|\mathcal{C}_n^{(m)}|}^{(n)})$,
   (iii) The set of SINRs of the private streams at the $(t-1)$-th TS $(\gamma_{p,1}^{(n)}, \gamma_{p,2}^{(n)}, \ldots, \gamma_{p,|\mathcal{C}_n^{(m)}|}^{(n)})$.
   Therefore, the state space of the n-th agent in the m-th AP, $\forall n \in \mathcal{Z}^{(m)}$, at the t-th TS is formally defined as:
   $$s_{n,t}^{(m)} = \{(g_{1,n}^{(m)}, \ldots, g_{|\mathcal{C}_n^{(m)}|,n}^{(m)}), (\gamma_{c,1}^{(n)}, \gamma_{c,2}^{(n)}, \ldots, \gamma_{c,|\mathcal{C}_n^{(m)}|}^{(n)}), (\gamma_{p,1}^{(n)}, \gamma_{p,2}^{(n)}, \ldots, \gamma_{p,|\mathcal{C}_n^{(m)}|}^{(n)})\}. \tag{33}$$
   At every TS, the state space provides information about the current state of the environment, including the instantaneous channel gains. Additionally, feedback is provided on the outcome of previous actions through the SINRs of the previous TS(s). This information helps the agents learn a suitable policy while also keeping track of the history of the effectiveness of their chosen actions over the fading channels.
3) **Actions:** Each action taken by each agent represents a set of discrete power levels between P_{min} and P_{max} for the agent's associated IIoT devices. The action space of the u-th UD in the n-th device cluster is defined as:
   $$\mathcal{A}_u^{(n)} = \{P_{min}, \frac{P_{max}}{NP-1}, \frac{2P_{max}}{NP-1}, \ldots, P_{max}\}, \tag{34}$$
   where NP is the total number of discrete power levels. Note that for a device cluster of K_D IIoT devices, a total of (K_D+1) discrete power levels (i.e., one level for the common message and K_D levels for the private messages) are selected. Thus, the action space of the n-th agent in the m-th AP, $\forall n \in \mathcal{Z}^{(m)}$ and $\forall m \in \mathcal{M}$, is defined as $\mathcal{A}_n = \mathcal{A}_1^{(n)} \times \mathcal{A}_2^{(n)} \times \cdots \mathcal{A}_{K_D}^{(n)} \times \mathcal{A}_{K_D+1}^{(n)}$, where $\mathcal{A}_i^{(n)}, i = 1 \cdots K_D$, denotes the set of discrete power levels for the i-th private message and $\mathcal{A}_{K_D+1}^{(n)}$ denotes the discrete power level for the common message.
4) **Reward:** The agents (i.e., device clusters) select transmit power levels for the IIoT devices to maximize the sum rate. We consider a collaborative learning framework in which all the agents jointly maximize the AP's total sum capacity. Accordingly, the reward function of the n-th agent associated with the m-th AP, $\forall m \in \mathcal{M}$,

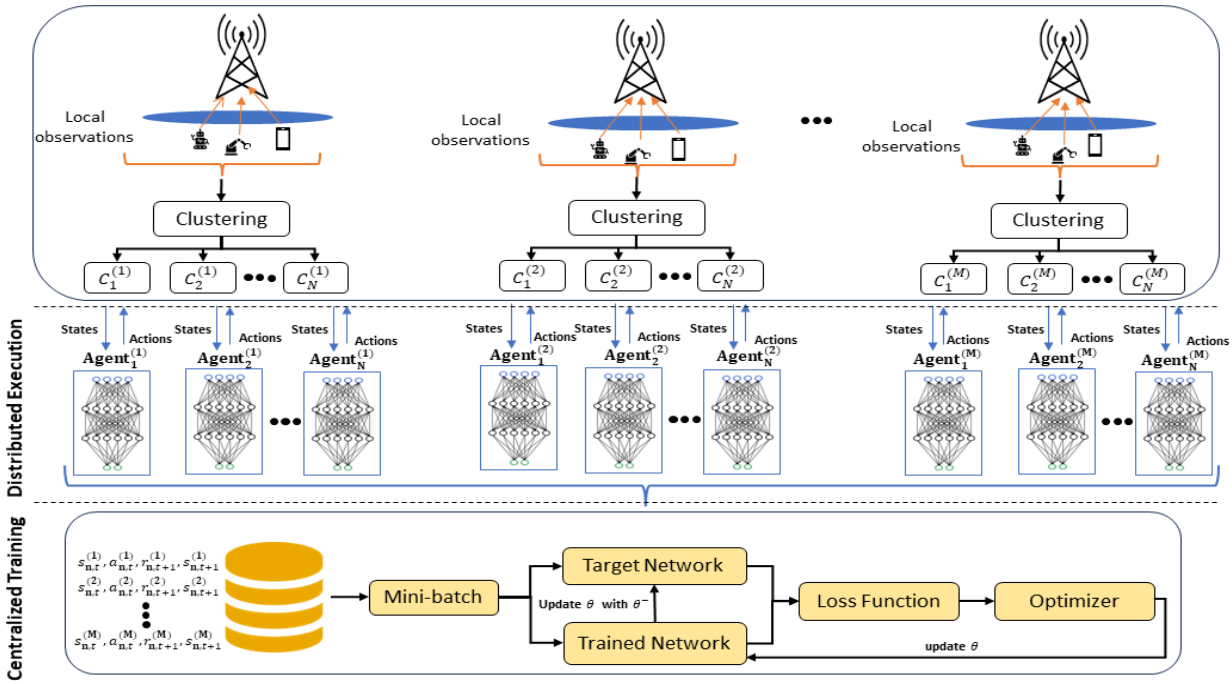


FIGURE 3. Illustration of the proposed multi-agent DRL algorithm with centralized training and distributed execution.

is defined as $\hat{R}_n^{(m)} = \mathrm{R}_{total}^{(m)}, \forall n \in \mathcal{Z}^{(m)}$, where $\mathrm{R}_{total}^{(m)}$ is defined in (18).

Fig. 3 shows the architecture of our proposed multi-agent DRL framework. The environment comprises M APs that each host K IIoT devices randomly distributed within their converge area. The position of each IIoT device changes from one TS to the next. Therefore, in each TS, the proposed solution begins by clustering the devices associated with each AP using Algorithm 1 to obtain, for example, the set of clusters $\{C_1^{(m)}, C_2^{(m)}, \cdots, C_N^{(m)}\}$ for the m-th AP, $\forall m \in \mathcal{M}$. These device clusters are our multi-agent DRL framework's learning agents. In each TS, these RL agents observe the states of the clustered IIoT devices and then allocate the AP's transmit power levels to the clustered devices. The RL agents can be considered independent DQN agents and trained by employing the DQN framework discussed in Section IV. A. However, extending the single-agent DRL framework to a multi-agent scenario in such a straightforward way entails the following two challenges. First, due to the total transmit power constraint C3, an agent's state transition depends on the policies adopted by the other agents belonging to the same AP. Since the agents continuously update their policies during training, the learning environment becomes non-stationary and violates the Markov properties of state transitions, which results in instability in DQN training. Second, a total of MN DQNs need to be trained simultaneously since each agent's policy is provided by an independent DQN, and this significantly limits the scalability of the learning framework. We overcome these challenges by employing a centralized training and distributed execution (CTDE) approach with parameter sharing [29]. In this approach, each agent maintains an identical DQN. The DQN is centrally trained by a centralized controller based on the experiences collected by all the agents and shared with all the agents. Unlike in [39], the CTDE approach significantly reduces the computational complexity involved and the amount of memory required, as only a single DQN needs to be trained. Furthermore, since all the agents use the same DQN simultaneously, the action space (i.e., DQN size) does not increase with the number of agents. Consequently, this approach is scalable and suitable for large-scale networks. Moreover, the CTDE approach trains the DQN model based on the collective experience of all the agents. This not only ensures better stability (and avoids biased training), but also enables collaborative learning so the agents can select suitable actions. As a result, the system reward (i.e., the network sum rate) improves. The CTDE approach is empirically shown to converge for multi-agent RL with homogeneous agents [40].

Algorithm 2 provides the overall steps for training our proposed multi-agent DRL framework using the CTDE approach. We consider episodic learning with Z RL episodes and T steps per RL episode, each of which represents a TS. Thus, the channel gains and user positions vary at each step. At each TS of an episode, Algorithm 2 randomly generates each device's location and channel gain, and clusters the devices using the proposed Algorithm 1. The inclusion of device clustering in DRL model training enables the DQN model to learn the inherent relationship that exists between power control and device clustering for decision-making.

Algorithm 2 is summarized as follows. Algorithm 2 initializes the replay memory, the PA agent (i.e., the DQN to be trained), and the target DQN. At each step in each RL episode,

the channel gains and device locations vary (Line 8). A set of device clusters (i.e., agents) is formed for each AP using Algorithm 1 (Line 10). Each agent collects state information from the environment (Line 12). The clustered devices' transmit power levels are selected using the ϵ-greedy approach (Lines 13-22). Note that although each agent uses the same DQN, the actions they select are different since their observed states are different. Line 23 normalizes the transmit power of all the devices associated with an AP so that constraint C3 is satisfied. The updated transmit power values are used to determine the agent's rewards and next states (Line 24). Each agent's experience, such as $(s_{n,t}^{(m)}, a_{n,t}^{(m)}, r_{n,t+1}^{(m)}, s_{n,t+1}^{(m)})$, is stored in the replay memory D (Line 25). A mini-batch of experiences is randomly sampled from the replay memory to train the DQN agent (Line 27). This makes it possible to train the DQN agent based on previous experiences instead of the most recent ones, which are potentially correlated and may introduce bias in the training. The DQN's parameters are updated by applying the backpropagation technique and (32) (Line 28). Finally, the target DQN's weights are updated with the new weights of the trained DQN after every T_{step} number of steps (Line 29). Lines 6-29 are iteratively repeated for all the steps and RL episodes. Algorithm 2's convergence is confirmed via simulations (see Fig. 8).

