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

Received 20 June 2023; revised 24 November 2023 and 19 February 2024; accepted 24 February 2024.
Date of publication 29 February 2024; date of current version 8 March 2024.

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

Digital Object Identifier 10.1109/TMLCN.2024.3371929

Cooperate or Not Cooperate: Transfer Learning With Multi-Armed Bandit for Spatial Reuse in Wi-Fi

PEDRO ENRIQUE ITURRIA-RIVERA[1] (Student Member, IEEE), MARCEL CHENIER[2] (Member, IEEE), BERNARD HERSCOVICI[2], BURAK KANTARCI[1] (Senior Member, IEEE), AND MELIKE EROL-KANTARCI[1] (Senior Member, IEEE)

[1]School of Electrical Engineering and Computer Science, University of Ottawa, Ottawa, ON K1N 6N5, Canada
[2]NetExperience Inc., Ottawa, ON K2K 2E2, Canada

CORRESPONDING AUTHOR: P. E. ITURRIA-RIVERA (pitur008@uottawa.ca)

This work was supported in part by the Mitacs Accelerate Program and in part by NetExperience Inc.

ABSTRACT The exponential increase in the demand for high-performance services such as streaming video and gaming by wireless devices has posed several challenges for Wireless Local Area Networks (WLANs). In the context of Wi-Fi, the newest standards, IEEE 802.11ax, and 802.11be, bring high data rates in dense user deployments. Additionally, they introduce new flexible features in the physical layer, such as dynamic Clear-Channel-Assessment (CCA) thresholds, to improve spatial reuse (SR) in response to radio spectrum scarcity in dense scenarios. In this paper, we formulate the Transmission Power (TP) and CCA configuration problem with the objective of maximizing fairness and minimizing station starvation. We present five main contributions to distributed SR optimization using Multi-Agent Multi-Armed Bandits (MA-MABs). First, we provide regret analysis for the distributed Multi-Agent Contextual MABs (MA-CMABs) proposed in this work. Second, we propose reducing the action space given the large cardinality of action combinations of TP and CCA threshold values per Access Point (AP). Third, we present two deep MA-CMAB algorithms, named Sample Average Uncertainty (SAU)-Coop and SAU-NonCoop, as cooperative and non-cooperative versions to improve SR. Additionally, we analyze the viability of using MA-MABs solutions based on the ϵ-greedy, Upper Bound Confidence (UCB), and Thompson (TS) techniques. Finally, we propose a deep reinforcement transfer learning technique to improve adaptability in dynamic environments. Simulation results show that cooperation via the SAU-Coop algorithm leads to a $\mathbf{14.7\%}$ improvement in cumulative throughput and a $\mathbf{32.5\%}$ reduction in Packet Loss Rate (PLR) in comparison to non-cooperative approaches. Under dynamic scenarios, transfer learning mitigates service drops for at least $\mathbf{60\%}$ of the total users.

INDEX TERMS 802.11ax, 802.11be, deep transfer reinforcement learning, multi-agent multi-armed bandits, spatial reuse, Wi-Fi.

I. INTRODUCTION

THE exponential increase in the use of wireless technology is forecast to reach 71% of the global population with some kind of wireless service. In the group of Wireless Local Area Networks (WLANs), Wireless Fidelity (Wi-Fi) technology presents a growth of up to 4-fold over 5 years from 2018 to 2023 [1]. The current Wi-Fi standard, IEEE-802.11ax, also known as Wi-Fi 6 and its ''Extended'' version Wi-Fi 6E, are expected to form 75% of all Wi-Fi chipset shipments by early 2024. Moreover, Wi-Fi 6E is forecast to represent the 32% of all Wi-Fi chipset shipments by 2025 [2]. Additionally, the IEEE standardization body has recently been working on a new standard, namely IEEE 802.11be (or Wi-Fi 7), which is scheduled to be released by May 2024 [3]. This standard will replace IEEE 802.11ax in the years to come.

Spatial reuse (SR) has been of interest for more than 20 years in the wireless community, as it contributes to the

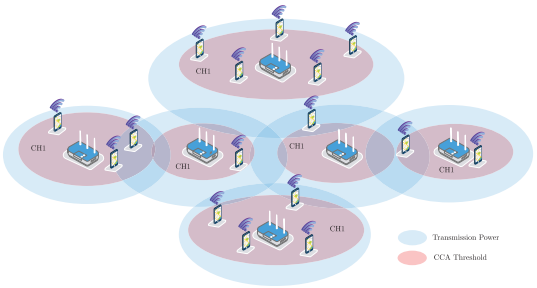

FIGURE 1. Typical operational scenario: Access Points (APs) adjust their Transmission Power and CCA threshold towards an efficient spatial reuse.

reduction of collisions among stations in medium access control [4]. As the number of dense WLAN deployments increases, SR becomes more challenging in the context of Carrier Sense Multiple Access (CSMA) technology, as used in Wi-Fi [5]. Firstly, Wi-Fi 6 was introduced to address diverse challenges, such as the increasing number of Wi-Fi users, dense hotspot deployments, and the high demand for services like Augmented, Mixed, and Virtual Reality. It included additional features, such as dynamic adjustment of the Clear Channel Assessment (CCA) threshold and Transmission Power (TP). Before Wi-Fi 6, the CCA threshold configuration was a static value per Access Point (AP), causing inefficient channel utilization in dense Wi-Fi deployments [6]. Additionally, adjusting TP allows the reduction of interference among the APs and consequently maximizes network performance [7]. Thus, SR and network performance can be positively improved by adjusting CCA and TP. However, the complex interactions between CCA and TP call for intelligent configuration of both. As part of the forthcoming IEEE 802.11be standard, Coordinated Spatial Reuse (CSR) has become one of several proposals to improve the current 802.11ax. Unlike the uncoordinated version introduced in 802.11ax, CSR requires inter-AP feedback to combat interference with neighboring APs [8]. Additionally, Wi-Fi 7 will enable to widening of the channel bandwidth to a substantial 320 MHz, elevating the modulation rate to an impressive 4096 QAM, Multi-Link Operation (MLO), and Multiple Resource Units (MRU) capabilities.

To this end, data scarcity and data access are key for any Machine Learning (ML) method [9]. Recently, AI-based wireless networks have attracted researchers in both Wi-Fi [10], [11], [12] and 5G domains [13]. However, the proposed solutions usually require complete availability of the data. In reality, data access is not always feasible due to privacy restrictions. Recent wireless network architectures have started to shift towards a more open and flexible design. In 5G networks, as well as in the O-RAN Alliance architecture, AI support is provided to orchestrate primary network functions [14]. In the context of Wi-Fi, OpenWiFi [15], released by Telecom Infra as a novel project to divide the Wi-Fi technology stack by utilizing open-source software for the cloud controller and an AP firmware operating system. These paradigm changes enable the development of many applications in the area of ML, and more specifically, in Reinforcement Learning (RL) applications.

In this paper, we intend to optimize TP and CCA thresholds to improve SR and overall network Key Performance Indicators (KPIs). More importantly, we aim to investigate whether cooperation significantly impacts SR by running a thorough requirement analysis for the newly proposed features such as CSR in Wi-Fi networks. To do so, we formulate the TP and CCA configuration problem with the objective of maximizing product network fairness and minimizing station starvation. We model our proposed solution as a distributed multi-agent decision-making problem and use a Multi-Agent Multi-Armed Bandit (MA-MAB) approach to solve it. This work differs from the existing solutions in the literature with the following five contributions:

1) We present the regret analysis for the distributed non-cooperative contextual MA-MAB (MA-CMAB) version of Sample Average Uncertainty-Sampling (SAU). SAU builds on a deep Contextual MAB.
2) We reduce the inherently huge action space given the possible combinations of TP and CCA threshold values per AP. We derive our solution via worst-case interference analysis.
3) We analyze the performance of the network KPIs of well-known distributed MA-MAB implementations such as ϵ-greedy, UCB, and Thompson on the selection of the TP and CCA values in cooperative and non-cooperative settings.
4) We introduce an MA-CMAB in cooperative and non-cooperative settings along with a thorough performance analysis.
5) To the best of our knowledge, for the first time in the literature, we propose a deep transfer learning-based solution to adapt TP and CCA parameters efficiently in dynamic scenarios.[1]

With these contributions, our simulation results show that the ϵ-greedy MAB solution improves throughput by at least 44.4%, provides a 12.2% improvement in terms of fairness, and achieves a 94.5% reduction in Packet Loss Ratio (PLR) over typical configurations when a reduced set of actions is known. Furthermore, when compared with non-cooperative approaches with a full set of actions, the SAU-Coop algorithm is shown to improve the throughput and PLR by 14.7% and 32.5%, respectively. Moreover, our proposed transfer learning-based approach reduces service drops by at least 60%.

The rest of the paper is organized as follows. Section II presents a summary of recent work that uses Machine Learning to improve SR in Wi-Fi. Section III covers the basics of Multi-Armed Bandits, including deep contextual

[1] In this work, the term dynamic scenarios is used to define settings where variations occur in the user load per AP. The user load refers to the number of users communicating to one AP. Thus, we aim to mimic real-life situations.

bandits, an analysis of the regret of the proposed algorithm, and an introduction to deep transfer reinforcement learning. In Section IV, we present our system model along with an analysis to reduce the action space via worst-case interference. Section V presents the proposed schemes, and the results are discussed in Section VI. Finally, Section VII concludes the paper.

II. RELATED WORK

Reinforcement learning-based spatial reuse has been of interest in recent literature. The studies have focused on distributed solutions with no cooperation or centralized schemes of multi-armed bandits. These studies are summarized below.

In [16], the authors present a comparison among well-known MABs such as ϵ-greedy, UCB, Exp3, and Thompson sampling in the context of decentralized spatial reuse via Dynamic Channel Allocation (DCA) and Transmission Power Control (TPC) in WLANs. The results showed that ''selfish learning'' in a sequential manner presents better performance than ''concurrent learning'' among the agents.

Additionally, [17] presents a centralized MAB that consists of an optimizer based on a modified Thompson Sampling (TS) algorithm and a sampler based on the Gaussian Mixture (GM) algorithm to improve spatial reuse in 802.11ax Wi-Fi. More specifically, to cope with the large action space that consists of TP and Overlapping BSS/Preamble-Detection (OBSS/PD) thresholds, the authors utilize a MAB variant, namely the Infinitely Many-Armed Bandit (IMAB). Furthermore, a distributed solution based on Bayesian optimizations of Gaussian processes to improve spatial reuse is proposed in [18].

Other solutions not related to reinforcement learning can be found in the literature with the aim of improving spatial reuse in WLANs. For instance, in [19], the authors propose a distributed algorithm where the APs decide their Transmission Power based on their Received Signal Strength Indicator (RSSI). Moreover, in [20], the authors present an algorithm to improve spatial reuse by utilizing diverse metrics such as SINR, proximity information, RSSI, and Basic Service Set (BSS) color and compare it to the legacy algorithms. The ultimate goal of the previous algorithm is the selection of the channel state (IDLE or BUSY) at the time of an incoming frame given the previous metrics. Finally, the authors in [21] present a supervised federated learning approach for spatial reuse optimization.

In all of the above works, the authors employ either centralized or decentralized schemes with no cooperation to address SR optimization in Wi-Fi. In this paper, we aim to bridge this gap via a coordination-based MA-MAB. Additionally, we tackle some of the issues previously encountered in other works, such as the size of the action space due to the set of possible values of TP and CCA. Finally, to the best of our knowledge, we propose to address SR adaptation in dynamic environments by utilizing deep transfer learning for the first time.

III. BACKGROUND

In this section, we present background information on Multi-Armed Bandits, including ϵ-greedy, Upper Confidence Bound, Thompson sampling bandits, and an introduction to contextual MABs with a focus on a neural network-based contextual bandit. Additionally, we introduce MABs to the multi-agent setting, and we conclude with background information on deep transfer reinforcement learning.

Multi-Armed Bandits (MABs) are a widely used RL approach that addresses the exploration-exploitation trade-off problem. Their implementation is usually simpler when compared with full RL off-policy or on-policy algorithms. It is important to note that there is some overlap between bandit algorithms and RL algorithms. In fact, multi-armed bandit problems can be seen as a simpler case of RL problems. One of the main differences is concerned with the exploration-exploitation dilemma. Indeed, both deal with this problem, but bandits typically focus solely on this trade-off, whereas RL algorithms often deal with more complex environments where the agent can learn from the consequences of its actions. Another important characteristic is the sequential decision-making nature of both algorithms. Bandit algorithms typically rely on independent, one-shot decisions, without taking into account long-term consequences. This is a quite convenient feature in scenarios where learning must be performed rapidly. In RL, the agent's decisions can have long-term consequences and influence the subsequent states and rewards it encounters. Finally, bandit algorithms are simpler complexity-wise, which makes them appealing in scenarios where computational resources are scarce, as considered in our work. However, simplicity often comes at the expense of suboptimal solutions [22].

The basic model of MABs corresponds to the stochastic bandit, where the agent has K possible actions to choose, called arms, and receives a certain reward R as a consequence of pulling the k^{th} arm over T environment steps [23]. In [24], the author classifies MABs according to their rewards by asking the following question: where do the rewards come from? The rewards can be modeled as independent and identically distributed (i.i.d), adversarial, constrained adversarial, or random-process rewards. Out of these four models, the following two are more commonly found in the literature: the i.i.d and the adversarial models. In the i.i.d model, each pulled arm's reward is drawn independently from a fixed but unknown distribution D_k with an unknown mean μ_k^*. On the other hand, in the adversarial model, each pulled arm's reward is randomly sampled from an adversary or an alien to the agent (such as the environment) and is not necessarily sampled from any distribution [25].

The performance of MABs is measured in terms of cumulative regret $R(T)$ or total expected regret over the T steps. Regret quantifies the missed opportunity in a multi-armed bandit problem, computed as the difference between the expected reward attainable by an oracle that selects the optimal arm at each time step and the actual reward obtained by

a given policy. Thus, the regret for an MAB can be formally defined as,

$$R(T) = \mu^* T - \sum_{k>1}^{K} \mu_k \mathbb{E}[n_k(T)], \tag{1}$$

where μ_k is the mean of the D_k random reward distribution, $\mu^* = \max\{\mu_1, \ldots, \mu_K\}$, and $n_k(t)$ is the number of times the k^{th} arm has been chosen at time t. The utmost goal of the agent is to minimize $R(T)$ over the T steps, such that $\lim_{T\to\infty} R(T)/T = 0$, which means the agent will identify the action with the highest reward within such limit.

A. ϵ-GREEDY, UPPER-CONFIDENCE-BOUND (UCB), AND THOMPSON SAMPLING MAB

The ϵ-greedy MAB is one of the simplest MABs, and as the name suggests, it is based on the ϵ-greedy policy. In this method, the agent selects greedily the best arm most of the time, and once in a while, with a predefined small probability (ϵ), it selects a random arm [26].

The UCB MAB tackles some of the disadvantages of the ϵ-greedy policy at the moment of selecting non-greedy arms. Instead of drawing randomly, the UCB policy measures how promising non-greedy arms are close to optimal. In addition, it takes into consideration the rewards' uncertainty in the selection process. The selected arm is obtained by drawing the action from $\texttt{argmax}_a \left[Q_t(a) + c\sqrt{\ln t / N_t(a)}\right]$, where $N_t(a)$ corresponds to the number of times that action a via the k^{th} arm has been chosen, and $Q_t(a)$ is the Q-value of action a [26], [27].

Finally, Thompson Sampling MAB action selection builds on the Thompson Sampling algorithm. Thompson sampling or posterior sampling is a Bayesian algorithm that constantly constructs and updates the distribution of the observed rewards given a previously selected action. This allows the MAB to select arms based on the probability of how optimal the chosen arm is. The parameters of the distribution are updated based on the selection of the distribution class [28].

B. DEEP CONTEXTUAL MULTI-ARMED BANDITS

Contextual Multi-Armed Bandits (CMABs), as a variant of MABs, observe a series of features (i.e., context) before selecting an arm [22]. Fig. 2 depicts the difference between the stateless MAB and CMAB. Different from the stateless MAB, a CMAB is expected to relate the observed context with the feedback or reward obtained from the environment in T episodes and consequently predict the best arm given the received features [25]. Thus, context plays a crucial role in contextual bandits because it provides valuable information that can help the algorithm make more informed decisions and ultimately lead to better rewards. Diverse CMABs have been proposed throughout the literature, such as LinUCB, Neural Bandit, Contextual Thompson Sampling, and Active Thompson Sampling [22]. More recently, a deep neural contextual bandit named SAU-Sampling has been presented in [29], where the context is related to the rewards using neural networks. The details of SAU-Sampling will be discussed in the following sections.

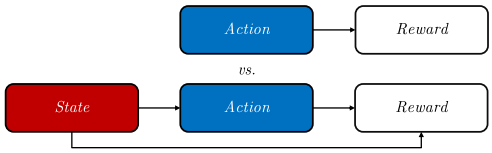

FIGURE 2. **MAB vs. contextual MAB.**

C. MULTI-AGENT MULTI-ARMED BANDITS (MA-MABs)

In this subsection, we intend to formally introduce the Multi-Agent Multi-Armed Bandit (MA-MAB) problem. An MA-MAB is the multi-agent variant of an MAB where several agents pull their arms and obtain feedback from the environment [30]. MA-MABs can be modeled as centralized or distributed. In centralized settings, the actions of the agents are taken by a centralized controller, and in distributed settings, each agent independently chooses its actions. Distributed decision-making settings scale more effectively [31] and naturally deal easily with a large set of arms (K) when compared to centralized settings that suffer from the cardinality explosion of K arms.

In this work, we consider a distributed scenario defined as follows. Let us consider a contextual K-armed bandit with N players or agents that form a team. It is worth noting that we use M to denote the number of APs, and consequently $N = M$. However, in this section, we use N to describe the number of agents in any MA-MAB problem. $X_{n,k}(t)$ denotes the reward obtained by the n^{th} agent by pulling its k^{th} arm. Additionally, let $\mu_{n,k}$ denote the mean of $X_{n,k}(t)$. If only one agent is considered, it can be assumed that the reward yields to $X_{n,k}(t) = Y_{n,k}(t)$, where $Y_{n,k}(t)$ is a random variable that models the reward of the environment. We assume a non-collision scheme [32], [33], [34] where if more than two agents play the same arm k at time t the yielded reward is not affected by that specific condition. Unlike the work in [35] where channel access is studied, our problem becomes non-trivial since agents can select the same spatial reuse parameters and still have viable performance. Thus, we assume that the decisions of the agents may or may not lead to a conflict, and the perceived reward can change according to the action decision of the team. Thus, $X_{n,k}(t) = \hat{Y}_{n,k}(t)$, where $\hat{Y}_{n,k}(t)$ is a random variable that models the reward obtained from the environment when multiple agents are interacting with the environment. All agents choose their arms concurrently in this setting. Let us define the action taken by agent n at time t by $k_n(t) \in \mathcal{A} := \{1, .., K\}$. In addition, the context of each agent observed by agent n at time t is defined by $c_n(t) \in \mathcal{C} := \{C_1, .., C_Q\}$, where Q is the number of contextual attributes used in the CMAB. Consequently, the history seen by agent n at time t corresponds to $\mathcal{H}_n(t) = \{k_n(1), c_n(1), X_{n,k_{n(1)}}(1), \ldots, k_n(t), c_n(t), X_{n,k_{n(t)}}(t)\}$. The final policy for agent n corresponds to $\pi_n : \mathcal{H}_n(t) \to \mathcal{A}$ where

$\pi_n = (k_n(t))_{t=1}^{\infty}$. Finally, the action system vector can be defined as bipartite matching $\mathbf{k}^* = \{(k_1, \ldots, k_N) : k_n \in \mathcal{A}\}$. The end goal is to maximize the system reward in two distributed approaches: non-cooperative and cooperative. The expected sum of rewards corresponds to $\mathbb{E}_\pi[\sum_{t=1}^{T} X_{\pi(t)}(t)]$. If the means $\mu_{n,k}$ are known, the set of actions can be selected by picking a bipartite matching as,

$$\mathbf{k}^{**} \in \underset{k \in \mathbf{k}^*}{\operatorname{argmax}} \sum_{n=1}^{N} \mu_{n,k_n}, \tag{2}$$

where $\mathbf{k}^{**}$ is the optimal but not unique bipartite matching.

The solution to the proposed problem would be finding a maximum weight matching on a labeled bipartite graph between the number and actions of the agents. A bipartite matching in this context refers to a set of edges in the bipartite graph such that no two edges share a common vertex. In this case, actions are chosen by selecting a specific arm in each of the N agents comprising the MA-MAB. Since each action is taken concurrently by all agents, the condition of bipartite matching is always fulfilled. The selection of the bipartite matching can be represented as a combinatorics problem where the permutation of each arm in a multi-agent scenario is mapped as,

$$\mathbf{k}^*(t) \in [1, P(K, N)], \tag{3}$$

where $\mathbf{k}^*(t)$ is the bipartite matching selected at time t and $P(K, N)$ corresponds to the number of permutations given K arms and N agents. Finally, the expected regret of a MA-MAB can be defined as,

$$R(T) = T\mu_{k^{**}} - \mathbb{E}_\pi \left[\sum_{t=1}^{T} X_{\pi(t)}(t) \right], \tag{4}$$

where $\mu_{k^{**}}$ corresponds to the optimal team policy to select $\mathbf{k}^{**}$ bipartite matching.

D. ANALYSIS OF REGRET

The expected regret of a single agent obtained with the SAU-Sampling algorithm has a logarithmic nature (*Proof:* See [29, Appendix A.8]) as its counterparts TS and UCB, and shows an improvement in comparison to the linear regret nature of the ϵ-greedy MAB [24]. Building on and extending the work of [29] and [35] along with the considerations in [36] and [37], we can obtain the expected regret for our distributed SAU-Sampling MA-CMAB algorithm as follows:

Theorem 1: If SAU-Sampling MA-CMAB is run on a distributed K-armed N-agents bandit problem ($K \geq 2$, $N \geq 1$) having an arbitrary reward distribution $X_{1,1}, \cdots, X_{N,K}$ with support in $[0, 1]$, then its expected regret after n number of plays is at most[2]*:*

$$R(T) \leq N^2 K \Delta_{k^{**}} \left(\frac{96 \log n}{\Delta_{k^{**}}^2} + \frac{1}{1+N} \right), \tag{5}$$

[2]A proof of Theorem 1 is provided in Appendix A.

TABLE 1. Notations.

Notation	Definition
s and $\mathcal{S}$	Index and set of stations
m and $\mathcal{M}$	Index and set and the number of APs
$x^{\lvert\mathcal{S}\rvert}$ and $y^{\lvert\mathcal{M}\rvert}$	Stations' positions and AP's positions
N_S	Total number of nodes
P_{cs}^m	CCA threshold of m^{th} AP
P_{tx}^m	Transmission Power of m^{th} AP
$R_T^{(s,m)}$	Throughput of s^{th} STA of m^{th} AP
$R_A^{(s,a)}$	Achievable throughput of s^{th} STA of m^{th} AP
$D_l^{(m,s)}$	Adaptive data link rate of s^{th} STA of m^{th} AP
$u_{IDLE}^{(s,m)}$	Binary function, $u_{IDLE}^{(s,m)} = 1$ if STA s is idle with respect to its m^{th} BSS and $u_{IDLE}^{(s,m)} = 0$, otherwise
$u_{SUCC}^{(s,m)}$	Binary function, $u_{SUCC}^{(s,m)} = 1$ if STA s successfully transmits a packet to its m^{th} AP and $u_{SUCC}^{(s,m)} = 0$, otherwise
$\phi_p^{(s,m)}$	Corresponds to the set of binary variables that define each s^{th} STA transmission properties to their m^{th} AP, where $\phi_p^{(s,m)} := \{u_{IDLE}^{(s,m)}, u_{SUCC}^{(s,m)}, \xi_{CCA}^{(s,m)}, \xi_{ED}^{(s,m)}\}$
$\xi_{CCA}^{(s,m)}$	Binary function, $\xi_{CCA}^{(s,m)} = 1$ if the sensed energy signal of a packet sent by the s^{th} STA to the m^{th} AP is below the CCA threshold (P_{cs}^m), and $\xi_{CCA}^{(s,m)} = 0$, otherwise
$\xi_{ED}^{(s,m)}$	Binary function, $\xi_{ED}^{(s,m)} = 1$ if the sensed energy signal of a packet sent by the s^{th} STA to the m^{th} AP is below the Energy Detection (ED) threshold (P_{ed}^m), and $\xi_{ED}^{(s,m)} = 0$, otherwise
$\xi_{STA}^{(s,m)}$	Binary function, $\xi_{STA}^{(s,m)} = 1$ if s^{th} station's throughput is greater than $\omega R_A^{(s,a)}$ and $\xi_{STA}^{(s,m)} = 0$, otherwise
$\mathbb{E}(T_g^{(s,m)})$ and $\mathbb{E}(I_g^{(s,m)})$	Expected length of general time-slot and expected information transmitted by the s^{th} STA of m^{th} AP
T_{TXOP}, T_{EDCA}	Packet transmission duration and time required for a successful Enhanced Distributed Channel Access (EDCA) transmission
$\delta^{(s,m)}$	Packet transmission duration and time required for a successful Enhanced Distributed Channel Access (EDCA) transmission
$\bar{P}^{fair}$ and $\bar{U}$	Average linear product-based network fairness and average station starvation
ω, $g^{(s,m)}$ and σ^2	Fraction of $R_A^{(s,a)}$ in which STAs are considered in starvation, the channel power gain and the power noise
P_{tx}^m and P_{tx}^r	The transmission power at the m^{th} transmitter (AP) and the received signal strength at the r^{th} receiver
$d_{r,m}$ and θ	Distance between the m^{th} transmitter and r^{th} receiver and path loss exponent
$\mathcal{F}_+^m$ and $\mathcal{F}_-^m$	Subset of interferers and non-AP interferers
$\gamma^{(r,m)}$, $C^{(r,m)}$ and C_T	Worst-case SINR and Shannon's maximum capacity of m^{th} transmitter and r^{th} receiver and cumulative maximum network capacity

where n is the number of times each agent in SAU-Sampling MA-CMAB chooses an action. Finally, $\Delta_{k^{**}}$ is defined as $\Delta_{k^{**}} = \mu_{n,k_n^{**}} - \mu_{n,k_n}$.

Corollary 1: The upper bound of the regret in the distributed and non-cooperative solution is polynomial in K and N, respectively, and exhibits logarithmic behavior in the time domain.

According to Corollary 1, the regret will suffer a polynomial and logarithmic increase, which is highly problematic, especially if K or N grows. This motivates subsection IV-A, where we reduce the $P(K, N)$ by analytically finding a reduced set of arms to alleviate the poor performance of such regret. On the other hand, we define the expected regret for the cooperative scenario by,

$$R(T) = TX_k - \mathbb{E}_\pi \left[\sum_{t=1}^{T} X_{\pi(t)}(t) \right] + \mathbb{E}[Cost], \qquad (6)$$

where $\mathbb{E}[Cost]$ is the expected cost of communication among agents. To further obtain the upper bounds in the cooperative case, a similar procedure should be followed as in [31] and [38], where agents can communicate an estimate of their reward. However, we leave the derivation of this regret outside the scope of this work due to its lengthy characteristic and empirically prove cooperation superiority when compared to the non-cooperative case. The goal in both scenarios is to find the set of policies for each agent in the distributed algorithm to minimize the regrets presented in Eq. (4) and Eq. (6). In this work, we consider two main approaches: distributed non-cooperative and cooperative MA-MABs with adversarial rewards. According to [24] adversarial rewards can be defined as "rewards can be arbitrary as if they are chosen by an adversary that tries to fool the algorithm. The adversary may be oblivious to the algorithm's choices, or adaptive thereto." The distributed, arbitrary nature and the reward dependency of a single agent of others make our reward model inherently adversarial. Next, we introduce deep transfer learning techniques and their application in this work.

E. DEEP TRANSFER REINFORCEMENT LEARNING

Transfer learning or knowledge transfer techniques improve learning time efficiency by utilizing prior knowledge. Typically, this is done by extracting the knowledge from tasks of one or multiple sources and then applying such knowledge in a target task [39]. If the tasks are related in nature and the target task benefits positively from the acquired knowledge from the source, then it is called inductive transfer learning [40]. This type of learning is not uncommon and it is used by the human brain on a daily basis. However, a phenomenon called negative transfer can occur if the target task performance is negatively affected following the knowledge transfer [41].

In the realm of transfer learning, we can find Deep Transfer Learning (DTL). DTL is a subset of transfer learning that studies how to utilize knowledge in deep neural networks. In the context of classification/prediction tasks, large volume of data is required to properly train the model of interest [42]. In many practical applications where training time is essential to respond to new domains [43], retraining with a large volume of data is not always feasible and possibly catastrophic

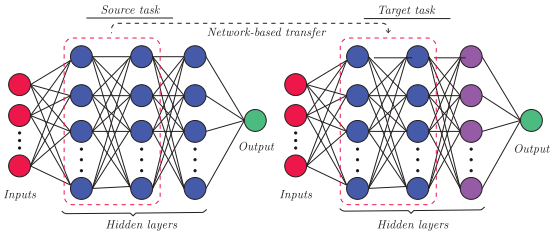

FIGURE 3. Network-based transfer learning: hidden layers of the neural network source task are re-utilized in the target network.

in terms of performance. "What to transfer?" defines one of the main research questions in transfer learning. Specifically, in the case of deep transfer learning, four categories have been broadly identified: instance-based transfer, where data instances from a source task are utilized; mapping-based transfer, where a mapping of two tasks is used on a new target task; network-based transfer, where the network pre-trained model is transferred to the target task; and adversarial-based transfer, where an adversarial model is employed to find which features from diverse source tasks can be transferred to the target task [44]. In this work, we utilize the DTL form called network-based transfer learning to adapt efficiently TP and CCA parameters in dynamic scenarios. An example of a network-based transfer learning technique is presented in Fig. 3. Such a technique is utilized in deep transfer reinforcement learning as part of a transfer learning type called policy transfer [45]. In particular, policy transfer takes a set of source policies $\pi_{S_1}, \ldots, \pi_{S_K}$ that are trained on a set of source tasks and uses them in a target policy π_T in a way that is able to leverage the former knowledge from the source policies to learn its own. More specifically, the weights and biases that comprise each of the hidden layers of the source policies are the elements transferred to the target policies. Note that in practice policies are modeled as neural networks. Furthermore, we take advantage of the design of a contextual multi-armed bandit presented in [29] and apply policy transfer to improve the agent's SR adaptability in dynamic environments. The results and observations of applying DTRL are discussed in section VI-E. In the next section, we discuss the details of the system model and present an analysis of reducing the cardinality of the action space in the proposed SR formulation.