Remark 3: In practice, the number of devices associated with an AP and the number of devices per cluster can vary. We fix the dimensions of the trained DQN's input and output layers to make it robust in all scenarios. Since the maximum number of devices per cluster is K_D consistent with the state space description, the dimension of the DQN's input layer is set to $3K_D$. The dimension of the DQN's output layer is set to NP (i.e., the total number of discrete power levels). If an AP has more IIoT devices than the total number device clusters are allowed, Algorithm 1 selects the best K_D devices for each cluster at each TS. Meanwhile, a device cluster's state space is padded with zeros when the cluster has fewer than K_D devices [41].

Algorithm 2 Algorithm for Training DQN-Enabled PA Agent

1: **Input:** Maximum number of episodes Z and maximum number of steps per episode T.
2: **Initialize:** the replay memory D to zero.
3: Create a DQN-enabled PA agent, $q(:, :; \theta)$ with random weights θ.
4: Create a target DQN $q(:, :; \theta^-)$ with $\theta^- = \theta$.
5: **for** *episode* $\leftarrow 1 : Z$ **do**
6: Initializes channel fading gains for all the devices and APs.
7: **for** $t \leftarrow 1 : T$ **do**
8: Vary the devices' locations within APs' coverage zone and channel fading gains using (1).
9: **for** $m \leftarrow 1 : M$ **do**
10: Update N IIoT device clusters $\left\{\mathcal{C}_1^{(m)}, \mathcal{C}_2^{(m)}, \cdots, \mathcal{C}_N^{(m)}\right\}$ using Algorithm 1.
11: **for** $n \leftarrow 1 : |\mathcal{Z}^{(m)}|$ **do**
12: Observe the state of the environment, $s_{n,t}^{(m)}$. Initialize $a_{n,t}^{(m)} \leftarrow \varnothing$.
13: **for** $u \leftarrow 1 : |\mathcal{C}_n^{(m)}| + 1$ **do**
14: Generate a random number z.
15: **if** $z \geq \epsilon$ **then**
16: Determine $i = \arg\max_{a \in \mathcal{A}_u^{(n)}} q(s_{n,t}^{(m)}, a, \theta)$.
17: **else**
18: Select $i \in \{1, 2, \cdots, |\mathcal{A}_u^{(n)}|\}$ randomly.
19: **end if**
20: Assign $a_{n,t}^{(m)} \leftarrow a_{n,t}^{(m)} \cup \mathcal{A}_u^{(n)}[i]$.
21: **end for**
22: **end for**
23: Normalize the common and private messages' transmit power over all the clusters so that $\frac{|a_{n,t}^{(m)}|_1}{P_t} = 1$ is satisfied.
24: Transmit using the normalized transmit power and measure the instantaneous SINRs at the devices. Determine the immediate reward $r_{n,t+1}^{(m)} = \mathrm{R}_{total}^{(m)}$ and the next state $s_{n,t+1}^{(m)}, \forall n \in \mathcal{Z}^{(m)}$.
25: Save the experience $\left(s_n^{(m)}, a_{n,t}^{(m)}, r_{n,t+1}^{(m)}, s_{n,t+1}^{(m)}\right)$, $\forall n \in \mathcal{Z}^{(m)}$, in replay memory D.
26: **end for**
27: Sample a random mini-batch of experience from D.
28: Update the weight of PA agent, θ, by applying (32) and the backpropagation method.
29: After every T_{step} time step, update $\theta^- = \theta$.
30: **end for**
31: **end for**
32: **Output:** Trained PA agent $q^*(s, a; \theta)$.

V. OVERALL DCPM ALGORITHM

A. DESCRIPTION OF DCPM ALGORITHM

All the steps of the proposed DCPM framework are set out in Algorithm 3, which is summarized as follows. Each AP is considered to have a copy of the trained DQN-enabled PA agent. At each TS, the APs first collect the instantaneous channel gains of their associated IIoT devices (Line 5). Next, they form N device clusters by executing Algorithm 1. Then, they determine the transmit power levels of the common and private messages in their associated device clusters by executing Lines 7-16 using the trained DQN. Finally, they transmit data to the devices in each cluster using the optimized PA. Note that Lines 5-16 can be executed simultaneously and independently at each AP without needing to exchange any information. Consequently, Algorithm 3 can be implemented in a distributed manner at each AP. Distributed implementation enables rapid algorithm execution, which makes Algorithm 3 particularly advantageous for delay-sensitive IIoT applications.

B. COMPUTATIONAL COMPLEXITY OF DCPM ALGORITHM

We first determine the computational complexity of executing Lines 5-16. Line 6's computational complexity is $\mathcal{O}(K^2N)$. Note that each device cluster can accommodate a maximum of K_D devices. Hence, the worst-case computational complexity of executing Lines 7-12 is $\mathcal{O}(NK_DNP)$, where NP is the number of available discrete power levels. The computational complexity of the remaining lines is $\mathcal{O}(1)$. Thus, the computational complexity of making clustering and PA decisions for a single AP is $\mathcal{O}(N(K^2 + K_DNP))$. Since there are a total of M APs in the system, the overall computational complexity of executing Algorithm 3 in a single TS is $\mathcal{O}(MN(K^2 + K_DNP))$.

Remark 4: For the optimal performance of Algorithm 3, the environment used to train the DQN agent in Algorithm 2

Algorithm 3 Distributed Clustering and Power Management (DCPM) Algorithm

1: **Input:** Number of APs M; number of IIoT devices K and RRBs N per AP; maximum number of devices per cluster K_D; trained DQN for PA $q^*(s, a; \theta)$; total number of TSs T_s.
2: **Initialize:** TS index $t = 1$.
3: **repeat**
4: **for** $m \leftarrow 1 : M$ **do**
5: Collect the instantaneous channel gains $\{g_{k,n}^{(m)}\}$, for the associated IIoT devices.
6: Determine N device clusters $\left\{\mathcal{C}_1^{(m)}, \mathcal{C}_2^{(m)}, \cdots, \mathcal{C}_N^{(m)}\right\}$ using Algorithm 1.
7: **for** $n \leftarrow 1 : N$ **do**
8: Acquire the state information $s_{n,t}^{(m)}$ described in Section IV-B.
9: **for** $u \leftarrow 1 : |\mathcal{C}_n^{(m)}| + 1$ **do**
10: Determine the optimal PA action $a_u^* = \arg\max_{a \in \mathcal{A}_u^{(n)}} q^*(s_{n,t}^{(m)}, a; \theta)$.
11: **end for**
12: **end for**
13: **for** $n \leftarrow 1 : N$ **do**
14: Determine the transmit power of the common message as

$$P_{c,n}^{(m)} = \frac{a^*_{|\mathcal{C}_n^{(m)}|+1}}{\sum_{n=1}^{N}\sum_{u=1}^{|\mathcal{C}_n^{(m)}|+1} a_u^*} \times P_t$$

15: Determine the transmit power of the k-th device's private message as

$$P_{p,k,n}^{(m)} = \frac{a_k^*}{\sum_{n=1}^{N}\sum_{u=1}^{|\mathcal{C}_n^{(m)}|+1} a_u^*} \times P_t, \forall k \in \mathcal{C}_n^{(m)}$$

16: **end for**
17: **end for**
18: Perform downlink data transmission at all cells using the optimized device clusters and PA; increase the TS index $t = t + 1$.
19: **until** $t > T_s$
20: **Output:** Device clusters and transmit PA solution to P0 at each TS.

and the environment observed during online inference need to be similar. Nevertheless, in practice, an IIoT network usually exhibits several dynamic factors that make it difficult to keep the environment identical throughout model training and online inference. This challenge can be addressed by periodically updating the trained DQN as follows. Train the initial DQN model using Algorithm 2 and a realistic network simulator or digital twin.[8] Then, deploy the trained model in a live network (i.e., in Algorithm 3), and periodically collect the experiences of the agents (i.e., device clusters) from the APs and send them to a replay buffer. Once a sufficient number of sample experiences have been collected, the weights of the DQN model can be updated by executing Lines 27-29 of Algorithm 2. Periodically updating the DQN in this way will enable Algorithm 3 to maintain its adaptability to dynamic IIoT networks. It is worth noting that the following two types of communication overhead are required at each TS for DQN model training: (1) APs send their experiences (i.e., observed states, actions, and rewards) to the central server, and (2) the updated DQN model is sent back to the APs by the central server. While the amount of information exchanged increases with the number of APs, TSs, and RL episodes, we assume that the APs are connected to the server via high-speed wired (e.g., Ethernet or optical fiber) links. As a result, the overhead can be afforded. Additionally, it is important to note that training is a one-time data-intensive process. Once the model has been trained, deploying it does not require significant communication overhead, as each AP can make distributed decisions independently using the trained model.

[8]Note that Algorithm 2 requires several rounds of exploration (i.e., trial and error) to collect sufficient samples and optimize the DQN network. However, conducting exploration in a live network is expensive. Algorithm 2 can be executed in a virtual domain while considering the realistic network simulator as the environment to reduce costs.

VI. SIMULATION RESULTS

A. BENCHMARK SCHEMES

This section presents our numerical simulation results and evaluates the performance of the proposed scheme for downlink RSMA-enabled IIoT networks that considers both FBL and HWI-induced distortions. In the subsequent numerical results, unless specified, our DCPM algorithm considers perfect SIC at the IIoT devices. For performance comparison, we consider the following benchmark schemes.