IV. SYSTEM MODEL AND PROBLEM FORMULATION

In this work, we consider an infrastructure mode Wi-Fi 802.11ax network $\mathcal{N}$ with $N_S = |\mathcal{S}| + |\mathcal{M}|$ nodes where $\mathcal{S}$ is the set of stations with $\{\boldsymbol{x}^1, \boldsymbol{x}^2, \ldots, \boldsymbol{x}^{|\mathcal{S}|}\} \in \mathbb{R}^2$ positions and $\mathcal{M}$ is the set of APs with $\{\boldsymbol{y}^1, \boldsymbol{y}^2, \ldots, \boldsymbol{y}^{|\mathcal{M}|}\} \in \mathbb{R}^2$ positions. We can assume that $|\mathcal{M}|$ APs positions correspond to cluster centers and the stations will attach to their closest AP. In addition, the list of notations utilized in this work can be found in Table 1.

In this paper, we improve SR via maximization of the linear product-based fairness and minimization of the number

of stations under starvation by configuring TP and CCA parameters.

$$\begin{aligned} &Max && \begin{pmatrix} \text{fairness} \\ \text{avg. station starvation complement} \end{pmatrix} && \text{(7a)} \\ &s.t. && \text{Throughput,} && \text{(7b)} \\ &var. && \text{Transmission power and CCA threshold,} \\ & && \text{Idle and success transmission indicators,} \\ & && \text{EDCA Idle and success transmission duration.} && \text{(7c)} \end{aligned}$$

Each s^{th} STA is defined by a set of binary variables of transmission properties to their m^{th} AP, where $\phi_p^{s,m} := \{u_{IDLE}^{(s,m)}, u_{SUCC}^{(s,m)}, \xi_{CCA}^{(s,m)}, \xi_{ED}^{(s,m)}\}$. Let us define the binary variable $u_{IDLE}^{(s,m)}$ of an STA being idle in a BSS as:

$$\sum_{m \in \mathcal{M}} u_{IDLE}^{(s,m)} = 1 \qquad \forall s \in \mathcal{S}, \tag{8}$$

where $u_{IDLE}^{(s,m)} = 1$ if s^{th} STA is idle with respect to its AP and $u_{IDLE}^{(s,m)} = 0$, otherwise. In addition, we proceed to define the binary variable $u_{SUCC}^{(s,m)}$ in which an STA will successfully transmit a packet as,

$$\sum_{s \in \mathcal{S}} u_{SUCC}^{(s,m)} u_{IDLE}^{(s,m)} \xi_{CCA}^{(s,m)} \xi_{ED}^{(s,m)} \leq 1 \quad \forall m \in \mathcal{M}, \tag{9}$$

$$\sum_{m \in \mathcal{M}} \xi_{CCA}^{(s,m)} = 1 \quad \forall s \in \mathcal{S}, \tag{10}$$

$$\sum_{m \in \mathcal{M}} \xi_{ED}^{(s,m)} = 1 \quad \forall s \in \mathcal{S}, \tag{11}$$

where $\xi_{CCA}^{(s,m)} = 1$ if the sensed energy signal of a packet sent by the s^{th} STA to the m^{th} AP is below the CCA threshold (P_{cs}^m), otherwise becomes zero. Here, $\xi_{ED}^{(s,m)} = 1$ if the sensed energy signal of a packet sent by the s^{th} STA to the m^{th} AP is below the Energy Detection (ED) threshold (P_{ed}^m), otherwise becomes zero. Additionally, we consider $P_{cs}^m = P_{ed}^m$.

The CCA threshold is a predefined signal strength level that a channel must fall below for a device to consider the channel clear and decide to transmit. Pragmatically speaking, if the signal strength on a channel is above the CCA threshold, a device assumes that the channel is busy, and it refrains from transmitting to avoid interference with ongoing transmissions. On the other hand, if the signal strength falls below the CCA threshold, the device assumes that the channel is clear and proceeds with transmission. The rationale for this threshold is to help avoid collisions and interference by ensuring that a device does not transmit when it detects ongoing transmissions on the channel.

While CCA is primarily used to assess whether the channel is clear for the initiation of Wi-Fi transmissions, the ED threshold is used to detect the overall energy level on a channel, including both Wi-Fi and non Wi-Fi signals. Devices use energy detection to determine if a channel is busy or idle. If the energy level is above the ED threshold, the channel is assumed to be busy, and devices may defer their transmissions. If the energy level is below the ED threshold, the channel is assumed to be clear, and devices may proceed with transmissions. In dynamic channel selection scenarios, where Wi-Fi networks may change channels dynamically to avoid interference, the ED threshold is used to make decisions about channel switching. Even though ED and CS thresholds are different in terms of what type of energy they track, both minimize interference, avoid collisions, and improve the overall reliability and performance of wireless communication by managing channel access and transmission.

P_{ed}^m by definition has a higher value than P_{cs}^m since it is firstly checked by the AP to set an STA on IDLE status when compared to the received power. Afterwards, the AP checks P_{cs}^m. Thus, we could assume that for all APs at least the condition $P_{cs}^m \leq P_{ed}^m$ holds and consequently we can simplify our analysis with equality.

As indicated in [46], two procedures can be defined when stations are using ECDA. The first one indicates the duration of a successful EDCA transmission whereas the second one indicates the duration of a collision. The first procedure can be generally described as:

$$T_{EDCA}^s = T_{TXOP} + SIFS + T_P + ACK + AIFS, \tag{12}$$

where T_{TXOP} defines the transmission duration, SIFS corresponds to the short inter-frame space, ACK denotes the time to send an acknowledgment signal, AIFS represents the arbitration inter-frame space, and T_P stands for the propagation delay. On the other hand, we can define the second procedure as the duration of a collision. The previous event can be calculated as:

$$T_{EDCA}^c = T_{TXOP} + T_P + AIFS. \tag{13}$$

All of the variables in Eqs. (12) and (13) are in the μs order with exception of T_{TXOP} which falls in the order of ms. Thus, with the following approximation, $T_{EDCA}^c \approx T_{EDCA}^s$, we can denote both values as T_{EDCA}.

As indicated by [47], the expected conditional length of the general time-slot $\mathbb{E}(T_g)$ and the expected conditional transmitted information $\mathbb{E}(I_g)$ by the s^{th} STA to m^{th} AP can be expressed as:

$$\begin{aligned} \mathbb{E}(T_g^{(s,m)} | u_{IDLE}^{(s,m)}) &= \delta^{(s,m)} u_{IDLE}^{(s,m)} \\ &\quad + u_{IDLE}^{(s,m)} T_{ECDA} \quad \forall m \in \mathcal{M}, s \in \mathcal{S}, \end{aligned} \tag{14}$$

$$\mathbb{E}(I_g^{(s,m)} | u_{SUCC}^{(s,m)}) = u_{SUCC}^{(s,m)} D_l^{(s,m)} T_{TXOP} \quad \forall m \in \mathcal{M}, s \in \mathcal{S}, \tag{15}$$

where $D_l^{(s,m)}$ denotes the link data rate, T_{EDCA} is a variable that corresponds to the time required for a successful Enhanced Distributed Channel Access (EDCA) transmission and $\delta^{(s,m)}$ represents the duration of an idle time slot. The link data rate adaptively depends on SNR [48] and is mapped based on the SNR/BER curves [49]. The received SNR can be defined as $P_{tx}^m g^{(s,m)} / \sigma^2$ where P_{tx}^m is the transmission power, $g^{(s,m)}$ is the channel power gain and σ^2 is the power noise.

According to Bianchi's work that is commonly used as a reference study [50], the value of the throughput R can be calculated as follows:

$$R = \frac{\mathbb{E}[\text{payload information transmitted in a slot time}]}{\mathbb{E}[\text{length of a slot time}]}. \tag{16}$$

Analogously, we can define the throughput of the s^{th} station attached to the m^{th} AP as:

$$\begin{aligned} R_T^{(s,m)} &= \frac{\mathbb{E}(I_g^{(s,m)}|u_{IDLE}^{(s,m)})}{\mathbb{E}(T_g^{(s,m)}|u_{SUCC}^{(s,m)})} \\ &= \frac{u_{SUCC}^{(s,m)} D_l^{(s,m)} T_{TXOP}}{\delta^{(s,m)} u_{IDLE}^{(s,m)} + u_{IDLE}^{(s,m)} T_{EDCA}}. \end{aligned} \tag{17}$$

Additionally, let us define the average linear product-based network fairness and average station starvation in a distributed setting by,

$$\bar{P}^{fair}(t) \leq 1, \tag{18}$$

$$\sum_{s\in\mathcal{S}} \xi_{STA}^{(s,m)}(t) = 1 \quad \forall m \in \mathcal{M}, \tag{19}$$

$$\bar{P}^{fair}(t) = \frac{1}{|\mathcal{M}|} \sum_{m\in\mathcal{M}} \prod_{s\in\mathcal{S}} \frac{R_T^{(s,m)}(t)}{R_A^{(s,a)}}, \tag{20}$$

$$\bar{U}(t) = \frac{1}{|\mathcal{M}|} \sum_{m\in\mathcal{M}} \frac{1}{|\mathcal{S}|} \sum_{s\in\mathcal{S}} \xi_{STA}^{(s,m)}(t), \tag{21}$$

where $R_A^{(s,a)}$ is the achievable throughput of the s^{th} station attached to the m^{th} AP. In practice, the value of $R_A^{(s,a)}$ is set by running simulations with one STA and gathering the corresponding throughput upper bound. Additionally, $\xi_{STA}^{(s,m)} = 1$ if s^{th} station's throughput is greater than $\omega R_A^{(s,a)}$ where the station starvation ratio, $\omega \in (0, 1]$, otherwise becomes zero, in which case the station is considered in starvation. ξ_{STA} measures the number of starving stations at t time. It uses throughput that behaves as a random variable due to many factors such as congestion, packet loss, and varying levels of network activity. Due to the property of *induced randomness*, ξ_{STA} takes the form of a random variable as well.

The considered problem is a multi-objective problem and can be addressed with the weighted sum approach. Thus, in each time step, the problem can be formulated as,

Problem 1:

$$\max_{\mathbf{P_{tx}},\mathbf{P_{cs}}} \; B_1 \bar{P}^{fair}(t) + B_2(1 - \bar{U}(t)) \tag{22}$$

$$\text{s.t. (8)-(19)}, \tag{23}$$

$$P_{tx}^m \in [P_{tx}^{min}, P_{tx}^{max}], P_{cs}^m \in [P_{cs}^{min}, P_{cs}^{max}] \quad \forall m \in \mathcal{M}, \tag{24}$$

where B_1 and B_2 are the importance coefficients associated to the variables $\bar{P}^{fair}$ and the complement of $\bar{U}(t)$, respectively.

As specified in equation (22), Problem 1 requires resolution in each time slot t. The parameters chosen in a given time slot and their outcomes are contingent not only on the selected transmission power and CCA threshold at time t but also on decisions made in the past and uncertain future events. Consequently, we can conceptualize this problem as a sequential decision problem, wherein actions optimizing performance are taken across a sequence of steps. This behavior is dictated by our setup, where each AP independently selects its parameters, sharing varying degrees of information or no information at all. Due to the dynamic nature of the scenario, the binary variables of each STA $\phi_p^{(s,m)}$ have a probabilistic nature, requiring an additional step to map them to EDCA parameters [47]. Instead, we simplify our analysis by utilizing a network simulator to model such dynamics. We propose to solve the previous mixed fractional and stochastic polynomial problem by using an MA-CMAB solution, as described in Section V.

A. OPTIMAL ACTION SET VIA WORST-CASE INTERFERENCE

Motivated by subsection III-D, we aim to find a reduced action set via worst-case interference analysis. Wi-Fi typical scenarios consist of APs and stations distributed non-uniformly. Contrary to the analysis presented in [51], we aim to obtain an optimal subset of TP and CCA threshold values to further reduce the action space size that is utilized in the optimization of SR. In this analysis, we only consider the Carrier Sense (CS) threshold term as a form of the CCA threshold.

First, let us consider the worst-case interference scenario in an arrangement with $N_S \geq 2$. For the sake of simplicity, we use the path-loss radio propagation model:

$$P_{rx}^r = \frac{P_{tx}^m}{d_{r,m}{}^{\theta}}, \tag{25}$$

where P_{tx}^m and P_{rx}^r are the TP at the m^{th} transmitter (AP) and the received signal strength at the r^{th} receiver, respectively. In addition, $d_{r,m}$ is the distance between the transmitter and receiver. Finally, $\theta \in [2, 4]$ corresponds to the path loss exponent. Thus, from the perspective of the m^{th} AP, the worst-case interference I^m is defined as:

$$I^m = \sum_{v\in\mathcal{F}_+^m} \frac{P_{tx}^v}{X^{(m,v)\theta}} + P_{tx}^{sta} \sum_{w\in\mathcal{F}_-^m} \frac{1}{X^{(m,w)\theta}}, \tag{26}$$

where $\mathcal{F}_+^m$ is the subset of interference sources $|\mathcal{F}_+^m| = |\mathcal{M}| - 1$, corresponding to APs interfering with the m^{th} AP, and $\mathcal{F}_-^m$ is the subset of non-AP interference sources $|\mathcal{F}_-^m| = |\mathcal{S}|$, corresponding to the stations interfering with the m^{th} AP. Furthermore, P_{tx}^v is the TP of the v^{th} interference source, and P_{tx}^{sta} is a constant corresponding to the fixed power assigned to all the stations, based on the fact that typically stations are not capable of modifying their TP. Additionally, $X^{(m,v)}$ and $X^{(m,w)}$ correspond to the distance from the m^{th} AP to the v^{th} AP interference source and m^{th} AP to the w^{th} station interference

source, respectively. $X^{(m,.)}$ is calculated as,

$$X^{(m,.)} = \sqrt{(D^m_{CCA} + x^{(m,.)})^2 + d^2_{r,m} - \frac{2(D^m_{CCA} + x^{(m,.)})}{(d_{r,m}\cos\varsigma_{r,.})^{-1}}}, \tag{27}$$

where (.) refers either to the AP or non-AP interference source, D^m_{CCA} is the CCA threshold range of the m^{th} AP, $\varsigma_{r,.}$ is the distance between the receiver to the interference source (.) and $x^{(m,.)}$ corresponds to the distance between any (.) interference source and D^m_{CCA}.

The corresponding worst-case SINR $\gamma^{(r,m)}$ at the receiver is defined as:

$$\gamma^{(r,m)} = \frac{P^m_{tx}}{{d_{r,m}}^{\theta}(I^m + N_0)}. \tag{28}$$

Let us assume that $N_0 \ll I^m$, thus the equation is reduced to:

$$\gamma^{(r,m)} = \frac{P^m_{tx}}{{d_{r,m}}^{\theta} I^m}. \tag{29}$$

By substituting equations (26) and (27) in (29), we obtain equation (30), as shown at the bottom of the page. The previous mentioned equation describes $\gamma^{(r,m)}$ in function of D^m_{CCA} and $d_{r,m}$. Additionally, we substitute $D^m_{CCA} = \left(P^m_{tx}/T^m_{cs}\right)^{1/\theta}$ in equation (30) obtaining,

$$\gamma^{(r,m)} = \frac{\frac{P^m_{tx}}{{d_{r,m}}^{\theta}}}{\sum_{v\in\mathcal{F}^+_m} \frac{P^m_{tx}}{\Gamma^m + P^{sta}_{tx}\sum_{w=1}^{K^-_m}\iota^{(m,w)}}}, \tag{31}$$

where,

$$\Gamma^m = \left(\sqrt{\left[\left(\frac{P^m_{tx}}{T^m_{cs}}\right)^{\frac{1}{\theta}} + x_{m,v}\right]^2 + d^2_{r,m} - H_1}\right)^{\theta},$$

$$H_1 = 2\left[\left(\frac{P^m_{tx}}{T^m_{cs}}\right)^{\frac{1}{\theta}} + x_{m,v}\right] d_{r,m}\cos\varsigma_{r,v},$$

$$\iota^{(m,w)} = \left(\sqrt{(\Omega_{sta} + x_{m,w})^2 + d^2_{r,m} - \frac{2(\Omega_{sta} + x_{m,w})}{(d_{r,m}\cos\varsigma_{r,w})^{-1}}}\right)^{-\theta},$$

$$\Omega_{sta} = \left(\frac{P^{sta}_{tx}}{T^{sta}_{cs}}\right)^{\frac{1}{\theta}}.$$

Hereafter, we proceed to define the maximum channel capacity in terms of TP and Carrier Sense (CS) threshold (T_{cs}). Given a certain value of SINR, the Shannon maximum capacity is expressed as:

$$C^{(r,m)} = W\log_2(1 + \gamma^{(r,m)}), \tag{32}$$

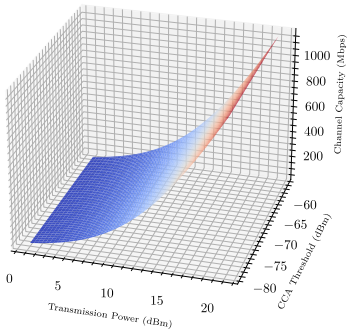

FIGURE 4. Network capacity as a function of TP and CS threshold.

where W is the channel bandwidth in Hz. Then, the cumulative maximum network capacity can be calculated as:

$$C_T = \sum_{m=1}^{|\mathcal{M}|-1}\sum_{r=1}^{N} C^{(r,m)}. \tag{33}$$

In Fig. 4, a graph of the network maximum capacity is shown as a function of TP and CS threshold. As observed, the network capacity achieves its higher values when a combination of high TP and low CS threshold is utilized. Note that, prior knowledge of the locations is required.

V. PROPOSED MULTI-AGENT MULTI-ARMED BANDIT ALGORITHMS

In this section, we present the action space, context definition, and reward function for the MA-MAB algorithms utilized in this work.

A. ACTION SPACE

In this work, we consider a discrete action space that corresponds to the number of combinations of P_{cs} and P_{tx}. In the context of MABs, this translates into the number of arms for each MAB agent. The action space is defined as,

$$A_{cs} = \{P^{\min}_{cs}, P^{\min}_{cs} + \frac{P^{\max}_{cs} - P^{\min}_{cs}}{L_{cs} - 1}, \ldots, P^{\max}_{cs}\}, \tag{34}$$

$$A_{tx} = \{P^{\min}_{tx}, P^{\min}_{tx} + \frac{P^{\max}_{tx} - P^{\min}_{tx}}{L_{tx} - 1}, \ldots, P^{\max}_{cs}\}, \tag{35}$$

where $P^{\min}_{cs}$, $P^{\max}_{cs}$, $P^{\min}_{tx}$, and $P^{\max}_{tx}$ are the minimum and maximum values of the CCA threshold and TP values, respectively. L_{cs} and L_{tx} denote the number of levels used to discretize the CCA threshold and TP values, respectively. Finally, the number of arms corresponding to the action space for the m^{th} agent, K^{AP}_m, is given by $|A^m_{cs}| \cdot |A^m_{tx}|$.

$$\gamma^{(r,m)} = \frac{\frac{P^m_{tx}}{{d_{r,m}}^{\theta}}}{\sum_{v\in\mathcal{F}^+_m} \frac{P^m_{tx}}{\left(\sqrt{(D^m_{CCA}+x_{m,v})^2 + d^2_{m,r} - \frac{2(D^m_{CCA}+x_{m,v})}{(d_{r,m}\cos\varsigma_{r,.})^{-1}}}\right)^{\theta}} + P^{sta}_{tx}\sum_{w\in\mathcal{F}^m_-} \frac{1}{\left(\sqrt{(D^m_{CCA}+x_{m,w})^2 + d^2_{r,m} - \frac{2(D^m_{CCA}+x_{m,w})}{(d_{r,m}\cos\varsigma_{r,w})^{-1}}}\right)^{\theta}}} \tag{30}$$

B. REWARD FUNCTION IN DISTRIBUTED NON-COOPERATIVE SETTINGS

The reward is defined following the optimization problem 1. Defined in a distributed manner, it resembles the reward presented in [17], which includes a linear product-based fairness and the starvation term of a station [17], [20]. A station is considered to be in starvation when its performance is below a predefined percentage of its achievable throughput. The reward is defined as,

$$r_{AP}^{m} = \frac{|\Psi_{AP}^{m}| \prod_{j \in \Psi_{AP}^{m}} \frac{R_T^{(s,m)}}{\omega R_A^{(s,m)}} + H_2}{N_{AP}^{m}(N_{AP}^{m}+1)},$$
$$H_2 = |N_{AP}^{m} \setminus \Psi_{AP}^{m}| \left(N_{AP}^{m} + \prod_{j \in N_{AP}^{m} \setminus \Psi_{AP}^{m}} \frac{R_T^{(s,m)}}{R_A^{(s,m)}} \right), \quad (36)$$

where Ψ_{AP}^{m} is the set of starving stations attached to the m^{th} AP, N_{AP}^{m} the set of stations attached to the m^{th} AP. We can also observe that $r_{AP}^{m} \propto C^{(r,m)}$ as defined in Eq. (32).

In the next subsection, we present the definition of the context considered in our MA-CMAB solution.

C. DISTRIBUTED SAMPLE AVERAGE UNCERTAINTY-SAMPLING MA-CMAB

In [29], the authors present an efficient contextual multi-arm bandit based on a "frequentist approach" to compute uncertainty instead of using Bayesian solutions such as Thompson Sampling. The frequentist approach consists of measuring the uncertainty of the action-values based on the sample average rewards just computed, rather than relying on the posterior distribution given the past rewards. In this work, we introduce multi-agent cooperative and non-cooperative variants of the previously mentioned RL algorithm.

Our problem definition describes the context with only the local observations of the APs:

1) Number of starving stations, $|\Psi_{AP}^{m}|$, where m corresponds to the m^{th} AP under ω fraction of their attainable throughput during episode t.
2) Average RSSI, $\overline{SAP}^{m}$, where m is the m^{th} AP during episode t.
3) Average Noise, $\overline{\Upsilon AP}^{m}$, where m denotes the m^{th} AP during episode t.

Additionally, the context is normalized as,

$$\psi_{AP}^{m} = |\Psi_{AP}^{m}|/N_{AP}^{m}, \quad (37)$$

$$s_{AP}^{m} = \begin{cases} 0, & -50 \text{ dBm} \leq \overline{S}_{AP}^{m} \leq -60 \text{ dBm}, \\ 0.25, & -60 \text{ dBm} \leq \overline{S}_{AP}^{m} \leq -70 \text{ dBm}, \\ 0.5, & -70 \text{ dBm} \leq \overline{S}_{AP}^{m} \leq -80 \text{ dBm}, \\ 0.75, & -80 \text{ dBm} \leq \overline{S}_{AP}^{m} \leq -90 \text{ dBm}, \\ 1, & -90 \text{ dBm} \geq \overline{S}_{AP}^{m}, \end{cases} \quad (38)$$

$$\hat{\Upsilon}_{AP}^{m} = \overline{\Upsilon}_{AP}^{m}/100. \quad (39)$$

The SAU-Sampling MA-CMAB algorithm, in its non-cooperative version (SAU-NonCoop), is described in Algorithm 1. The algorithm begins with the initialization of action-value functions $\mu(\boldsymbol{x}m|\hat{\boldsymbol{\theta}}m)$ as deep neural networks and the exploration parameters $J_{m,k}^2$ and $n_{m,k}$ for each m^{th} AP. Here, $n_{m,k}$ represents the number of times action a was selected in the m^{th} AP, and $J_{m,k}^2$ is defined as an exploration bonus. In each environmental step (Algorithm 1, `line 2`), each agent observes its local context and computes the selected arm based on the reward prediction. In (Algorithm 1, `line 11`), each CMAB agent updates $\hat{\boldsymbol{\theta}}_{m,k}$ using stochastic gradient descent on the loss between the predicted reward and the real observed reward. Finally, the exploration parameters are updated accordingly, given the prediction error, as depicted in (Algorithm 1, `line 12`).

Algorithm 1 SAU-Sampling MA-CMAB

1 **Initialize** network $\hat{\boldsymbol{\theta}}_m$, exploration parameters $J_m^2(t=0)=1$ and $n_m(t=0)=0$ for all actions $k \in K_m$.
2 **for** *environment step* $t \leftarrow 1toT$ **do**
3 **for** *agent* m **do**
4 Observe context $\boldsymbol{x}_m(t) = [\psi_{AP}^{m}(t), s_{AP}^{m}(t), \hat{\Upsilon}_{AP}^{m}(t)]$
5 **for** $k = 1, \ldots, K_m$ **do**
6 Calculate reward prediction $\hat{\mu}_{i,t}(t) = \mu(\boldsymbol{x}_m|\hat{\boldsymbol{\theta}}_m)$ and $\tau_{m,k}^2(t) = J_{m,k}^2/n_{m,k}$
7 $\tilde{\mu}_{m,k} \sim \mathcal{N}(\hat{\mu}_{m,k}, n_{m,k}^{-1}\tau_{m,k}^2)$
8 **end**
9 Compute $a_m(t) = \texttt{argmax}_a(\{\tilde{\mu}_{m,k}(t)\}_{a \in K_m})$ if $t > K_m$, otherwise $a_m(t) \sim \mathcal{U}(0, K)$;
10 Select action $a_m(t)$, observe reward r_{AP}^{m};
11 Update $\hat{\boldsymbol{\theta}}_{m,k}$ using SGD with gradients $\partial l_m/\partial\theta$ where $l_m = 0.5(r_{AP}^{m} - \hat{\mu}_{m,k}(t))$;
12 Update $J_{m,k}^2 \leftarrow J_{m,k}^2 + e_m^2$ using prediction error $e_m = r_{AP}^{m}(t) - \hat{\mu}_{m,k}(t)$ and $n_{m,k} \leftarrow n_{m,k} + 1$;
13 **end**
14 **end**

D. COOPERATIVE SAMPLE AVERAGE UNCERTAINTY-SAMPLING MA-CMAB

In this section, we present a cooperative version of SAU-Sampling MA-CMAB named SAU-Coop, which differs from the non-cooperative version by having the total reward r_C^{m} consider the network Jain's fairness index in addition to their local reward r_{AP}^{m} as:

$$r_C^{m} = r_{AP}^{m} + r_{\mathcal{J}}, \quad (40)$$

where $r_{\mathcal{J}}$ as the overall network Jain's fairness index is defined as,

$$r_{\mathcal{J}} = \mathcal{J}(R^1, \ldots, R^M) = \frac{(\sum_{m=1}^{|\mathcal{M}|} R^m)^2}{|\mathcal{M}| \cdot \sum_{m=1}^{|\mathcal{M}|} (R^m)^2}, \quad (41)$$

Algorithm 2 Reward-Cooperative ϵ-Greedy MA-MAB

Initialize $\epsilon_m(t=0)=\epsilon_0$, $Q_{m,k}(t=0)\leftarrow 0$, $N_{m,k}(t=0)\leftarrow 0$ and β.
for *environment step* $t \leftarrow 1toT$ **do**
  for *agent* m **do**
    Execute action $a_m(t)$:
    $a_m(t) = \begin{cases} \texttt{argmax}_{k=1,\dots,K} r_{k,i}(t) & \text{with probability } 1-\epsilon_m(t) \\ k \sim \mathcal{U}(0,K) & \text{o.w} \end{cases}$
    Calculate reward $r^m_{AP}(t)$ based on feedback of the environment
    Update $Q_{m,k}(t+1) = Q_{m,k}(t) + \frac{1}{N_m(t)}[(r^m_{AP} + \beta \cdot \frac{1}{M-1}\sum_{m=1}^{M-i} r^m_{AP}) - Q_{m,k}(t)]$
    Update $N_m \leftarrow N_m(t)+1$;
    Update $\epsilon_m \leftarrow \frac{\epsilon_m(t)}{\sqrt{t}}$
  end
end

Algorithm 3 SAU-Sampling MA-CMAB Transfer Learning

Function Detect_Singularity($\mathcal{K}$) ; `// returns True if anomaly is detected in network KPIs data` $\mathcal{K}$ `at time t, and False otherwise.`
Let $\mathcal{L} = \{l | l \in \mathbb{N}, l > 0\}$ the set of layers of model $\hat{\boldsymbol{\theta}}^l_{m,k}$ and $\mathcal{M} \subset \mathcal{L}$ the subset of layers to be transferred.
Run algorithm SAU-Sampling MA-CMAB (`Algorithm 1`)
while *environment step* $t < T$ **do**
  if ¬Detect_Singularity **then**
    continue;
  else
    Reset exploration parameters $S^2_{m,k}, n_{m,k}$;
    Reinitialize weights w and biases b of the l^{th} layer of $\hat{\boldsymbol{\theta}}^{l\notin\mathcal{M}}_{m,k}$ via: $v_l = \left(\sqrt{|\hat{\boldsymbol{\theta}}^{l\notin\mathcal{M}}_{m,k}|}\right)^{-1}$;
    $\hat{\boldsymbol{\theta}}^{l\notin\mathcal{M}}_{m,k}(w,b) \rightarrow w_l \sim \mathcal{U}(-v_l, v_l), b_l \sim \mathcal{U}(-v_l, v_l)$;
    Transfer weights and biases via:
    $\hat{\boldsymbol{\theta}}^{l\in\mathcal{M}}_{m,k}(w,b) \rightarrow \hat{\boldsymbol{\theta}}^{l\in\mathcal{M}'}_{m,k}(w,b)$;
  end
end

where $R^m = \sum_{s=1}^{|\mathcal{S}^m|} R_T^{(s,m)}$ is the total throughput of all $|\mathcal{S}^m|$ stations of the m^{th} AP.