1) **Clustering algorithms:** To demonstrate the superiority of our proposed clustering algorithm, we compare its performance to that of the following clustering algorithms.
   - **Random clustering:** This scheme allocated devices to clusters in a purely random fashion, with no specific pattern or criteria governing allocation.
   - **Clustering using channel gain:** In this case, device clusters are obtained by assigning the IIoT devices to their most suitable RRBs from the perspective of channel gain. This clustering scheme follows the same steps that are mentioned in Algorithm 1 except for when it comes to calculating the score Δ_k for the k-th device, $\forall k$. In this scheme, the score is calculated as the difference between the highest and second-highest channel gains of the k-th device obtained from the RRBs, which implies the importance of assigning the k-th device to its most suitable RRB. The pseudocode of the channel gain-based device clustering algorithm is summarized in Algorithm 4.

   Once the IIoT devices have been clustered, the transmit PA is selected using the trained DQN from Algorithm 2.
2) **Interference management schemes:** We compare the performance of the proposed DCPM algorithm (i.e. Algorithm 3) to that of the following interference management schemes.
   - **Treat interference as noise (TIN):** In this scheme, no common message is transmitted only the private messages are. Each device decodes its intended message by treating the interference from other devices in the same cluster as pure noise.
   - **RSMA with imperfect SIC (ImpSIC):** This scheme considers that the receivers' SIC process is imperfect. Consequently, residual interference

from the common message is encountered when decoding private messages. We take this interference into account by considering $\theta_c > 0$ in (12).

Since both the TIN and RSMA with ImpSIC schemes employ the DCPM framework, they are denoted by DCPM-TIN and DCPM-ImpSIC, respectively, in the ensuing discussion.

3) **PA algorithms:** We compare the performance of the proposed DCPM algorithm to that of the following benchmark PA algorithms. Note that the benchmark PA algorithms are applied only after Algorithm 1 has been executed to obtain each AP's device clusters.

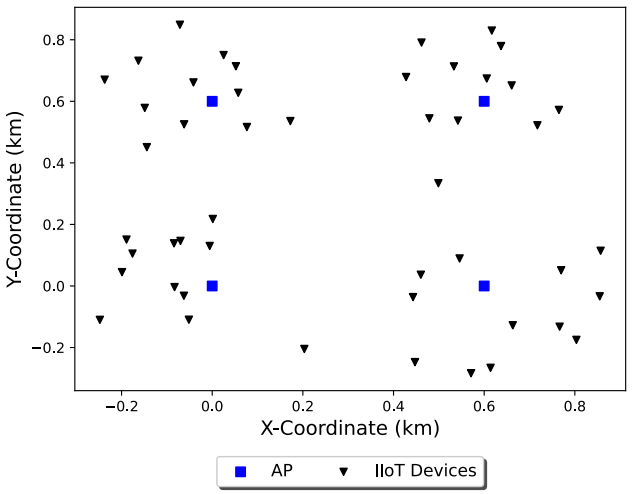
- **Weighted minimum mean square error (WMMSE):** The conventional WMMSE algorithm, which is given in [42], is used to optimize the transmit PA of the clustered devices.
- **Random power (RP):** Each IIoT device's transmit power level at each TS is chosen randomly between 0 and P_{max}.
- **Maximum power (MaxP):** The IIoT devices in each cluster are given the maximum power level P_{max} at each TS.

Algorithm 4 IIoT Device Clustering Algorithm for the m-th AP Based on the Channel Gain

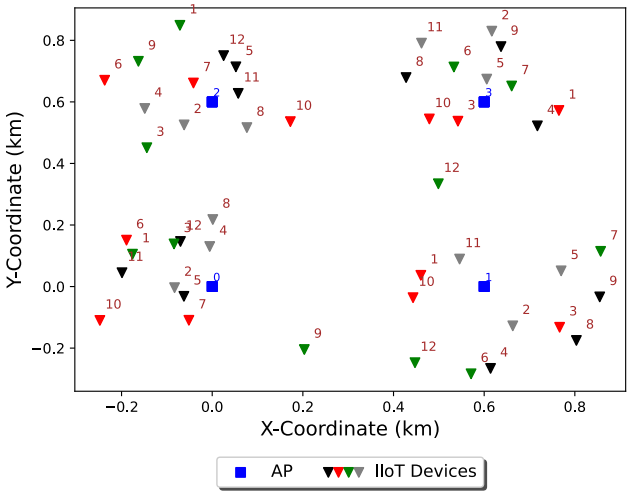
Input: Number of IIoT devices K, number of RRBs N, channel gains $\{g_{k,n}^{(m)}\}$.
Initialize: $\mathbf{UM}_U = \{1, 2, \cdots, K\}$, $D_n = 0, \forall n \in \{1, 2, \cdots, N\}$; IIoT device clusters $\mathcal{C}_n = \varnothing, \forall n \in \{1, 2, \cdots, N\}$.
repeat
  Find $\mathbf{UM}_R = \{n \in \{1, 2, \cdots, N\} | D_n < K_D\}$.
  for $k = 1 : |\mathbf{UM}_U|$ **do**
    Calculate $n_k^* = \arg\max_{n \in \mathbf{UM}_R} g_{k,n}^{(m)}$.
    Calculate $\Delta_k = g_{k,n_k^*}^{(m)} - \max_{\substack{n \in \mathbf{UM}_R \\ n \neq n_k^*}} g_{k,n}^{(m)}$.
  end for
  Find $k^* = \arg\max_{k \in \mathbf{UM}_U} \Delta_k$.
  Assign $\mathcal{C}_{n_k^*} \rightarrow \mathcal{C}_{n_k^*} \cup \{k^*\}$; update $D_{n_k^*} = D_{n_k^*} + 1$ and $\mathbf{UM}_U \rightarrow \mathbf{UM}_U \setminus \{k^*\}$.
until $\mathbf{UM}_U = \varnothing$ or $\mathbf{UM}_R = \varnothing$
Output: IIoT device clusters $\mathcal{C}_1, \mathcal{C}_2, \cdots, \mathcal{C}_N$.

B. SIMULATION SETTINGS

Our simulation model comprises four IIoT cells that each contain one AP and K IIoT devices placed randomly around the AP. To ensure that the position of the devices and the AP do not overlap, a small device-free region of radius r_s is maintained around each AP. Fig. 4(a) shows an example of a network configuration for 4 APs and 48 IIoT devices without clustering. When the IIoT devices are clustered using Algorithm 1 in such a way that the SINR is maximized, the network configuration becomes as depicted in Fig 4(b). Note that the devices shown in the same color in the figure are part of the same cluster. In our specific setup, we consider that each AP can have $N = 4$ clusters, and each cluster can support a maximum of $K_D = 3$ devices. The fixed power level used to execute Algorithm 1 is obtained as $P_0 = \frac{P_t}{(K_D+1)N}$.



(a) Without clustering.



(b) With clustering.

FIGURE 4. Network configuration for 4 APs and 48 IIoT devices.

Our network simulation settings are shown in Table 2. The path loss parameter values provided in Table 2 were selected in accordance with [31, Table 7.2-4]. We use a linear distortion model to simulate HWIs and consider $\sqrt{\sigma_t^2 + \sigma_r^2} = 0.1$ in accordance with [43].

In our proposed solution, each agent trains the DQN with one input layer, two fully connected hidden layers, and one output layer. The input layer N_1 contains the state elements described in Section IV-B.1. The number of neurons in the two hidden layers (N_2, N_3) is (64, 32). Finally, we set the output layer $N_4 = NP = 20$ outputs. The hyperparameters adopted to train the DQN are listed in Table 3. The weight-update time step, T_{step}, is set to 100. This indicates that every 100 time steps, the weights of the trained DQN, which are denoted by θ, are aligned with the weights of the target DQN, which are represented by θ^-. When updating θ, we employ an RMSprop optimizer with an adaptive learning rate $\alpha(t)$, where $\alpha(t) = (1 - \lambda)\alpha(t - 1)$. The initial learning rate was set to $\alpha(0) = 5e - 3$, and λ, to 10^{-3}. This initial learning

TABLE 2. Simulation settings: IIoT network's parameters.

Parameter	Value
Network Parameters	
Number of AP M	4
Total number of devices	48
Number of clusters N	4
Number of devices per cluster K_D	3
Cell radius R	300 m
Small region radius r_s	50 m
Total power P_t	38 dBm
Maximum power P_{max}	30 dBm
Minimum power P_{min}	5 dBm
AWGN power σ^2	-174 dBm
Blocklength n_b	256
Bit error rate ϵ	10^{-5}
Total bandwidth B	100 MHz
Doppler Frequency f_d	15 Hz
Time slot duration T_s	20 ms
Path Loss Parameters	
Height of AP H	3 m
Height of IIoT devices h_k	1 m
Effective clutter height h_c	2 m
Carrier frequency f_c	1 GHz
Typical clutter size $d_{clutter}$	10 m
Clutter density r	0.4

TABLE 3. Simulation setting: Hyper-parameters of DQN-based PA.

Parameter	Value
Initial learning rate $\alpha(0)$	$5e^{-4}$
Discount factor γ	0.8
Number of episodes Z	1000
Replay memory buffer size D	5000
Mini-batch size D_b	32
Loss function	MSE
Optimizer	RMSprop
Activation function	tanh
Number of power levels	20
Time step T_{step}	100

rate was determined through experimentation and was found to improve learning performance in our simulation results. Furthermore, we utilize the adaptive ϵ-greedy algorithm to balance exploration and exploitation during the learning process. This algorithm adjusts the exploration rate ϵ over time to ensure the agent gradually transitions from exploration to exploitation as it learns more about the environment. The formula used to update ϵ is $\epsilon(t) = \max\{\epsilon_{\min}, (1-\lambda_\epsilon)\epsilon(t-1)\}$, where $\epsilon_{\min}$ represents the minimum exploration probability and λ_ϵ is the decay rate. In our implementation, we initialize $\epsilon(0)$ at 0.7 and set $\epsilon_{\min}$ to 10^{-2} and λ_ϵ to 10^{-3}.

We performed training using Algorithm 2 for 1000 episodes to help the agent learn the optimal PA policy for different channel conditions. To train the DQN of the DCPM framework, experiences (i.e., datasets) were collected by iteratively interacting the DQN with a simulated IIoT environment having the parameters indicated in Table 2. The experiences, which take the form of (state, action, reward, and next state) tuples, are stored in a reply buffer of size (5,000) and randomly sampled in a mini-batch to iteratively train the DQN model. Note that the experiences generated include a wide range of states and actions that reflect various network conditions, devices' HWI configurations, and dynamic channel gains. This ensures that the DQN agent is exposed to a wide variety of scenarios during training, which enhances its ability to generalize and perform well in real-world situations. Once the DQN was trained, it was deployed in each AP and Algorithm 3 was executed. The sum rate of the ensuing numerical results is from the online execution of Algorithm 3 in different channel conditions. For simplicity, we consider a similar networking environment for both model training and online inference. We implemented our program in Python 3 and ran it on a 64-bit Windows 10 machine with an Intel Core i7-6700 CPU with a 3.40 GHz processor and 8 GB of RAM.

TABLE 4. Jain fairness index for RSMA and OFDMA technologies.

Number of clusters	Number of devices per cluster (K_D)	Fairness index for RSMA	Fairness index for OFDMA
2	6	0.361	1/6
4	3	0.437	1/3
6	2	0.603	1/2
12	1	1	1

C. RSMA VERSUS OFDMA: NUMBER OF CLUSTERS VERSUS JAIN FAIRNESS INDEX

The Jain fairness index quantifies the fairness of resource allocation among multiple devices in a network or system. From [44], we obtain Jain's fairness index as:

$$\mathcal{J}(x) = \frac{(\sum_{k=1}^{K_D} x_k)^2}{K_D \sum_{k=1}^{K_D} x_k^2}, \tag{35}$$

where x_k is the achievable rate of the k-th device and K_D is the total number of devices allowed per RRB (i.e., device cluster). The Jain fairness index measures how fairly the bandwidth of an RRB is allocated among its associated devices, on a scale from 0 to 1 [44], with 1 representing perfect fairness and indicating that all users receive an equitable share of the available resources. Conversely, a Jain index value that approaches 0 indicates substantial unfairness, meaning that some users receive disproportionately more resources, while others are given comparatively less resources. Essentially, the Jain fairness index quantitatively measures how resources are distributed and is a valuable tool for evaluating fairness in different network and resource-sharing scenarios. Note that each AP has 4 RRBs in the setup considered. Hence, at any given time, OFDMA can support one device per RRB and a maximum of 4 devices per AP. In contrast, RSMA can support 12 devices per AP with intra-cluster interference. As a result, the sum rate is not a rational metric for comparing the two schemes. We therefore use the Jain fairness index to assess and compare the RSMA and OFDMA schemes' ability to fairly allocate resources.

In Table 4, we evaluate the number of clusters (i.e., RRBs) per AP and the fairness index for the OFDMA and RSMA schemes. It is evident from the table that increasing the number of clusters per AP decreases the fairness index. When it comes to the OFDMA scheme, each RRB (or cluster) can

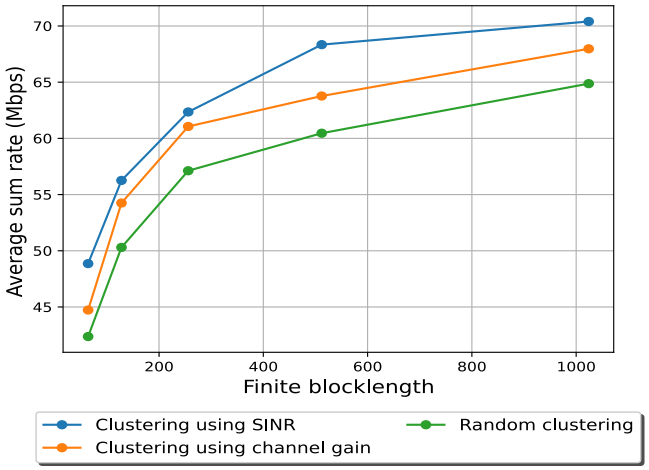


FIGURE 5. Average sum rate for different blocklengths and clustering algorithms.

support only one user at a time. Hence, its fairness index is always $1/K_D$. Meanwhile, the RSMA scheme simultaneously supports all the clustered devices with certain data rates by effectively mitigating intra-cluster interference. Accordingly, for a given number of RRBs, RSMA outperforms OFDMA when it comes to the fairness index, as Table 4 shows. For example, in the third scenario, where $N = 6$ and $K_D = 2$, RSMA and OFDMA achieve Jain fairness index values of 0.603 and 0.5, respectively.

In the last scenario, we consider 12 RRBs and thus 12 clusters per AP, with each cluster supporting only one device. In this context, the RSMA and OFDMA schemes align, which means they both obtain same fairness index. We conclude that the RSMA scheme allocates resources more fairly than the OFDMA scheme in scenarios where there are fewer RRBs than devices.

D. AVERAGE SUM RATE FOR DIFFERENT BLOCKLENGTHS AND CLUSTERING ALGORITHMS

Fig. 5 plots our proposed solution' average sum rate for various blocklengths and clustering algorithms. We consider that $\epsilon = 10^{-5}$ and $n_b \in \{64, 128, 256, 512, 1024\}$. It is evident from the figure that the average sum rate increases as the blocklength value increases. For instance, the proposed DCPM framework[9] achieves an average sum rate of 48.86 Mbps and 70.4 Mbps when $n_b = 64$ and $n_b = 1024$, respectively. This observation is logically intuitive, as larger blocklengths bring the average data rate closer to the near-Shannon channel capacity. It is worth noting that our proposed DCPM framework achieves a non-negligible average rate in the short blocklength regime, which makes it a compelling scheme for URLLC scenarios.

Fig. 5 also highlights that our proposed clustering algorithm outperforms the benchmark clustering algorithms for both short and long blocklengths. For example, when the blocklength is $n_b = 256$, the proposed DCPM framework achieves an average sum rate that is 1.29 Mbps higher than the average sum rate of the DQN-based PA with clustering using channel gain algorithm and 5.23 Mbps higher than the average rate of the DQN-based PA with random clustering algorithm. These outcomes are anticipated, given that our proposed clustering algorithm (i.e., Algorithm 1) is designed to cluster IIoT devices with the objective of improving the SINR and, consequently, enhancing system capacity. In contrast, Algorithm 4 assigns IIoT devices to the most suitable RRBs without considering interference. Hence, it does not necessarily maximize the SINR(s) and thus reduces the system capacity. Both Algorithms 1 and 4 have the same computational complexity. Meanwhile, the random clustering scheme can result in severe intra-cluster interference and consequently achieve the lowest sum rate. We therefore conclude that our proposed SINR-based device clustering algorithm has clear merit over both the channel gain-based and random device clustering schemes.

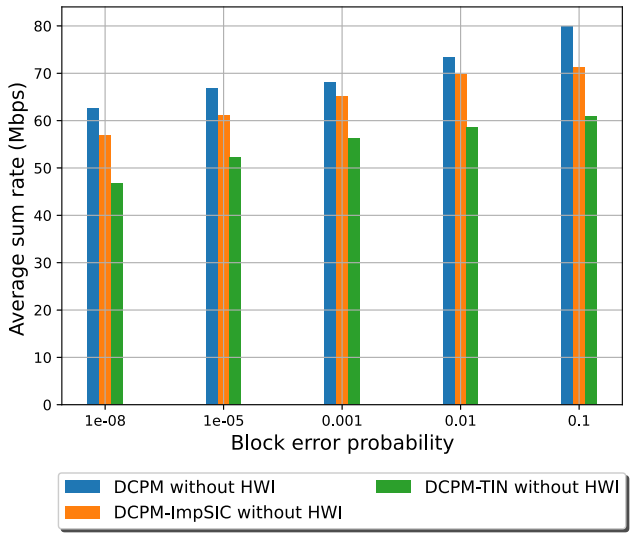
E. AVERAGE SUM RATE FOR DIFFERENT BLOCK ERROR PROBABILITY AND INTERFERENCE MANAGEMENT SCHEMES

Figs. 6(a)-(b) plot the average sum rate with respect to the block error probability considering $n_b = 256$ and $\epsilon \in \{10^{-1}, 10^{-2}, 10^{-3}, 10^{-5}, 10^{-8}\}$. The average sum rate increases as ϵ increases. This is intuitively expected since $\mathrm{Q}^{-1}(\epsilon) \to 0$ as $\epsilon \to 1$ in (16), and consequently, the device's average data rate approaches Shannon capacity as the tolerable block error probability increases.