E. REWARD-COOPERATIVE ϵ-GREEDY MA-MAB

In addition to the previous cooperative algorithm, we propose a cooperative approach based on the classical ϵ-greedy strategy [26] that considers a percentage of the average reward of other agents in the reward update of the action. This procedure is described in Algorithm 2.

Finally, in the next subsection, we present the details of the DTRL scheme to improve SR adaptation in dynamic environments.

F. SAMPLE AVERAGE UNCERTAINTY-SAMPLING MA-CMAB BASED DEEP TRANSFER REINFORCEMENT LEARNING

Typically, RL agents learn their best policy based on the feedback received from the environment over a T horizon time. However, in real-world scenarios, environmental conditions can vary at $T+1$, and thus, adapting to the updated environment is necessary [52]. In such cases, the ''outdated'' policy of the agent may not be optimal to adjust to the new conditions efficiently. For instance, a modification in the distribution of the stations over the APs can cause the SR-related parameters chosen by the ''outdated'' agent's policy to affect network performance.

To address the previous situation, we propose two main solutions: **1.** If the agent detects a change in the environment indicated by a singularity, it will decide to correct its configuration via forgetting the policy already learned (**forget**) or **2.** adapting the agent's policy to the new conditions via a transfer learning technique. A singularity is defined as an anomalous behavior of the KPIs of interest after the policy of each CMAB agent has converged. In this work, we don't delve into how to detect a singularity, and we assume the existence of an anomaly detector in our system [53]. In Algorithm 3, we present the transfer learning algorithm depicting the second proposed solution. At $t = 0$, each SAU-Sampling MA-CMAB agent will reset their weights and biases and start learning as part of Algorithm 1. At $t = S1$, where $S1$ corresponds to the time when an anomaly is detected and the transfer procedure is activated (Algorithm 3, `line 7`). In our setup, we transfer $l = 2$ and reset $l = 1$ (Algorithm 3, `line 11`), where l corresponds to the layer of the neural network utilized in the SAU-Sampling MA-CMAB agent. However, as indicated (Algorithm 3, `line 13`), the transfer is not constrained to one layer but more generally to a set of layers. The set of transferred layers is considered a hyperparameter to be tuned. Partial transfer of a model avoids negative transfer by giving the agent room to adapt to the new context since it mitigates model overfitting.

VI. PERFORMANCE EVALUATION

In this section, we intend to demonstrate the positive impact on network KPIs of the reduced action set derived in Section IV-A, as presented in Subsections VI-B and VI-C. Further in Subsection VI-D, we present a comparison between the cooperative and non-cooperative versions of our proposed algorithm along with a comparison against the two baselines. Furthermore, we leverage a transfer learning approach to avoid starvation in dynamic environments in Subsection VI-E. Finally, in Section VI-F, we present a complexity analysis of the MA-CMAB algorithm.

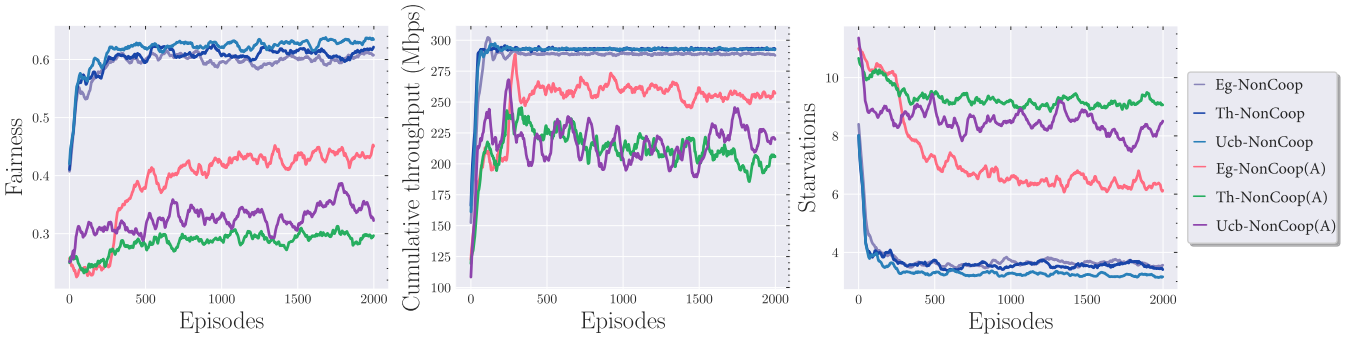

FIGURE 5. Convergence performance of ϵ-greedy (Eg-NonCoop), Thompson Sampling (Th-NonCoop) and UCB (Ucb-NonCoop) MA-MABs under non-cooperative and distributed regimen. (A) indicates the usage of the full set of actions.

TABLE 2. Learning hyperparameters.

Parameter	Value
ϵ-greedy MAB	Annealing ϵ: $\sqrt{T}$
Thompson Sampling MAB	Prior distribution: Beta
Upper Confidence Bound MAB	Level of exploration, $c = 1$
SAU-Sampling	Number of hidden layers, $N_h = 2$
	Number of neurons per hidden layer, $n_h = 100$
	Number of inputs, $\|x_m(t)\| = 3$ and number of outputs, $N_o = K$
	Batch size, $B_s = 64$
	Optimizer : RMSProp (8e-3)
	Weight decay : 5e-4
	Activation function : ReLU
Gym environment step time	0.05 s

TABLE 3. Network settings.

Parameter	Value
Number of APs	6
Number of Stations	15
Number of antennas (AP)	2
Max Supported Tx Spatial Streams	2
Max Supported Tx Spatial Streams	2
Channel Number [3]	1
Propagation Loss Model	Log Distance Propagation Loss Model
Wi-Fi standard	802.11 ax
Frequency	5 GHz
Channel Bandwidth	80 MHz
Traffic Model - UDP application	$[0.011, 0.056, 0.11$ [55], $0.16]$ Gbps
Maximum & minimum Transmission Power	$P_{tx}^{max} = 21.0$ dBm & $P_{tx}^{min} = 1.0$ dBm
Maximum & minimum CCA threshold	$P_{cs}^{max} = -62.0$ dBm & $P_{cs}^{min} = -82.0$ dBm
	$K_{cs} = 1$ and $K_{tx} = 1$
Station starvation ratio	$\omega = 1$

A. SIMULATION SETTINGS

We consider two scenarios in our simulations. The first one considers stationary users whereas the second scenario considers mobile users to model dynamic scenarios (see Section VI-E). In addition, each station and AP are devices that are equipped with two antennas supporting up to two spatial streams in transmission and reception. In this work, we assume a frequency of 5 GHz with an 80 MHz channel bandwidth in a Line of Sight (LOS) setting.[3] The propagation loss follows the Log-Distance propagation loss model with a constant speed propagation delay. We implement our proposed solutions in ns-3, and we also use OpenAI Gym to interface between ns-3 and the MA-MAB solution [54]. To ensure the validity of the proposed algorithms, we conduct simulations using various seed values, resulting in random deployment positions for all users and affecting the traffic dynamics in ns-3. We consider an adaptive rate data mode with UDP downlink traffic, which entails that there is no guarantee of data delivery, ordering, or duplicate protection. Thus, users' packet collisions are also random and governed by the simulations. Additionally, the time at which each user starts transmitting is randomly chosen based on the utilized seed. The number of runs using different seeds is set at 10. In Table 2 and Table 3, we present the learning hyperparameters and network settings parameters, respectively.

[3]We assume that all APs are configured to use one channel out of the available 11. This is a practical selection to create dense deployment scenarios.

B. REDUCED SET OF ACTIONS VS. ALL ACTIONS

In subsection IV-A, we have presented a mathematical analysis to obtain a reduced set of optimal actions with the goal of decreasing exploration time and consequently improving convergence time. As concluded in Fig. 4, high TP and low CCA threshold values maximize the network capacity in the simulation scenario under study. Therefore, we select a fixed value of CCA threshold ($P_{cs} = -82.0$ dBm) and a reduced set of TP ($P_{tx} \in 15, 16, 17, 18, 19, 20, 21$ dBm) and observe the performance against the full set of possible actions described in V-A.

In Fig. 5, we present the convergence performance of three MA-MAB algorithms under UDP traffic of 0.056 Gbps under non-cooperative and cooperative settings (indicated with "**Non-coop**" and "**Coop**," respectively). These algorithms are to ϵ-greedy (Eg-NonCoop), UCB (Ucb-NonCoop), and Thompson Sampling (Th-NonCoop) MA-MABs. For each algorithm, three convergence graphs are plotted to present fairness, cumulative throughput, and station starvation representing the behavior when a reduced set of actions and the full action set (indicated with (A)) are used, respectively. Under the set of optimal actions, while no remarkable change in the performance is observed, only a slight improvement is obtained when MAB-Thompson Sampling is utilized. A noticeable improvement can be observed under the full

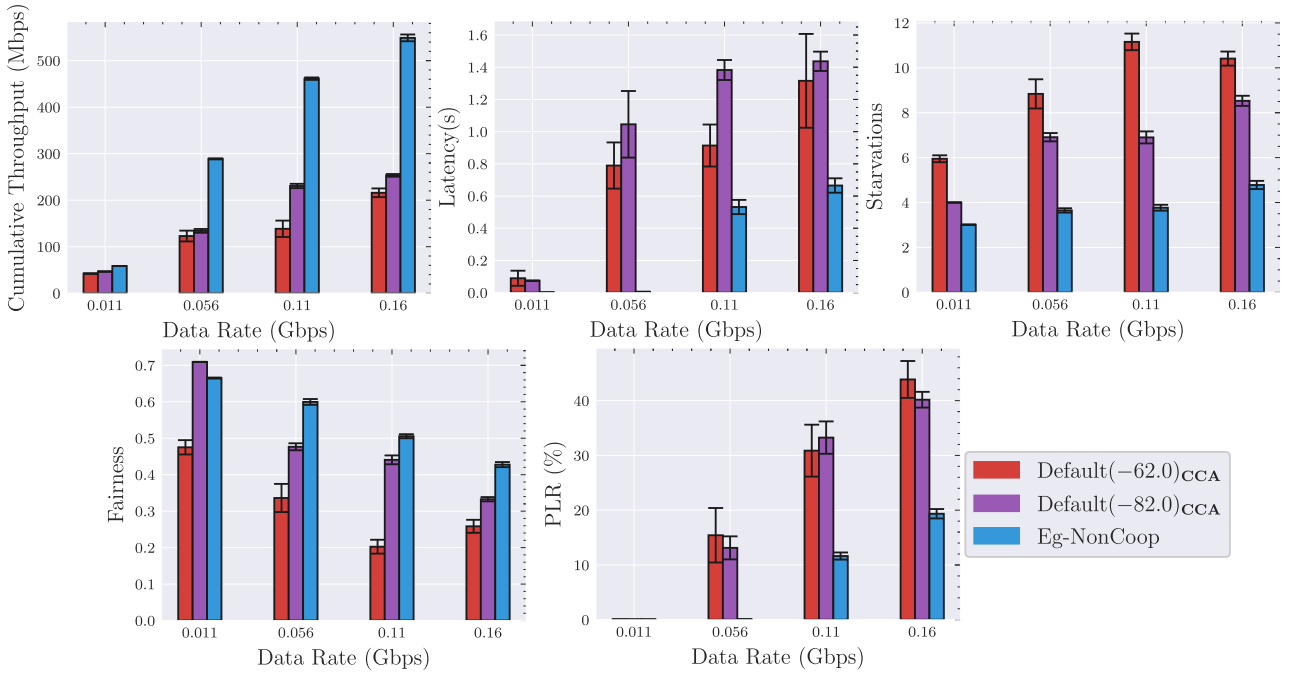

FIGURE 6. **Performance results: ϵ-greedy MAB w/ optimal set vs. default configuration with $P_{CS} \in \{-62.0, -82.0\}$ dBm.**

action set and with the MAB ϵ-greedy algorithm. In [56], the authors study the unreasonable behavior of greedy algorithms when K is sufficiently large. They conclude that when K increases above 27 arms, intelligent algorithms are significantly affected by the exploration stage. The former results validate ours based on the fact that $K = |A_{cs}| \cdot |A_{tx}| = 21^2$. Finally, it can be noted that the impact of utilizing reduced optimal actions in terms of convergence time and KPI maximization. The set of optimal tasks allows reducing station starvation when compared to the best performer Eg-NonCoop by an average of two starving users. However, to obtain such a set, prior knowledge of stations and the geographical locations of the APs is required. In the following section, we compare the results of ϵ-greedy MA-MAB to a default typical configuration without machine learning.

C. DISTRIBUTED ϵ-GREEDY MA-MAB VS. DEFAULT CONFIGURATION PERFORMANCE RESULTS

In this subsection, we present the comparative results and advantages of utilizing a distributed intelligent solution such as MAB ϵ-greedy over the default CCA threshold and TP configuration with no ML. In Fig. 6, we present the performance under four different UDP data traffic regimes: $\{0.011, 0.056, 0.11, 0.16\}$ Gbps. We consider two typical configurations of the CCA threshold: -82.0 dBm and -62.0 dBm. In both cases, the AP's TP is 16.0 dBm. It can be observed that MAB ϵ-greedy achieves significant improvement over the default configuration ($P_{cs} = -82.0$ dBm) with an average gain of 44.4% over all of the considered data rates in terms of cumulative throughput. Furthermore, it leads to an improvement of 70.9% in terms of station starvation, 12.2% in terms of fairness, 138.0% in terms of latency, and 94.5% in terms of packet loss ratio (PLR). Additionally, a gain over the default configuration ($P_{cs} = -62.0$ dBm) with an average gain of 53.9% in terms of cumulative throughput, 138.4% in terms of station starvation, 43.0% in terms of fairness, 84.0% in terms of latency, and 105.4% in terms of packet loss ratio (PLR) is shown over all the considered data rates.

D. COOPERATION VS. NON-COOPERATION PERFORMANCE RESULTS

In the two past subsections, we have shown the results considering the set of optimal actions. In this subsection, we assume the absence of stations and APs location information and thus, we must rely on the full set of actions. Consequently, we investigate if cooperation can improve the KPIs of interest by utilizing the cooperative proposal of the MAB ϵ-greedy algorithm (Rew-Coop) and the SAU-Sampling MA-CMAB algorithm (SAU-Coop). Additionally, we present two non-cooperative algorithms: SAU-NonCoop, which represents the non-cooperative version of the SAU-Sampling MA-CMAB, and Eg-NonCoop, which refers to the MAB ϵ-greedy algorithm utilized in the previous section.

As observed in Fig. 7, simulation results show that SAU-Coop improves Eg-NonCoop with an average of 14.7% in terms of cumulative throughput, 21.3% in terms of station starvation, 4.64% in terms of network fairness, 36.7% in terms of latency, and 32.5% in terms of PLR under all considered data rates. Similarly, SAU-NonCoop presents an enhanced performance over Eg-NonCoop, indicating that context is beneficial to solving the current optimization problem. Additionally, SAU-Coop exhibits performance

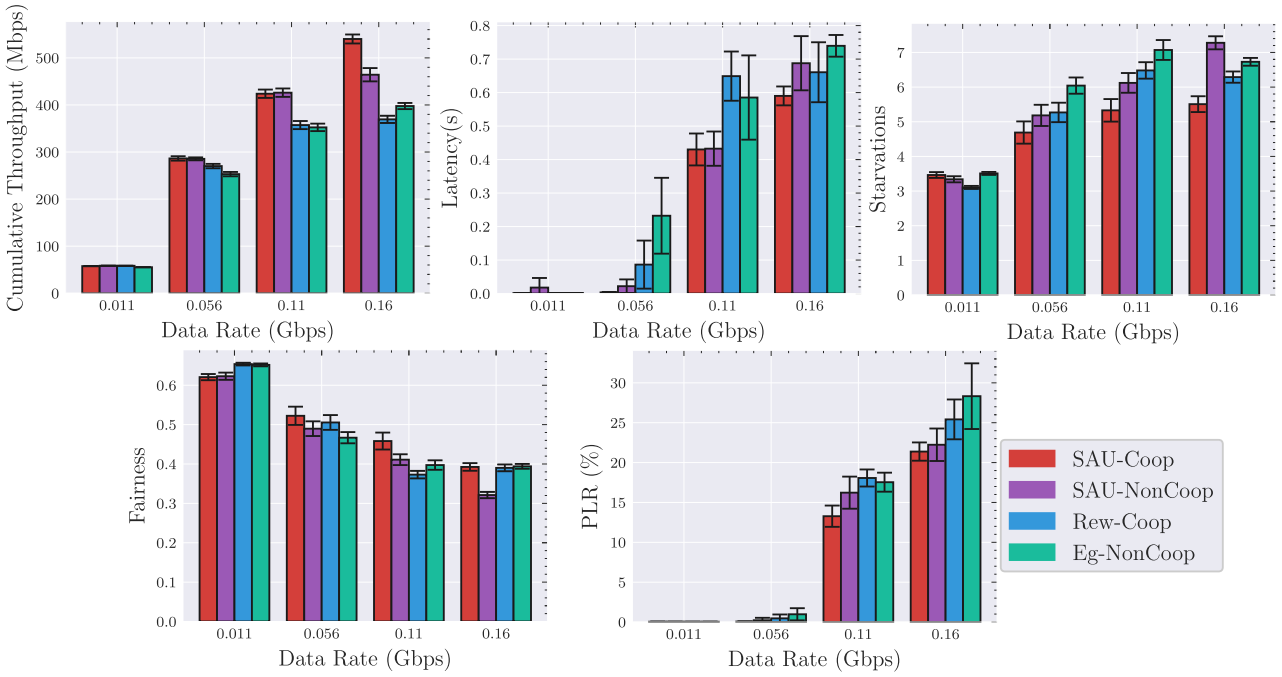

FIGURE 7. Performance results of cooperative algorithms: ϵ-greedy MA-MAB (Rew-Coop), SAU-Sampling MA-CMAB (SAU-Coop) and non-cooperative versions of the previous algorithms SAU-NonCoop and Eg-NonCoop under full-set of actions.

TABLE 4. Dynamic scenario load distribution.

	$t = 0$ min	$t = 3$ min	$t = 6$ min
$\mathcal{L}_1$	8	5	2
$\mathcal{L}_2$	5	5	2
$\mathcal{L}_3$	2	5	11

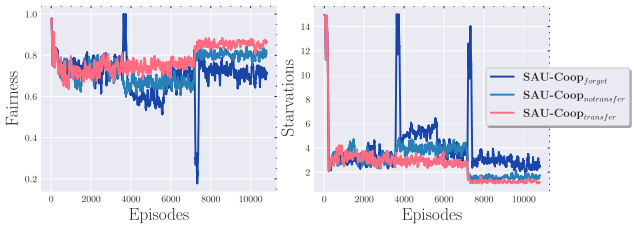

FIGURE 8. Network response in terms of fairness and station starvation when utilizing the forget, full transfer, and transfer strategies.

improvement over its non-cooperative version, especially when the data rate increases up to 0.16 Gbps where it leads to a gain of 14.1% in terms of cumulative throughput, 32.1% in terms of station starvation, 18.2% in terms of network fairness, 16.5% in terms of latency, and 4% in terms of PLR. To sum up, cooperative approaches contribute positively to the improvement of SR in Wi-Fi over non-cooperative approaches. Furthermore, in cases where cooperation is not possible, it is advisable to utilize contextual multi-armed bandits over stateless multi-armed bandits.

E. DEEP TRANSFER LEARNING IN ADAPTIVE SR IN DYNAMIC SCENARIOS RESULTS

To model a dynamic scenario, we design a simulation environment where the users move across the simulated terrain and attach to the AP that offers the best signal quality. Consequently, the user load in each AP varies, reflecting the dynamics of the environment. We model this scenario with 3 APs and 15 users, where the load changes twice throughout the simulation. As depicted in Table 4, the user load of the m^{th} AP, denoted as $\mathcal{L}_m$, change at two instances in time: at the 3^{rd} and 6^{th} minutes, respectively.

In Fig. 8, we present the network behavior in terms of fairness and station starvation under the scenario depicted in Table 4. In addition to the two methods previously introduced (**forget** and **transfer**), we present the performance of a third approach called **full transfer**, where complete transfer of the model is considered. During the first interval ($0 - 3$ min), the performance is similar for all three methods, as expected. However, following the two changes in the network load, singularities in each graph become visible in terms of fairness and starvation.

Specifically, the **forget** method exhibits the worst behavior, with a 54.3% decrease in station starvation and an 11.7% decrease in fairness compared to the **transfer** method. The **forget** method shows peaks at the moments of singularities, impacting 60% of the total users with a service drop. This behavior is inherently related to the agents' process of restarting learning and cannot be avoided. From a quality of service perspective, such disturbances are highly undesirable.

Meanwhile, the **full transfer** method is outperformed by the **transfer** method, resulting in an 18.7% decrease in station starvation and a 6% decrease in fairness. Notably, in the second interval under study (3 − 6 min), the **forget** method outperforms the **full transfer** method by the end of the period. This is due to the negative transfer effect that results from transferring the entire model. Partial transfer learning does not only significantly reduce the peaks in performance of the **forget** method but also achieves better adaptation than the **full transfer** method. Under all methods, the cumulative throughput is similar, but as observed in Fig. 8, station starvation, and consequently fairness are affected.

F. COMPLEXITY ANALYSIS

The SAU-Sampling MA-CMAB variants (Coop and Non-Coop) are built upon [29], where it has demonstrated an empirical reduction in running time and comparable complexity with other MAB techniques such as TS [57]. More specifically, the SAU metric τ_k^2 introduced in this work resembles TS according to `Proposition 2, page 5` in [29] with an empirical behavior similar to TS as well. This may not be an improvement when compared to the TS MAB from the regret theoretical analysis, but this approach can be adapted to any action-value function since it does not require access to the uncertainty of the expected reward as TS does.

Each agent in the SAU-Sampling MA-CMAB consists of a neural network that predicts the reward given the observed context. Such a predictor is composed of 2 hidden layers: the input layer, and the output layer. Consequently, three matrices are needed to represent the weight relationships of the four layers: W_{ab}, W_{ca}, W_{dc}, where a, b, c, d are the number of nodes of each layer. The propagation from layer a to b can be written in the following fashion:

$$S_{at} = W_{ab} * Z_{bt}, \ \texttt{\#Propagation from layer a to b} \tag{42}$$

$$Z_{at} = f(S_{at}). \ \texttt{\#Activation function} \tag{43}$$

The operation complexity in Eq. (42) corresponds to $\mathcal{O}(a * b * t)$, meanwhile Eq. (43) is $\mathcal{O}(a * t)$, which corresponds to a total complexity of $\mathcal{O}(a * b * t)$. Analogously, we can proceed with the matrices W_{ca} and W_{dc} and obtain a total time complexity of $\mathcal{O}(n * t * (ab + ca + dc))$, which can be further reduced to $\mathcal{O}(ab+ca+dc)$ since $n = t = 1$ in our MA-CMAB proposal. Thus, this represents a low time complexity with no implications for the performance of the algorithm.

VII. CONCLUSION

In this paper, we have proposed Machine Learning (ML)-based solutions to optimize Spatial Reuse (SR) in distributed Wi-Fi 802.11ax/802.11be scenarios. We have presented a solution to reduce the huge action space given the possible values of Transmission Power (TP) and Clear-Channel-Assessment (CCA) threshold values per Access Point (AP) and analyzed its impact on diverse well-known distributed Multi-Agent Multi-Armed Bandit (MA-MAB) implementations. In distributed scenarios, we have shown that the ϵ-greedy MA-MAB significantly improves the performance over typical configurations when the optimal actions are known. Moreover, the Contextual Multi-Agent Multi-Armed (MA-CMAB), named SAU-Sampling in the cooperative setting, contributes positively to an increase in throughput and fairness and a reduction of PLR when compared with non-cooperative approaches. Under dynamic scenarios, transfer learning benefits the SAU-Sampling algorithm to overcome service drops for at least 60% of the total users when utilizing the *forget* method. Additionally, we have shown that partial transfer learning is beneficial when compared to the *full transfer* method. In conclusion, the utilization of the cooperative version of the MA-CMAB to improve SR in Wi-Fi scenarios is preferable since it outperforms the presented ML-based solutions and prevents service drops in dynamic environments via transfer learning.

APPENDIX A
PROOFS

A. USEFUL PROPERTIES

Let us assume that action 1 refers to the optimal matching given an optimal team policy $\mu_{k^{**}}$, where $\mu_1 = \mu_{k^{**}}$ and k^{**} corresponds to the optimal bipartite matching. Similar to [29] we define $r_{k,1}, r_{k,2}, \cdots, r_{k,n_k}$ be the random variables referring to the rewards yielded by action k of successive n_k plays. Also, we fix n and avoid the usage of such subscript. The notation "k, j" means that the number of plays of action k is j. X_k is a random variable drawn from the normal distribution $\mathcal{N}(\hat{\mu}_{k,n_k}, \tau_k^2/n_k)$, where

$$\hat{\mu}_{k,n_k} = \frac{1}{n_k}\sum_{j=1}^{n_k} r_{k,j} \quad \text{and } \tau_k^2 = \frac{1}{n_k}\sum_{j=1}^{n_k}(r_{k,j} - \hat{\mu}_{a,j-1})^2.$$

Thus X_k becomes:

$$X_k = \hat{\mu}_{k,n_k} + \delta_{k,n_k}, \quad \text{where } \delta_{k,n_k} \sim \mathcal{N}(0, \tau_k^2/n_k). \tag{44}$$

Gaussian Tail bound: Based on the definition in the SAU-Sampling algorithm in [29], δ_{k,n_k} in probability. By the Gaussian tail bound [58, Definition 2.1], for $\alpha > 0$,

$$\Pr\left\{\delta_{k,n_k} \geq \alpha | \tau_k^2\right\} \leq \exp\left\{-\frac{\alpha^2 n_k}{2\tau_k^2}\right\} \leq \exp\left\{-n_k\alpha^2/2\right\}, \tag{45}$$

where the second inequality is from that $\tau_k^2 \leq 1$. Eq. (45) follows that

$$\Pr\left\{\delta_{k,n_k} \geq \alpha\right\} \leq \exp\left\{-n_k\alpha^2/2\right\}. \tag{46}$$

B. PROOF OF THEOREM 1

Proof: Let us define first the team regret based on the suboptimal selection of bipartite matching k*:

$$R(n) = \sum_{k*:\mu_{k*}<\mu_{k^{**}}} \Delta_{k^{**}}[T_{k*}(n)]$$

$$= \Delta_{k^{**}} \sum_{k*:\mu k* < \mu k^{**}} [T_{k*}(n)]$$
$$\leq \Delta_{k^{**}} \sum_{i=1}^{N} \sum_{j=1}^{K} [\tilde{T}_{ij}(n)] \tag{47}$$

where $\tilde{T}_{ij}(n) \in \mathbb{R}^{K \times N}$ corresponds to a counter that is incremented by 1 when a non-optimal matching is selected by the SAU-Sampling MA-CMAB algorithm. Additionally, the tuple (i, j) indicates the j^{th} arm selected by the i^{th} agent. $\Delta_{k^{**}}$ is defined as $\Delta_{k^{**}} = \mu_{n,k^{**}}(n) - \mu_{n,k}(n)$.