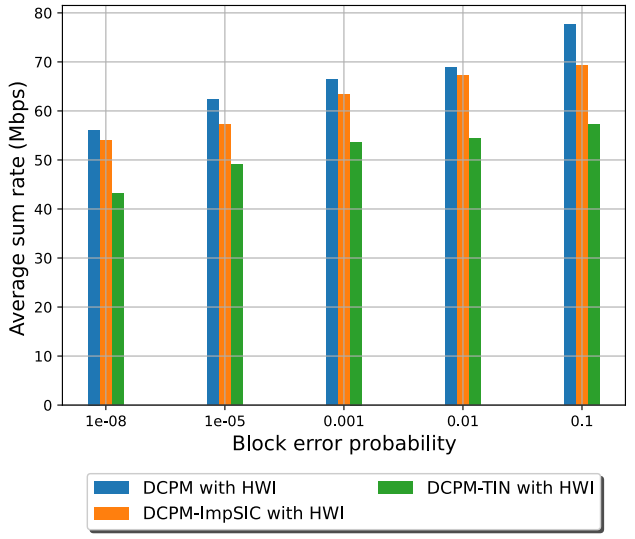
In Fig. 6(a), we also compare the average sum rate of the proposed DCPM framework with that of different interference management schemes without considering any HWI-induced distortions. Fig. 6(a) clearly depicts that the proposed DCPM framework outperforms both the DCPM-TIN and DCPM-ImpSIC schemes. This observation can be explained by the following arguments. First, for the imperfect SIC scheme with $\theta_c = 10^{-4}$, the SINRs of the private streams decrease in RSMA (12) due to uncancelled interference from the common message. This leads to a reduction in the average sum rate of the DCPM-ImpSIC scheme. Meanwhile, no interference cancellation is considered in the TIN scheme for decoding the devices' messages, and consequently, the scheme usually exhibits less capacity than RSMA. Hence, the proposed DCPM framework also outperforms the DCPM-TIN scheme. For instance, Fig. 6(a) depicts that at $\epsilon = 10^{-3}$, the DCPM framework achieves an average sum rate that is 2.93 Mbps higher than that of the DCPM-ImpSIC scheme and 11.9 Mbps higher than that of the DCPM-TIN scheme.

In Fig. 6(b), we compare the average sum rate of the proposed DCPM framework with that of different interference management schemes while taking into account the impact of HWI-induced distortions. It is worth noting that HWI-induced distortions have a detrimental effect on the performance of all schemes. For example, when HWI-induced distortions are considered, the average sum rate

[9]In Fig. 5, the legend entry "*Clustering using SINR*" implies with the DCPM framework.



(a) Without HWI.



(b) With HWI.

FIGURE 6. Average sum rate for different block error probability and interference management schemes.

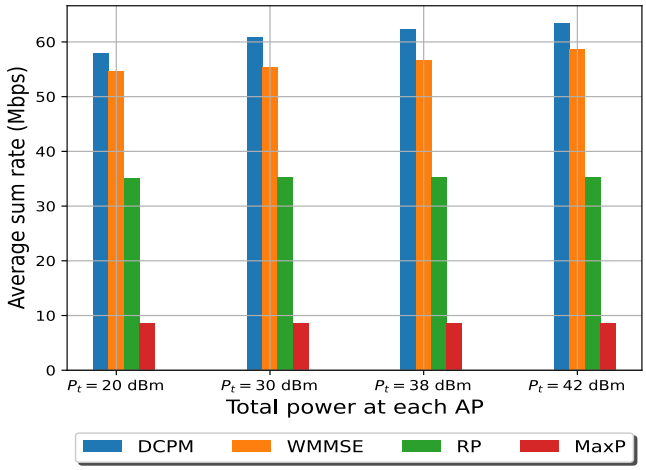


FIGURE 7. Average sum rate for different total power values per AP and power management schemes.

of the DCPM, DCPM-ImpSIC, and DCPM-TIN schemes decreases approximately 2 to 6 Mbps, 2 to 4 Mbps, and 3 to 4 Mbps, respectively. Despite the reduction in capacity, our proposed DCPM framework also consistently outperforms both the DCPM-TIN and DCPM-ImpSIC schemes in the presence of HWI-induced distortions. Overall, our proposed DCPM framework is better able to manage interference than both the DCPM-TIN and DCPM-ImpSIC schemes are.

F. AVERAGE SUM RATE FOR DIFFERENT TOTAL POWER VALUES PER AP AND POWER MANAGEMENT SCHEMES

Fig. 7 compares the average sum rate of the proposed DCPM scheme with that of three benchmark schemes, namely WMMSE, RP, and MaxP, for an IIoT network with $n_b = 256$ and $\epsilon = 10^{-5}$. It is evident from Fig. 7 that the average sum rates of all the PA schemes gradually increase as the total power available at each AP increases. For instance, when the total power $P_t = 20$ dBm, the DCPM scheme achieves an average sum rate of 57.9 Mbps. However, when the total power is increased to $P_t = 42$ dBm, the DCPM algorithm's average sum rate increases to 63.44 Mbps.

Fig. 7 shows that our proposed DCPM algorithm achieves a higher average sum rate than the state-of-the-art transmit PA schemes for both small and large transmit power limits. For instance, when $P_t = 30$ dBm, the DCPM scheme's average sum rate exceeds that of the WWMSE, RP, and MaxP schemes by 5.46 Mbps, 25.61 Mbps, and 52.22 Mbps, respectively. We emphasize that the WMMSE scheme (i) is more time-consuming/computationally complex due to iterative optimization, especially in the context of large-scale networks, (ii) is primarily optimized for interference channels in the infinite blocklength regime and (iii) does not take into consideration any HWI-induced signal distortions. Due to these limitations, it not only is sub-optimal for interference channels in FBL-coded IIoT networks but also suffers from high computational and implementation complexity. Meanwhile, the RP scheme does not guarantee any performance gain over fading channels, and the MaxP scheme generates severe intra-cell interference in the network. In contrast to these PA schemes, the proposed DCPM algorithm can learn and adapt to dynamic wireless channel conditions. When the devices' wireless channel conditions vary over time, the optimal PA strategy also changes. The DQN-based approach is able to effectively solve this problem by learning to adapt the PA strategy based on past experiences. Furthermore, it can choose suitable power levels to jointly mitigate interference and HWI-induced signal distortions. In essence, the proposed DCPM approach's resilience to intra-cell interference and HWI-induced signal distortions gives it clear merit over the existing transmit PA methods.

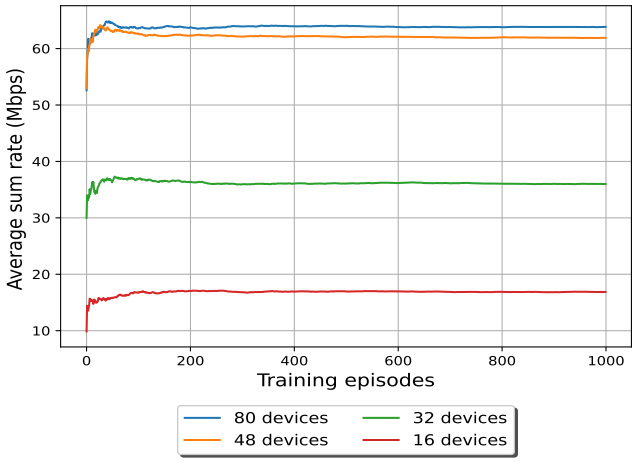


FIGURE 8. Average sum rate for different numbers of IIoT devices and numbers of training episodes.

G. CONVERGENCE AND SCALABILITY OF THE DCPM ALGORITHM

1) CONVERGENCE OF THE DCPM

Fig. 8 plots the DCPM algorithm's average sum rate with respect to the number of training episodes considered in the PA scheme for different numbers of IIoT devices in the network. Fig. 8 shows that the DCPM algorithm converges to a stable system capacity in approximately 200 episodes for the device quantities considered. As a result, its convergence is guaranteed. Fig. 8 also shows that the average sum rate increases as the number of devices increases. This observation is expected since each AP's average rate (which is given by (18)) is equal to the sum of the rates of all the devices in its associated clusters. However, intra-cluster interference also increases as the number of devices per cluster increases. We emphasize that the proposed DCPM algorithm efficiently conducts device clustering and PA, and thereby mitigates intra-cluster interference and achieves multi-user capacity gain. Note that we consider $N = 4$ and $k_D = 3$ for each AP, and consequently, each AP can accommodate a maximum of 12 devices. As a result, the system can support a total of 48 IoT devices at any given time. The DCPM algorithm selects the top 12 devices per AP that have the highest SINRs when there are 20 IIoT devices per AP. Hence, the DCPM scheme achieves a nearly identical system capacity in the scenarios with 48 and 80 devices. For instance, its average sum rate is 62.07 Mbps and 63.93 Mbps for the scenarios with 48 and 80 IIoT devices, respectively.

2) RUN TIME DURATION OF THE TRAINED DCPM ALGORITHM

In our proposed DCPM framework, the DQN model is saved after training for 1000 episodes, and the subsequent evaluations are conducted by applying the trained model in Algorithm 3 across various network configurations. Note that this algorithm runs in parallel in a distributed manner at each AP.

TABLE 5. Required time versus different numbers of IIoT devices.

	4 devices	8 devices	12 devices	20 devices
Time required (ms)	16	35	62	150

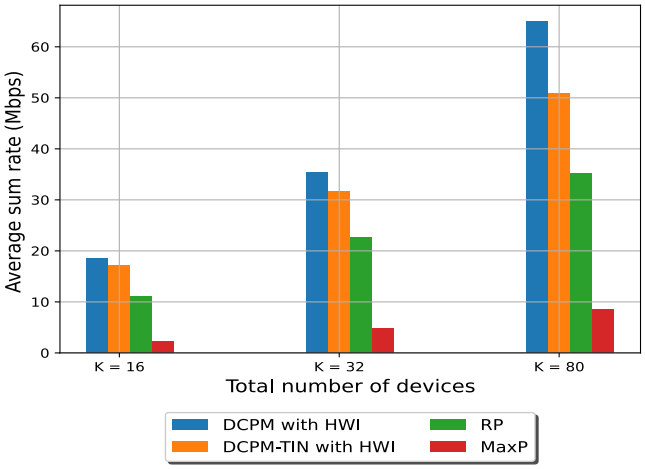


FIGURE 9. Average sum rate of DCPM and other interference management schemes for different numbers of IIoT devices.

Table 5 shows the average amount of time required to obtain the optimal decision variables (i.e., device clustering and transmit PA) for a single AP and a single TS. We observe that the DCPM framework's execution time increases with the number of devices. This observation is logically intuitive, as the computational complexity of Algorithm 3 increases (quadratically) with the number of devices. For instance, the DCPM framework's execution time for 4 and 20 IIoT devices per AP is 16 ms and 150 ms, respectively. Note that this duration includes the time it takes to obtain all the required information (e.g., instantaneous CSI and device positions), compute the SINRs, cluster all the IIoT devices, and apply PA strategy. We note that all the evaluations presented in this section were conducted on a personal machine. The run time duration of the trained DCPM algorithm could be further reduced by using hardware with powerful computational capabilities, such as graphic processor units (GPUs) and cloud computing.

3) SCALABILITY OF THE DCPM ALGORITHM

To assess the scalability of our proposed solution, we conducted simulations using the DCPM algorithm and benchmark schemes with different numbers of IIoT devices. Fig. 9 plots the average sum rate of different benchmark schemes with respect to the number of IoT devices while taking into account the impact of HWI-induced distortions. It is evident that our proposed algorithm outperforms the benchmark schemes in terms of average sum rate for all the device counts considered. For example, when the number of devices is set to 80, the proposed algorithm achieves an average sum rate that exceeds that of the DCPM-TIN with HWI, RP, and MaxP schemes by 13.99 Mbps, 29.8 Mbps, and 56.4 Mbps, respectively. Thus, the proposed DCPM algorithm is scalable and

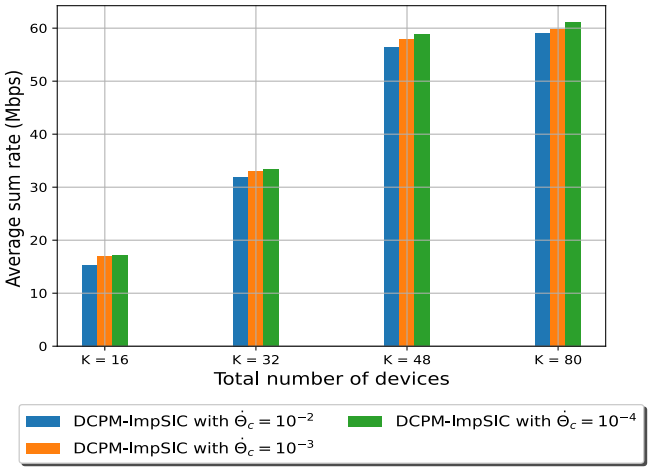


FIGURE 10. Average sum rate for different numbers of IIoT devices and different SIC capabilities.

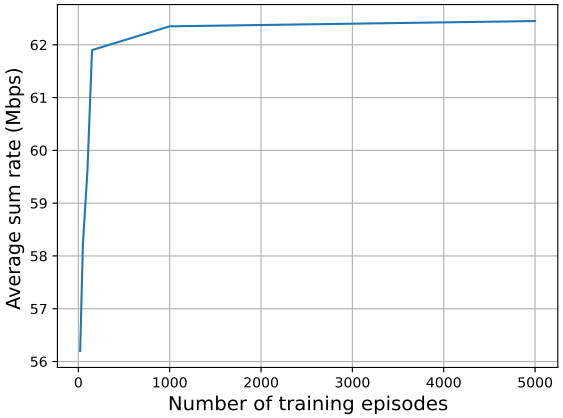


FIGURE 11. Average sum rate for different numbers of training episodes.

has a clear advantage over the benchmark schemes when it comes to mitigating interference in large-scale IIoT networks.

4) IMPACT OF SIC ON THE DCPM FRAMEWORK'S PERFORMANCE

Fig. 10 plots the DCPM algorithm's average sum rate with imperfect SIC and HWI-induced signal distortions with respect to the total number of IIoT devices considered.

It is expected that the DCPM algorithm's average sum rate would decrease as the SIC error coefficient θ_c increases. This is because as the SIC error coefficient θ_c increases, the residual interference from the common message also increases during private message decoding at the devices. However, we observe that the imperfect SIC does not significantly deteriorate the DCPM algorithm's performance. For instance, if 80 IIoT devices are considered, the DCPM-ImpSIC with HWI scheme's average sum rate is 58.98 Mbps, 59.82 Mbps, and 61.18 Mbps when θ_c is set to 10^{-2}, 10^{-3}, and 10^{-4}, respectively. Furthermore, even when the DCPM algorithm has imperfect SIC, its average sum rate increases as the number of IIoT devices increases. Based on these observations, we can conclude that our proposed DCPM algorithm is robust to different SIC capabilities, which makes it a practical solution for RSMA-based IIoT networks.

5) IMPACT OF THE NUMBER OF TRAINING EPISODES ON THE DCPM FRAMEWORK'S PERFORMANCE

In this section, we investigate how varying the number of training episodes affects the DCPM framework's performance. We consider $Z = \{20, 50, 200, 1000, 5000\}$ training episodes. After each specified number of episodes, the DQN model is saved and then tested in our environment. Fig. 11 illustrates the DCPM algorithm's average sum rate corresponding to the different numbers of training episodes. The results clearly show that increasing the number of episodes improves the proposed scheme's performance. For instance, when $Z = 20$ and $Z = 100$, the DCPM framework achieves an average sum rate of 56.2 Mbps and 59.62 Mbps, respectively. However, beyond 1000 episodes, the performance gains plateau and the DCPM algorithm's performance changes very little. Therefore, based on these observations, we limit our training episodes to $Z = 1000$ to balance training efficiency and performance improvement. This decision ensures that we achieve efficient performance without unnecessary computational overhead from additional episodes that provide diminishing returns.

H. COMPARISON OF DCPM USING THE SEQUENTIAL TECHNIQUE AND DCPM USING ALTERNATING OPTIMIZATION

In this section, we compare the performance of the proposed sequential technique and the alternating optimization (AO) method to evaluate their performance gap. AO is renowned for iteratively refining solutions by optimizing different sets of variables having the potential to improve performance. In the AO approach, the device clusters are updated at each iteration based on the transmit powers obtained in the previous iteration. Afterward, the SINRs (with HWI-induced distortions) are estimated for the newly formed device clusters. Finally, the PA is updated based on the new SINR values. These three steps are iteratively repeated 1000 times for each channel fading state. After each iteration, the solutions (i.e., device clusters and PA) and sum rates obtained are saved. Once all the iterations have been completed, the best sum rate and its corresponding cluster and PA are selected. As can be seen in Table 6, the DCPM using the AO scheme achieves a higher average sum rate than our proposed scheme for the different values of P_{max} considered. For instance, when $P_{max} = 20$ dBm, the DCPM using the sequential technique scheme achieves an average sum rate of 45.51 Mbps, whereas the DCPM using AO scheme achieves an average sum rate of 83.48 Mbps. However, AO requires significantly more time to achieve its final solution. More specifically, the AO approach took approximately 1000 times longer than the sequential method did to obtain its average sum rates reported in Table 6. Note that sequential optimization takes around 62 ms when there are 12 devices per AP. For the same number

TABLE 6. Average sum rate for different values of P_{max}.

	$P_{max} =$ 10 dBm	$P_{max} =$ 20 dBm	$P_{max} =$ 30 dBm
DCPM using the sequential technique	35.80 Mbps	45.51 Mbps	62.34 Mbps
DCPM using AO	42.35 Mbps	83.48 Mbps	138.36 Mbps

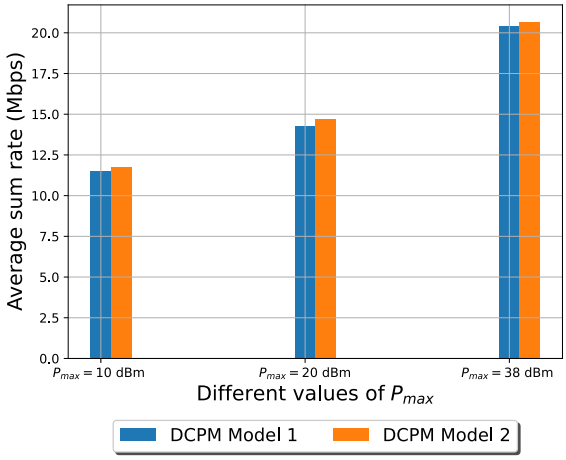


FIGURE 12. Average sum rate for different values of P_{max}.

of devices, the AO approach takes around 60.54 s. Clearly, this latter approach is not practically feasible for FBL-coded IIoT networks, which require fast computation for latency-sensitive control operations. Therefore, despite its improved sum rate, the AO approach lacks scalability, which makes it impractical for large-scale IIoT networks. Furthermore, it is clear from our simulation results that the gap between the approaches' performance increases as P_{max} increases. Thus, our solution is particularly suitable for scenarios with low P_{max} values.

I. PERFORMANCE EVALUATION OF THE PROPOSED ALGORITHM IN UNFAMILIAR HWI CONDITIONS

In this section, we evaluate the DCPM algorithm's performance in the presence of another HWI model outlined in the literature [45]. To this end, we consider the following two scenarios.