With $c_{1,n} = N\sqrt{\frac{(N+1)\log n}{n_1}}$ and $c_{k,n} = N\sqrt{\frac{(N+1)\log n}{n_k}}$, let $P_1 = \{\hat{\mu}_{1,n_1} > \mu_1 - \frac{1}{N}c_{1,n}\}$, and $P_k = \{\hat{\mu}_{k,n_k} < \mu_k + \frac{1}{N}c_{k,n}\}$ for $k = 2, \cdots, K$. where $\bar{P}_1$ and $\bar{P}_a$ be the complements of P_1 and P_k respectively. $\Pr\{\bar{P}_1\}$ and $\Pr\{\bar{P}_a\}$ are bounded from the Azuma-Hoeffding inequality [58, Corollary 2.1]:

$$\Pr\{\bar{P}_1\} = \Pr\left\{\hat{\mu}_{1,n_1} \leq \mu_1 - \frac{1}{N}c_{1,n}\right\}$$
$$\leq \exp(-2(M+1)\log n) = n^{-2(M+1)} \tag{48}$$
$$\Pr\{\bar{P}_a\} = \Pr\left\{\hat{\mu}_{k,n_k} \geq \mu_k + \frac{1}{N}c_{k,n}\right\} \tag{49}$$
$$\leq \exp(-2(M+1)\log n) = n^{-2(M+1)}.$$

Defining for $\eta \in \mathbb{R}$:

$$Q_{k,n_k}(\eta) = \Pr(X_k \geq \eta).$$

Following Eq. (47) and taken into account [36], the following lemma is used:

$$R(n) \leq \sum_{k*:\mu_{k*} < \mu_{k^{**}}} \Delta_{k**}(R_a + R_b), \tag{50}$$

where

$$R_a = \sum_{n_1=0}^{n-1} \left[\min\left\{\frac{1}{NQ_{1,n_1}(\eta)} - 1, n\right\}\right] \text{ and}$$
$$R_b = \sum_{n_k=0}^{n-1} \Pr[NQ_{k,n_k}\eta) > 1/n] + 1.$$

We set

$$\eta_{k**} = \mu_{k**} + \frac{\Delta_{k**}}{2} = \mu_1 - \frac{\Delta_{k**}}{2}. \tag{51}$$

First, we split R_a into two terms $R_a^{(1)}$ and $R_a^{(2)}$ and bound them by applying Theorem [36, Theorem 1]. Now in the first step we derive the upper bound on $R_a^{(1)}$. Denote $\bar{n}_{k**} = \frac{24N^2\log n}{\Delta_k^2}$. Noting $Q_{1,0}(\eta_k) = 1$ and $\lceil \bar{n}_{k**} \rceil$ is the smallest integer not less than $\bar{n}_{k**}$.

$$R_a = \sum_{n_1=1}^{n-1} \left[\min\left\{\frac{1}{NQ_{1,n_1}(\eta_{k**})} - 1, n\right\}\right]$$
$$= \sum_{n_1=1}^{\lceil \bar{n}_{k**} \rceil - 1} \left[\min\left\{\frac{1}{NQ_{1,n_1}(\eta_{k**})} - 1, n\right\}\right] \tag{52}$$
$$+ \sum_{n_1=\lceil \bar{n}_{k**} \rceil}^{n-1} \left[\min\left\{\frac{1}{NQ_{1,n_1}(\eta_{k**})} - 1, n\right\}\right]$$
$$=: R_a^{(1)} + R_a^{(2)}, \tag{53}$$

The law of total probability implies that, when $n_1 \geq \bar{n}_{k**}$,

$$\begin{aligned}
&Q_{1,n_1}(\eta_{k**}) \\
&= \Pr\{X_1 > \eta_{k**}\} = 1 - \Pr\{X_1 < \eta_{k**}\} \\
&= 1 - \Pr(P_1)\Pr\{X_1 < \eta_{k**}|P_1\} \\
&\quad - \Pr(\bar{P}_1)\Pr\{X_1 < \eta_{k**}|\bar{P}_1\} \\
&> 1 - \Pr\left\{\hat{\mu}_{1,n_1} + \delta_{1,n_1} < \mu_1 - \Delta_{k**}/2|P_1\right\} - \Pr(\bar{P}_1) \\
&\geq 1 - \Pr\left\{\delta_{1,n_1} \leq -\Delta_{k**}/2 + \frac{1}{N}c_{1,n}\right\} - \Pr(\bar{P}_1) \\
&> 1 - \Pr\left\{\delta_{1,n_1} \leq -\frac{\sqrt{2}}{N}c_{1,n}\right\} - \Pr(\bar{P}_1) \\
&\geq 1 - n^{-(N+1)} - n^{-2(N+1)},
\end{aligned} \tag{54}$$

where the 1st inequality is from the facts that $\Pr(P_1) < 1$ and $\Pr\{X_1 < \eta_{k**}|\bar{P}_1\} < 1$, the 2nd inequality is from the definition of P_1, the 3rd inequality is from $\Delta_{k**}/2 - \frac{1}{N}c_{1,n} \geq \frac{\sqrt{2}}{N}c_{1,n}$ as $n_1 \geq \bar{n}_{k**}$, and the last inequality is from Eqs. (46) and (48).

Eq. (54) implies that for $n \geq 3$ and $N \geq 1$,

$$\begin{aligned}
R_a^{(1)} &= \sum_{n_1=\lceil \bar{n}_{k**} \rceil}^{n-1} \left[\min\left\{\frac{1}{NQ_{1,n_1}(\eta_{k**})} - 1, n\right\}\right] \\
&= \sum_{n_1=\lceil \bar{n}_{k**} \rceil}^{n-1} \min\left\{\frac{1}{NQ_{1,n_1}(\eta_{k**})} - 1, n\right\} \\
&< \sum_{n_1=\lceil \bar{n}_{k**} \rceil}^{n-1} \frac{1}{N(1 - n^{-(N+1)} - n^{-2(N+1)})} - 1 \\
&< 2N \sum_{n_1=\lceil \bar{n}_{k**} \rceil}^{n-1} (n^{-(N+1)} + n^{-2(N+1)}) < \frac{3N}{N+1},
\end{aligned} \tag{55}$$

where the 1st inequality fulfills the condition $\frac{1}{Q_{1,n_1}(\eta_{k**})} - 1 < \frac{1}{1 - n^{-(N+1)} - n^{-2(N+1)}} - 1 < n$, the 2nd inequality is from that $\frac{1}{1 - n^{-(N+1)} - n^{-2(N+1)}} - 1 < 2(n^{-(N+1)} + n^{-2(N+1)})$ when $n \geq 3$ and $N \geq 1$.

It follows the calculation of the lower bound of R_a, $R_a^{(2)}$ in Eq. (52). A second lower bound can be derived from $Q_{1,n_1}(\eta_{k**})$ as follows: Let

$$P_1^L = \{\hat{\mu}_{1,n_1} \geq \mu_1\}.$$

Denote $\bar{P}_1^L$ be the complements of P_1^L, we have:

$$\Pr\{P_1^L\} = 1/2; \quad \Pr\{\bar{P}_1^L\} = 1/2 \tag{56}$$

Similarly as Eq. (54), we have:

$$\begin{aligned}
&Q_{1,n_1}(\eta_{k**}) \\
&= \Pr\{X_{1,n_1} > \eta_{k**}\} \\
&= 1 - \Pr(P_1^L)\Pr\{X_{1,n_1} < \eta_{k**}|P_1^L\}
\end{aligned}$$

$$
\begin{aligned}
&- \Pr(\bar{P}_1^L)\Pr\left\{X_{1,n_1} < \eta_{k**}|\bar{P}_1^L\right\}\\
&>1 - \frac{1}{2}\Pr\left\{\hat{\mu}_{1,n_1} + \frac{1}{N}c_{1,n}\delta_{1,n_1} < \mu_1 - \frac{\Delta_{k**}}{2}|P_1^L\right\}\\
&-\Pr(\bar{P}_1^L)\\
&\geq 1/2 - 1/2\Pr\left\{\frac{1}{N}c_{1,n}\delta_{1,n_1} \leq -\Delta_{k**}/2\right\}\\
&\geq \frac{1}{2}\left[1 - \exp\left(-\frac{n_1\Delta_{k**}^2}{8N^2\log n}\right)\right], \qquad (57)
\end{aligned}
$$

where the 1st inequality is from the facts that $\Pr\left\{X_{1,n_1} < \eta_{k**}|\bar{P}_1^L\right\} \leq 1$, and the 2nd inequality is from the definition of P_1^L and Eq. (56), and the last inequality is from Eq. (46).

We have that (1) $\log(1 - \frac{2}{n+1}) \geq \frac{-3}{n+1}$ when $n \geq 4$; (2) $\frac{-3}{n+1} \geq \frac{-3}{n}$; and (3) $-\frac{\Delta_{k**}^2}{8N^2\log n} \leq \frac{-3}{n+1}$ when $\frac{n}{\log n} \geq \frac{24N^2}{\min_{k**}\Delta_{k**}^2}$. The three inequalities imply that when $n \geq \max\{\frac{24N^2\log n}{\min_{k**}\Delta_{k**}^2}, 4\}$,

$$
1 - \exp\left(-\frac{\Delta_{k**}^2}{8N^2\log n}\right) \geq \frac{2}{n+1}.
$$

Thus, Eq. (57) follows:

$$
\begin{aligned}
R_a^{(1)} &= \sum_{n_1=1}^{\lceil\bar{n}_{k**}\rceil-1}\left[\min\left\{\frac{1}{NQ_{1,n_1}(\eta_{k**})} - 1, n\right\}\right]\\
&\leq \sum_{n_1=1}^{\lceil\bar{n}_{k**}\rceil-1}\left[\frac{2}{1 - \exp\left(-\frac{n_1\Delta_{k**}^2}{8N^2\log n}\right)} - 1\right]\\
&< \sum_{n_1=1}^{\lceil\bar{n}_{k**}\rceil-1}\left[4\exp\left(-\frac{n_1\Delta_{k**}^2}{8N^2\log n}\right) - 1\right]\\
&< 3\bar{n}_{k**}. \qquad (58)
\end{aligned}
$$

Therefore, inserting Eqs. (55) & (58) into Eq. (52),

$$
R_a^{(1)} \leq 3\bar{n}_k + 4. \qquad (59)
$$

In the next step, we derive the upper bound on $R_a^{(2)}$. When $n_{k**} \geq \bar{n}_{k**}$,

$$
\Delta_{k**}/2 - c_{k,n} \geq \sqrt{2}c_{k,n}. \qquad (60)
$$

From the definition of event P_k, the law of total probability implies:

$$
\begin{aligned}
&\Pr\left\{\frac{1}{N}Q_{k,n_k}(\eta_k) > 1/n\right.\\
&= \Pr(P_k)\Pr\left\{\frac{1}{N}Q_{k,n_k}\eta_k\right) > 1/n\Big|\ P_k\Big\}\\
&\quad + \Pr\left(\bar{P}_a\right)\Pr\left\{\frac{1}{N}Q_{k,n_k}(\eta_k) > 1/n\Big|\ \bar{P}_a\right\}\\
&\leq \Pr\left\{\frac{1}{N}Q_{k,n_k}(\eta_k) > 1/n\Big|\ P_k\right\} + \Pr\left(\bar{P}_a\right). \qquad (61)
\end{aligned}
$$

When $n_k \geq \bar{n}_k$, given event P_k,

$$
\begin{aligned}
Q_{k,n_k}(\eta_k) =& \Pr\{X_k > \eta_k|P_k\} = \qquad (62)\\
&\Pr\left\{\hat{\mu}_a + \delta_{k,n_k} > \Delta_k/2 + \mu_k|P_k\right\}\\
\leq& \Pr\left\{\delta_{k,n_k} \geq \Delta_k/2 - \frac{1}{N}c_{k,n}\right\}\\
\leq& \Pr\left\{\delta_{k,n_k} \geq \frac{\sqrt{2}}{N}c_{k,n}\right\}\\
&\leq \exp\{-(N+1)\log n\} = n^{-(N+1)}, \qquad (63)
\end{aligned}
$$

where the 1st inequality is from the definition of event P_k, the 2nd inequality is from Eq. (60), the 3rd inequality is from Eq. (46).

Eq. (62) follow that when $n_k \geq \bar{n}_k$,

$$
\Pr\left\{\frac{1}{N}Q_{k,n_k}(\eta_k) > 1/n|P_k\right\} = 0. \qquad (64)
$$

Inserting Eq. (64) into Eq. (61),

$$
\Pr[\frac{1}{N}Q_{k,n_k}(\eta_k) > 1/n] \leq \quad \Pr(\bar{P}_a) \leq n^{-(N+1)}, \qquad (65)
$$

where the last step is from Eq. (49). We have:

$$
\begin{aligned}
R_b &= \sum_{n_k=0}^{n-1}\Pr[NQ_{k,n_k}(\eta_k) > 1/n] + 1\\
&\leq \sum_{n_k=0}^{\lceil\bar{n}_k\rceil}\Pr[NQ_{k,n_k}(\eta_k) > 1/n] + 1\\
&+ \sum_{n_k=\lceil\bar{n}_k\rceil}^{n-1}\Pr[NQ_{k,n_k}(\eta_k) > 1/n] + 1\\
&\leq N\bar{n}_k + \sum_{n_k=\lceil\bar{n}_k\rceil}^{n-1}\Pr[NQ_{k,n_k}(\eta_k) > 1/n] + 1\\
&< N\bar{n}_k + (N+1)\\
&< N(\bar{n}_k + 1) + 1, \qquad (66)
\end{aligned}
$$

Finally, we substitute R_a and R_b terms in Eq. (50) having:

$$
R(T) \leq N^2K\Delta_{k**}\left(\frac{96\log n}{\Delta_{k**}^2} + \frac{1}{1+N}\right) \qquad (67)
$$

■

REFERENCES

[1] *Cisco Annual Internet Report (2018–2023)*, Cisco Syst. Inc., San Jose, CA, USA, 2020.

[2] Wi-Fi Alliance. (2023). *Wi-Fi by the Numbers: Technology Momentum in 2023*. [Online]. Available: https://www.wi-fi.org/beacon/the-beacon/wi-fi-by-the-numbers-technology-momentum-in-2023

[3] IEEE 802.11. *Official IEEE 802.11 Working Group Project Timelines*. Accessed: Dec. 5, 2023. [Online]. Available: https://www.ieee802.org/11/Reports/802.11_Timelines.htm

[4] F. Ye, S. Yi, and B. Sikdar, "Improving spatial reuse of IEEE 802.11 based ad hoc networks," in *Proc. IEEE Global Telecommun. Conf.*, vol. 2, Dec. 2003, pp. 1013–1017.

[5] F. Wilhelmi, S. Barrachina-Muñoz, C. Cano, I. Selinis, and B. Bellalta, "Spatial reuse in IEEE 802.11ax WLANs," *Comput. Commun.*, vol. 170, pp. 65–83, Mar. 2021.

[6] C. Thorpe and L. Murphy, "A survey of adaptive carrier sensing mechanisms for IEEE 802.11 wireless networks," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 3, pp. 1266–1293, 3rd Quart., 2014.

[7] T. Huehn and C. Sengul, "Practical power and rate control for WiFi," in *Proc. 21st Int. Conf. Comput. Commun. Netw. (ICCCN)*, Jul. 2012, pp. 1–7.

[8] D. Nunez, F. Wilhelmi, S. Avallone, M. Smith, and B. Bellalta, "TXOP sharing with coordinated spatial reuse in multi-AP cooperative IEEE 802.11be WLANs," in *Proc. IEEE 19th Annu. Consum. Commun. Netw. Conf. (CCNC)*, Jan. 2022, pp. 864–870.

[9] C. Khosla and B. S. Saini, "Enhancing performance of deep learning models with different data augmentation techniques: A survey," in *Proc. Int. Conf. Intell. Eng. Manage. (ICIEM)*, Jun. 2020, pp. 79–85.

[10] P. E. Iturria-Rivera, M. Chenier, B. Herscovici, B. Kantarci, and M. Erol-Kantarci, "Channel selection for Wi-Fi 7 multi-link operation via optimistic-weighted VDN and parallel transfer reinforcement learning," in *Proc. IEEE 34th Annu. Int. Symp. Pers., Indoor Mobile Radio Commun. (PIMRC)*, Sep. 2023, pp. 1–6.

[11] S. Szott et al., "Wi-Fi meets ML: A survey on improving IEEE 802.11 performance with machine learning," *IEEE Commun. Surveys Tuts.*, vol. 24, no. 3, pp. 1843–1893, 3rd Quart., 2022.