- DCPM Model 1: This model was trained using the HWI model considered in this study.
- DCPM Model 2: This model was trained using the non-linear distortion model outlined in [45].

After the training phase, both models were saved and subsequently tested in an environment where only the non-linear distortion model outlined in [45] was present. The aim of this analysis was to demonstrate that even when the DCPM model has been trained in a different environment (with alternate HWI models), it does not perform significantly worse than models that have been trained specifically for the testing environment in question. For instance, when $P_{max} = 10$ dBm, DCPM Model 1 and DCPM Model 2 achieve an average sum rate of 11.53 Mbps and 11.78 Mbps, respectively. Fig. 12 underscores that the proposed scheme is more adaptable than various HWI models considered in the literature. Based on our findings, we conclude that our proposed algorithm is generic in nature since it learns to achieve efficient resource allocation while interacting with the system. Therefore, it can effectively work with unfamiliar HWI models.

VII. CONCLUSION

In this paper, we proposed a resource optimization framework to manage co-channel interference and maximize system capacity in a downlink RSMA-enabled FBL-coded IIoT network with dynamic channel variation and HWI-induced signal distortions. The proposed resource optimization problem was proven to be NP-hard, and a distributed clustering and power management (DCPM) algorithm was proposed to address its computational intractability. The proposed DCPM algorithm first divides the IIoT devices associated with each AP into multiple non-overlapping RSMA clusters to enhance the clustered devices' SINRs. It then employs a trained DQN (obtained via multi-agent RL) in each cluster to allocate the AP's transmit power to different devices in the presence of realistic impairments (e.g., co-channel interference, HWI-induced distortions, and FBL). The proposed DCPM scheme does not require iterative optimization or instantaneous information about signal distortions for transmit PA. Extensive simulations confirm the DCPM algorithm can effectively maximize the capacity of the FBL-coded IIoT network. The simulation results confirm that the DCPM algorithm: (i) is able to adapt to changes in channel condition, block-length, block error probability, and IIoT device count; (ii) is more resilient to co-channel interference and HWI-induced signal distortions than existing PA and interference management schemes; and (iii) converges and can be scaled up for large-scale IIoT networks without a significant increase in computation time.

APPENDIX

The completion of the proof for Proposition 1 involves reducing the optimal coalition structure (OCS) problem to $P0$. To this end, we first describe the OCS problem as follows. Let A denote a set of K distinct agents, which are denoted as $\mathrm{A} = \{a_1, \cdots, a_K\}$. A coalition structure (CS) is defined as the partitioning of agents into disjoint coalitions, with each agent belonging to exactly one coalition. Given a CS denoted as CS and value $v_s \geq 0$ assigned to each coalition, where $s \in CS$, the total value of CS is expressed as $V(CS) = \sum_{S \in CS} v_{S_n}$. The objective of the OCS problem is to find the optimal CS^*, where $CS^* = \arg\max_{CS \in U} V(CS)$, with U representing the universal set of all possible CSs. In other words, the OCS problem aims to find the CS with the highest value.

To reduce the OCS problem to an instance of $P0$, we make the initial assumption that the transmit PA of all the APs in $P0$ is known. In this scenario, $P0$ becomes a device-to-cluster association problem. Each device is considered an agent in set A. Thus, the set of devices associated with a cluster forms a coalition, and these coalitions are disjoint since a device can be associated with only one cluster. Each AP in the IIoT network has a total of N clusters, and a

CS consists of at most N disjoint coalitions, which can be expressed as $CS = \{S_1, S_2, \cdots, S_N\}$, where S_n represents the set of devices associated with the n-th cluster. The value of coalition S_n is defined as $v_{S_n} = R_{C_n}^{(m)}$, with $R_{C_n}^{(m)}$ given by (17). Consequently, the total value of CS is determined as $V(CS) = \sum_{S_n \in CS} v_{S_n} = \sum_{n=1}^{N} \mathrm{R}_{C_n}^{(m)}$.

With this reduction, an optimal solution to $P0$ determines the optimal set of device clusters (or optimal CS) that has the maximum system capacity (or largest CS value). Clearly, an optimal solution to $P0$ corresponds to an optimal solution to the OCS problem. Stated differently, the ability to optimally solve $P0$ in polynomial time implies it is possible to solve the OCS problem optimally in polynomial time. As the OCS problem is known to be NP-hard [46, Proposition 2], it follows that $P0$ is also NP-hard. □

REFERENCES

[1] A. Mahmood et al., "Industrial IoT in 5G-and-beyond networks: Vision, architecture, and design trends," *IEEE Trans. Ind. Informat.*, vol. 18, no. 6, pp. 4122–4137, Jun. 2022.

[2] A. Ranjha, G. Kaddoum, and K. Dev, "Facilitating URLLC in UAV-assisted relay systems with multiple-mobile robots for 6G networks: A prospective of agriculture 4.0," *IEEE Trans. Ind. Informat.*, vol. 18, no. 7, pp. 4954–4965, Jul. 2022.

[3] H. R. Chi, C. K. Wu, N.-F. Huang, K.-F. Tsang, and A. Radwan, "A survey of network automation for industrial Internet-of-Things toward Industry 5.0," *IEEE Trans. Ind. Informat.*, vol. 19, no. 2, pp. 2065–2077, Feb. 2023.

[4] M. Z. Hassan, M. J. Hossain, J. Cheng, and V. C. M. Leung, "Device-clustering and rate-splitting enabled device-to-device cooperation framework in fog radio access network," *IEEE Trans. Green Commun. Netw.*, vol. 5, no. 3, pp. 1482–1501, Sep. 2021.

[5] A. Mishra, Y. Mao, O. Dizdar, and B. Clerckx, "Rate-splitting multiple access for 6G—Part I: Principles, applications and future works," *IEEE Commun. Lett.*, vol. 26, no. 10, pp. 2232–2236, Oct. 2022.

[6] B. Clerckx et al., "A primer on rate-splitting multiple access: Tutorial, myths, and frequently asked questions," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 5, pp. 1265–1308, May 2023.

[7] Md. Z. Hassan, Md. J. Hossain, J. Cheng, and V. C. M. Leung, "Joint throughput-power optimization of fog-RAN using rate-splitting multiple access and reinforcement-learning based user clustering," *IEEE Trans. Veh. Technol.*, vol. 70, no. 8, pp. 8019–8036, Aug. 2021.

[8] Z. Yang et al., "Optimization of rate allocation and power control for rate splitting multiple access (RSMA)," *IEEE Trans. Commun.*, vol. 69, no. 9, pp. 5988–6002, Sep. 2021.

[9] M. Z. Hassan et al., "Energy-spectrum effcient content distribution in fog-RAN using rate-splitting, common message decoding, and 3D-resource matching," *IEEE Trans. Wireless Commun.*, vol. 20, no. 8, pp. 4929–4946, Aug. 2021.

[10] J. Xu, O. Dizdar, and B. Clerckx, "Rate-splitting multiple access for short-packet uplink communications: A finite blocklength analysis," *IEEE Commun. Lett.*, vol. 27, no. 2, pp. 517–521, Feb. 2023.

[11] Y. Xu et al., "Rate-splitting multiple access with finite blocklength for short-packet and low-latency downlink communications," *IEEE Trans. Veh. Technol.*, vol. 71, no. 11, pp. 12333–12337, Nov. 2022.

[12] C. Sun, C. She, C. Yang, T. Q. S. Quek, Y. Li, and B. Vucetic, "Optimizing resource allocation in the short blocklength regime for ultra-reliable and low-latency communications," *IEEE Trans. Wireless Commun.*, vol. 18, no. 1, pp. 402–415, Jan. 2019.

[13] W. R. Ghanem, V. Jamali, Y. Sun, and R. Schober, "Resource allocation for multi-user downlink MISO OFDMA-URLLC systems," 2019, *arXiv:1910.06127*.

[14] A. A. Nasir, H. D. Tuan, H. H. Nguyen, M. Debbah, and H. V. Poor, "Resource allocation and beamforming design in the short blocklength regime for URLLC," *IEEE Trans. Wireless Commun.*, vol. 20, no. 2, pp. 1321–1335, Feb. 2021.

[15] J. Cao, X. Zhu, Y. Jiang, Y. Liu, and F.-C. Zheng, "Short frame structure optimization for industrial IoT with heterogeneous traffic and shared pilot," in *Proc. IEEE Global Commun. Conf.*, Dec. 2020, pp. 1–6.

[16] A. Ranjha and G. Kaddoum, "Quasi-optimization of uplink power for enabling green URLLC in mobile UAV-assisted IoT networks: A perturbation-based approach," *IEEE Internet Things J.*, vol. 8, no. 3, pp. 1674–1686, Feb. 2021.

[17] H. Ren, C. Pan, Y. Deng, M. Elkashlan, and A. Nallanathan, "Resource allocation for URLLC in 5G mission-critical IoT networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2019, pp. 1–6.

[18] H. Ren et al., "Joint pilot and payload power allocation for massive-MIMO-enabled URLLC IIoT networks," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 5, pp. 816–830, May 2020.

[19] Y. Polyanskiy, H. V. Poor, and S. Verdu, "Channel coding rate in the finite blocklength regime," *IEEE Trans. Inf. Theory*, vol. 56, no. 5, pp. 2307–2359, May 2010.

[20] Y. Y. Munaye, R.-T. Juang, H.-P. Lin, G. B. Tarekegn, and D.-B. Lin, "Deep reinforcement learning based resource management in UAV-assisted IoT networks," *Appl. Sci.*, vol. 11, no. 5, p. 2163, Mar. 2021.

[21] J. Wang, C. Jiang, K. Zhang, X. Hou, Y. Ren, and Y. Qian, "Distributed Q-learning aided heterogeneous network association for energy-efficient IIoT," *IEEE Trans. Ind. Informat.*, vol. 16, no. 4, pp. 2756–2764, Apr. 2020.

[22] A. Kaur and K. Kumar, "Energy-efficient resource allocation in cognitive radio networks under cooperative multi-agent model-free reinforcement learning schemes," *IEEE Trans. Netw. Service Manag.*, vol. 17, no. 3, pp. 1337–1348, Sep. 2020.

[23] H. Zhang, H. Zhang, W. Liu, K. Long, J. Dong, and V. C. M. Leung, "Energy efficient user clustering, hybrid precoding and power optimization in terahertz MIMO-NOMA systems," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 9, pp. 2074–2085, Sep. 2020.

[24] J. Ren, Z. Wang, M. Xu, F. Fang, and Z. Ding, "An EM-based user clustering method in non-orthogonal multiple access," *IEEE Trans. Commun.*, vol. 67, no. 12, pp. 8422–8434, Dec. 2019.

[25] A. Abdelnasser, E. Hossain, and D. I. Kim, "Clustering and resource allocation for dense femtocells in a two-tier cellular OFDMA network," *IEEE Trans. Wireless Commun.*, vol. 13, no. 3, pp. 1628–1641, Mar. 2014.

[26] A. Hatoum, R. Langar, N. Aitsaadi, R. Boutaba, and G. Pujolle, "Cluster-based resource management in OFDMA femtocell networks with QoS guarantees," *IEEE Trans. Veh. Technol.*, vol. 63, no. 5, pp. 2378–2391, Jun. 2014.

[27] M. Katwe, K. Singh, B. Clerckx, and C.-P. Li, "Rate-splitting multiple access and dynamic user clustering for sum-rate maximization in multiple RISs-aided uplink mmWave system," *IEEE Trans. Commun.*, vol. 70, no. 11, pp. 7365–7383, Nov. 2022.

[28] H. Mohammadi and V. Marojevic, "Artificial neuronal networks for empowering radio transceivers: Opportunities and challenges," in *Proc. IEEE 94th Veh. Technol. Conf. (VTC-Fall)*, Sep. 2021, pp. 1–5.

[29] Y. S. Nasir and D. Guo, "Multi-agent deep reinforcement learning for dynamic power allocation in wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 10, pp. 2239–2250, Oct. 2019.

[30] L. Liang, J. Kim, S. C. Jha, K. Sivanesan, and G. Y. Li, "Spectrum and power allocation for vehicular communications with delayed CSI feedback," *IEEE Wireless Commun. Lett.*, vol. 6, no. 4, pp. 458–461, Aug. 2017.

[31] *Study on Channel Model for Frequencies From 0.5 to 100 GHz*, document 3GPP TR 38.901, V.16.1.0, 2020.

[32] R. Maldonado et al., "Comparing Wi-Fi 6 and 5G downlink performance for industrial IoT," *IEEE Access*, vol. 9, pp. 86928–86937, 2021.

[33] Y. Chen, Z. Yang, J. Zhang, and M.-S. Alouini, "Further results on detection and channel estimation for hardware impaired signals," *IEEE Trans. Commun.*, vol. 69, no. 11, pp. 7167–7179, Nov. 2021.

[34] S. He, Z. An, J. Zhu, J. Zhang, Y. Huang, and Y. Zhang, "Beamforming design for multiuser URLLC with finite blocklength transmission," *IEEE Trans. Wireless Commun.*, vol. 20, no. 12, pp. 8096–8109, Dec. 2021.

[35] M. B. Ghorbel, B. Berscheid, E. Bedeer, M. J. Hossain, C. Howlett, and J. Cheng, "Principal component-based approach for profile optimization algorithms in DOCSIS 3.1," *IEEE Trans. Netw. Service Manage.*, vol. 15, no. 3, pp. 934–945, Sep. 2018.

[36] H. E. Romeijn and D. R. Morales, "A class of greedy algorithms for the generalized assignment problem," *Discrete Appl. Math.*, vol. 103, nos. 1–3, pp. 209–235, Jul. 2000.

[37] V. Mnih et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[38] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2018.

[39] A. M. Seid, G. O. Boateng, B. Mareri, G. Sun, and W. Jiang, "Multi-agent DRL for task offloading and resource allocation in multi-UAV enabled IoT edge network," *IEEE Trans. Netw. Service Manage.*, vol. 18, no. 4, pp. 4531–4547, Dec. 2021.
[40] J. K. Gupta et al., "Cooperative multi-agent control using deep reinforcement learning," in *Proc. Int. Conf. Auto. Agents Multiagent Syst.*, Cham, Switzerland, 2017, pp. 66–83.
[41] N. Naderializadeh, J. J. Sydir, M. Simsek, and H. Nikopour, "Resource management in wireless networks via multi-agent deep reinforcement learning," *IEEE Trans. Wireless Commun.*, vol. 20, no. 6, pp. 3507–3523, Jun. 2021.
[42] H. Sun, X. Chen, Q. Shi, M. Hong, X. Fu, and N. D. Sidiropoulos, "Learning to optimize: Training deep neural networks for interference management," *IEEE Trans. Signal Process.*, vol. 66, no. 20, pp. 5438–5453, Oct. 2018.
[43] M. Matthaiou, A. Papadogiannis, E. Bjornson, and M. Debbah, "Two-way relaying under the presence of relay transceiver hardware impairments," *IEEE Commun. Lett.*, vol. 17, no. 6, pp. 1136–1139, Jun. 2013.
[44] A. B. Sediq, R. H. Gohary, R. Schoenen, and H. Yanikomeroglu, "Optimal tradeoff between sum-rate efficiency and jain's fairness index in resource allocation," *IEEE Trans. Wireless Commun.*, vol. 12, no. 7, pp. 3496–3509, Jul. 2013.
[45] B. M. Lee, "Energy-efficient operation of massive MIMO in Industrial Internet-of-Things networks," *IEEE Internet Things J.*, vol. 8, no. 9, pp. 7252–7269, May 2021.
[46] T. Sandholm, K. Larson, M. Andersson, O. Shehory, and F. Tohmé, "Coalition structure generation with worst case guarantees," *Artif. Intell.*, vol. 111, nos. 1–2, pp. 209–238, Jul. 1999.

NAHED BELHADJ MOHAMED received the B.E. degree in telecommunication engineering from the Higher School of Communication of Tunis (SUP'COM), Tunisia, in 2018. She is currently pursuing the Ph.D. degree in electrical engineering program with École de technologie supérieure (ÉTS), Montreal, QC, Canada. Her research interests include wireless communications, the Internet of Things, radio resource management, and the application of machine learning in physical layer communications. She is a reviewer in several prestigious conferences, such as ICC and ICMLCN.

MD. ZOHEB HASSAN (Member, IEEE) received the Ph.D. degree from the Department of Electrical and Computer Engineering, The University of British Columbia, Vancouver, BC, Canada. He is currently an Assistant Professor with the Department of Electrical and Computer Engineering, Université Laval, Canada. Prior to joining Université Laval, he was a Senior Post-Doctoral Research Fellow at École de Technologie Supérieure (ÉTS) and a Research Assistant Professor with the Bradley Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA, USA. He has co-authored over 35 Journal articles and 20 conference papers in radio resource optimization, interference management, spectrum sharing, and optical wireless communications. He was the recipient of the NSERC Postdoctoral Fellowship Award in 2021, with a top-ranked application, and the Four-Year Fellowship at the University of British Columbia in 2014. He has served/is serving as a TPC member for various prestigious IEEE conferences, such as IEEE GLOBECOM, ICC, MILCOM, VTC, and PIMRC, and a Senior Reviewer of IEEE Open Journal of Communications Society.

GEORGES KADDOUM (Senior Member, IEEE) received the bachelor's degree in electrical engineering from École Nationale Supérieure de Techniques Avancées (ENSTA Bretagne), Brest, France, the M.S. degree in telecommunications and signal processing (circuits, systems, and signal processing) from Université de Bretagne Occidentale and Telecom Bretagne (ENSTB), Brest, in 2005, and the Ph.D. degree (Hons.) in signal processing and telecommunications from the National Institute of Applied Sciences (INSA), University of Toulouse, Toulouse, France, in 2009.

He is currently a Professor and the Tier 2 Canada Research Chair with École de Technologie Supérieure (ÉTS), Université du Québec, Montreal, Canada, and a Faculty Fellow with the Cyber Security Systems and Applied AI Research Center, Lebanese American University. In 2014, he was awarded the ÉTS Research Chair in physical-layer security for wireless networks. Since 2010, he has been a Scientific Consultant in the field of space and wireless telecommunications for several U.S. and Canadian companies. He has published more than 300 journals, conference papers, two chapters in books, and has eight pending patents.

Dr. Kaddoum received the Best Papers Awards at the 2014 IEEE International Conference on Wireless and Mobile Computing, Networking, Communications (WIMOB), with three coauthors; and the 2017 IEEE International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC), with four coauthors. Moreover, he received IEEE Transactions on Communications Exemplary Reviewer Award in 2015, 2017, and 2019. In addition, he received the Research Excellence Award of the Université du Québec in 2018. In 2019, he received the Research Excellence Award from ÉTS in recognition of his outstanding research outcomes. Finally, he received the IEEE TCSC Award for Excellence in Scalable Computing in 2022. He is currently serving as an Area Editor for IEEE Transactions on Machine Learning in Communications and Networking and an Associate Editor for IEEE Transactions on Information Forensics and Security, IEEE Transactions on Communications, and IEEE Communications Letters.