[12] P. E. Iturria-Rivera, M. Chenier, B. Herscovici, B. Kantarci, and M. Erol-Kantarci, "RL meets multi-link operation in IEEE 802.11be: Multi-headed recurrent soft-actor critic-based traffic allocation," in *Proc. IEEE Int. Conf. Commun.*, May 2023, pp. 4001–4006.

[13] M. Elsayed and M. Erol-Kantarci, "AI-enabled future wireless networks: Challenges, opportunities, and open issues," *IEEE Veh. Technol. Mag.*, vol. 14, no. 3, pp. 70–77, Sep. 2019.

[14] P. E. Iturria-Rivera, H. Zhang, H. Zhou, S. Mollahasani, and M. Erol-Kantarci, "Multi-agent team learning in virtualized open radio access networks (O-RAN)," *Sensors*, vol. 22, no. 14, p. 5375, Jul. 2022.

[15] TIP. (2022). *OpenWiFi Release 2.4 GA*. Accessed: Feb. 2, 2023. [Online]. Available: https://openwifi.tip.build/

[16] F. Wilhelmi, C. Cano, G. Neu, B. Bellalta, A. Jonsson, and S. Barrachina-Muñoz, "Collaborative spatial reuse in wireless networks via selfish multi-armed bandits," *Ad Hoc Netw.*, vol. 88, pp. 129–141, May 2019.

[17] A. Bardou, T. Begin, and A. Busson, "Improving the spatial reuse in IEEE 802.11ax WLANs," in *Proc. 24th Int. ACM Conf. Model., Anal. Simul. Wireless Mobile Syst.*, Nov. 2021, pp. 135–144.

[18] A. Bardou and T. Begin, "INSPIRE: Distributed Bayesian optimization for improving spatial reuse in dense WLANs," 2022, *arXiv:2204.10184*.

[19] H. Lee, H.-S. Kim, and S. Bahk, "LSR: Link-aware spatial reuse in IEEE 802.11ax WLANs," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Mar. 2021, pp. 1–6.

[20] H. Kim and J. So, "Improving spatial reuse of wireless LAN uplink using BSS color and proximity information," *Appl. Sci.*, vol. 11, no. 22, p. 11074, Nov. 2021.

[21] F. Wilhelmi et al., "Federated spatial reuse optimization in next-generation decentralized IEEE 802.11 WLANs," 2022, *arXiv:2203.10472*.

[22] D. Bouneffouf, I. Rish, and C. Aggarwal, "Survey on applications of multi-armed and contextual bandits," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jul. 2020, pp. 1–8.

[23] T. Lattimore and C. Szepesvári, *Bandit Algorithms*. Cambridge, U.K.: Cambridge Univ. Press, 2020.

[24] A. Slivkins, "Introduction to multi-armed bandits," *Found. Trends Mach. Learn.*, vol. 12, nos. 1–2, pp. 1–286, 2019.

[25] L. Zhou, "A survey on contextual multi-armed bandits," 2015, *arXiv:1508.03326*.

[26] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. Cambridge, MA, USA: MIT Press, 2018.

[27] R. Agrawal, "Sample mean based index policies by $O(\log n)$ regret for the multi-armed bandit problem," *Adv. Appl. Probab.*, vol. 27, no. 4, pp. 1054–1078, Dec. 1995.

[28] D. J. Russo, B. Van Roy, A. Kazerouni, I. Osband, and Z. Wen, "A tutorial on Thompson sampling," *Found. Trends Mach. Learn.*, vol. 11, no. 1, pp. 1–96, 2018.

[29] R. Zhu and M. Rigotti, "Deep bandits show-off: Simple and efficient exploration with deep networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 17592–17603.

[30] S. Hossain, E. Micha, and N. Shah, "Fair algorithms for multi-agent multi-armed bandits," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 24005–24017.

[31] P. C. Landgren, "Distributed multi-agent multi-armed bandits," Ph.D. dissertation, Princeton Univ. Press, Princeton, NJ, USA, 2019.

[32] G. Lugosi and A. Mehrabian, "Multiplayer bandits without observing collision information," *Math. Oper. Res.*, vol. 47, no. 2, pp. 1247–1265, May 2022.

[33] W. Huang, R. Combes, and C. Trinh, "Towards optimal algorithms for multi-player bandits without collision sensing information," in *Proc. 35th Conf. Learn. Theory*, in Proceedings of Machine Learning Research, vol. 178, P.-L. Loh and M. Raginsky, Eds., 2022, pp. 1990–2012. [Online]. Available: https://proceedings.mlr.press/v178/huang22a.html

[34] P.-A. WANG, A. Proutiere, K. Ariu, Y. Jedra, and A. Russo, "Optimal algorithms for multiplayer multi-armed bandits," in *Proc. 23rd Int. Conf. Artif. Intell. Statist.*, in Proceedings of Machine Learning Research, vol. 108, S. Chiappa and R. Calandra, Eds., 2020, pp. 4120–4129. [Online]. Available: https://proceedings.mlr.press/v108/wang20m.html

[35] Y. Gai, B. Krishnamachari, and R. Jain, "Learning multiuser channel allocations in cognitive radio networks: A combinatorial multi-armed bandit formulation," in *Proc. IEEE Symp. New Frontiers Dyn. Spectr. (DySPAN)*, Apr. 2010, pp. 1–9.

[36] B. Kveton, C. Szepesvári, S. Vaswani, Z. Wen, M. Ghavamzadeh, and T. Lattimore, "Garbage in, reward out: Bootstrapping exploration in multi-armed bandits," in *Proc. 36th Int. Conf. Mach. Learn.*, 2019, pp. 3601–3610.

[37] S. Agrawal and N. Goyal, "Near-optimal regret bounds for Thompson sampling," *J. ACM*, vol. 64, no. 5, pp. 1–24, Oct. 2017.

[38] P. Landgren, V. Srivastava, and N. E. Leonard, "Distributed cooperative decision making in multi-agent multi-armed bandits," *Automatica*, vol. 125, Mar. 2021, Art. no. 109445.

[39] S. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, pp. 1345–1359, Nov. 2010.

[40] T. Scott, K. Ridgeway, and M. C. Mozer, "Adapted deep embeddings: A synthesis of methods for k-shot inductive transfer learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, 2018, pp. 76–85.

[41] F. Zhuang et al., "A comprehensive survey on transfer learning," *Proc. IEEE*, vol. 109, no. 1, pp. 43–76, Jan. 2021.

[42] L. Vu, Q. U. Nguyen, D. N. Nguyen, D. T. Hoang, and E. Dutkiewicz, "Deep transfer learning for IoT attack detection," *IEEE Access*, vol. 8, pp. 107335–107344, 2020.

[43] M. Elsayed, M. Erol-Kantarci, and H. Yanikomeroglu, "Transfer reinforcement learning for 5G new radio mmWave networks," *IEEE Trans. Wireless Commun.*, vol. 20, no. 5, pp. 2838–2849, May 2021.

[44] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, "A survey on deep transfer learning," in *Proc. Int. Conf. Artif. Neural Netw.*, 2018, pp. 270–279.

[45] Z. Zhu, K. Lin, A. K. Jain, and J. Zhou, "Transfer learning in deep reinforcement learning: A survey," 2020, *arXiv:2009.07888*.

[46] Y. Xiao, "IEEE 802.11e: QoS provisioning at the MAC layer," *IEEE Wireless Commun.*, vol. 11, no. 3, pp. 72–79, Jun. 2004.

[47] M. Derakhshani, X. Wang, D. Tweed, T. Le-Ngoc, and A. Leon-Garcia, "AP-STA association control for throughput maximization in virtualized WiFi networks," *IEEE Access*, vol. 6, pp. 45034–45050, 2018.

[48] G. Holland, N. Vaidya, and P. Bahl, "A rate-adaptive MAC protocol for multi-hop wireless networks," in *Proc. 7th Annu. Int. Conf. Mobile Comput. Netw.*, Jul. 2001, pp. 236–251.

[49] G. F. Riley and T. R. Henderson, *The NS-3 Network Simulator*. Berlin, Germany: Springer, 2010, pp. 15–34, doi: 10.1007/978-3-642-12331-3_2.

[50] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 3, pp. 535–547, Mar. 2000.

[51] T.-S. Kim, H. Lim, and J. C. Hou, "Improving spatial reuse through tuning transmit power, carrier sense threshold, and data rate in multihop wireless networks," in *Proc. 12th Annu. Int. Conf. Mobile Comput. Netw.*, Sep. 2006, pp. 366–377.

[52] S. Padakandla, "A survey of reinforcement learning algorithms for dynamically varying environments," *ACM Comput. Surv.*, vol. 54, no. 6, pp. 1–25, Jul. 2022.

[53] A. Blázquez-García, A. Conde, U. Mori, and J. A. Lozano, "A review on outlier/anomaly detection in time series data," *ACM Comput. Surv.*, vol. 54, no. 3, pp. 1–33, 2021.

[54] P. Gawłowicz and A. Zubow, "NS-3 meets OpenAI gym: The playground for machine learning in networking research," in *Proc. 22nd Int. ACM Conf. Modeling, Anal. Simulation Wireless Mobile Syst.*, Nov. 2019, pp. 113–120.

[55] F. Wilhelmi, S. Barrachina-Mu noz, B. Bellalta, C. Cano, A. Jonsson, and G. Neu, "Potential and pitfalls of multi-armed bandits for decentralized spatial reuse in WLANs," *J. Netw. Comput. Appl.*, vol. 127, pp. 26–42, Feb. 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1084804518303655
[56] M. Bayati, N. Hamidi, R. Johari, and K. Khosravi, "Unreasonable effectiveness of Greedy algorithms in multi-armed bandit with many arms," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 1713–1723.
[57] C. Riquelme, G. Tucker, and J. Snoek, "Deep Bayesian bandits showdown: An empirical comparison of Bayesian deep networks for Thompson sampling," in *Proc. 6th Int. Conf. Learn. Represent.*, 2018, pp. 1–6.
[58] M. J. Wainwright, *High-Dimensional Statistics: A Non-Asymptotic Viewpoint*. Cambridge, U.K.: Cambridge Univ. Press, 2019.

PEDRO ENRIQUE ITURRIA-RIVERA (Student Member, IEEE) received the B.Eng. degree from Universidad Central "Marta Abreu" de Las Villas, Santa Clara, in 2015, and the M.Sc. degree from Instituto Politecnico Nacional in 2019. He is currently pursuing the Ph.D. degree with the School of Electrical Engineering and Computer Science, University of Ottawa. He is a member of the Networked Systems and Communications Research (NETCORE) Laboratory. His research interests include distributed multi-agents for AI-enabled wireless networks, 5G and 6G wireless communications, Wi-Fi, data privacy, data science, and color science. He was a recipient of the two Best Paper Awards from the 2023 IEEE ICC Conference.

MARCEL CHENIER (Member, IEEE) received the B.Eng. degree from the University of Ottawa. He is currently the CTO and the Co-Founder of NetExperience Inc., Canada, where he brings three decades of technology leadership in the field of public and enterprise wireline, wireless, and cloud networking. Prior to NetExperience Inc., he took the role of the CTO and the VP of Engineering at KodaCloud, building a cloud network control solution for MSP WLAN networks. Before KodaCloud, he led product architecture for the WLAN portfolio and small-cell plug-and-play solutions at Ericsson. Prior to Ericsson, he was the Vice President of Engineering at BelAir Networks Inc., a WLAN carrier-class small-cell equipment provided that was acquired by Ericsson. Prior to BelAir Networks Inc., he was the Head of engineering at Zhone Technologies, Canada, from the early company stage to IPO responsible for voice and video packet products. As the VP of Research and Development for Premisys, a smart DLC equipment provider, he was a part of the leadership team that led the acquisition by Zhone Technologies. He started his career with Nortel Networks working on silicon, hardware, and software research and development and as a Research and Development Executive for the Signaling System Product Line. He is a co-inventor on several patents in the area of wireless networking architecture.

BERNARD HERSCOVICI received the M.B.A. degree from the University of Ottawa and the M.A.Sc. degree in wireless technology from the University of Toronto. He is currently the Co-Founder and the CEO of NetExperience Inc., a company that provides the industry's first and most widely used software platform to control, manage, and automate disaggregated WiFi networks compatible OpenWiFi architecture. He has started and led as the CEO of several companies in the Wi-Fi industry, including BelAir Networks Inc. and KodaCloud, and held executive positions in various companies, such as Ericsson, Breezecom, and Newbridge Networks. He has authored several patents in the wireless space.

BURAK KANTARCI (Senior Member, IEEE) received the Ph.D. degree in computer engineering. He is currently a Full Professor and the Founding Director of the Smart Connected Vehicles Innovation Centre (SCVIC) and the Next Generation Communications and Computing Networks (NEXTCON) Research Laboratory, University of Ottawa. He is the author/coauthor of more than 250 publications in established journals and conferences, and 15 book chapters. He holds an Exemplary Editor Award from the IEEE Communications Surveys and Tutorials in 2021, and multiple Best Paper Awards from various conferences, most recently from the IEEE Globecom2021, the Wireless World Research Forum 2022, IEEE ICC2023, and IEEE VCC2023. He was also a recipient of the Minister's Award of Excellence from Ontario Ministry of Colleges and Universities in 2021 and the Winner of the 2023 Technical Achievement Award from the IEEE ComSoc Communications Software Technical Committee. He was a Distinguished Speaker of the Association of Computing Machinery (ACM) from 2019 to 2021. He is currently a Distinguished Lecturer of the IEEE Communications Society and the IEEE Systems Council. He has been a keynote/invited speaker or panelist in 40 events. From 2019 to 2020, he was the Chair of the Communications Systems Integration and Modeling Technical Committee, Institute of Electrical and Electronics Engineers (IEEE). He has been the general chair, the program chair, or the track chair of more than 30 international conferences. He is an Editor of the IEEE Communications Surveys and Tutorials and IEEE Internet of Things Journal and an Associate Editor of IEEE Networking Letters and *Vehicular Communications* (Elsevier). He has been listed among the top-cited scientists in telecommunications and networking based on the data reported by Stanford University since 2020.

MELIKE EROL-KANTARCI (Senior Member, IEEE) is currently the Chief Cloud RAN AI/ML Data Scientist at Ericsson and the Canada Research Chair in AI-enabled Next-Generation Wireless Networks and a Full Professor with the School of Electrical Engineering and Computer Science, University of Ottawa. She is also the Founding Director of the Networked Systems and Communications Research (NETCORE) Laboratory. She is the co-editor of three books on smart grids, smart cities, and intelligent transportation. She has over 200 peer-reviewed publications. Her main research interests include AI-enabled wireless networks, 5G and 6G wireless communications, smart grids, and the Internet of Things. She is a Senior Member of ACM. She received numerous awards and recognitions. She has acted as the general chair and the technical program chair for many international conferences and workshops. She is on the editorial board of IEEE Transactions on Communications, IEEE Transactions on Cognitive Communications and Networking, and IEEE Networking Letters. She has delivered more than 70 keynotes, plenary talks, and tutorials around the globe. She was an IEEE ComSoc Distinguished Lecturer from 2021 to 2023.