International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 08 Issue: 06 | June - 2024 SJIF Rating: 8.448 ISSN: 2582-3930

Machine Learning Based Beam Selection for Maximizing Wireless Netwok Capacity

Sunad Kumar A N
Computer Science and Engineering,
BGS Institute of Technology,
BG Nagar, Karnataka

Manoj C D
Computer Science and Engineering,
BGS Institute of Technology,
BG Nagar, Karnataka

***Abstract* — In today's and future wireless communications, especially in 5G and 6G networks, machine learning (ML) methods are crucial. Potentially, these techniques bring many benefits such as increased data throughput, improved security, reduced latency, and, on the whole, enhanced network efficiency. Furthermore, to facilitate the processing of large amounts of data in real-time situations, machine learning is used for various functions in wireless networks. This article aims to explore the significance and application of machine learning, with a particular focus on classic reinforcement learning, in the context of predicting optimal beam configurations within wireless communications scenarios. Our goal is to minimize interference between transmitters by finding the optimal beamforming angles. For this, ray tracing techniques are deployed. We see this research as a step forward towards integrating digital twin (DT) technology in network management and control. In this article, different machine learning methods are used and their performance is compared. Firstly, the most effective angles for beamforming, maximizing channel capacity are identified. Then, by using these methods and after verifying their accuracy, the optimal antenna angles in scenarios with an increased number of transmitters and receivers is found and evaluated**

I. INTRODUCTION

For reaching the ambitious goals of next generation networks, more optimal utilization of the limited resource bandwidth by spatial re-use and multiplexing is needed. To meet the evolving demands of future networks, 6G is slated to embrace an extensive integration of artificial intelligence (AI) and machine learning (ML) techniques. This deployment of AI and ML is aimed at achieving heightened automation and superior operational reliability, currently unattainable by the incumbent 5G technology. Many research endeavors are already underway to lay the foundations for 6G wireless communication networks.

Machine learning, which mimics human cognitive processes, is critical for improving wireless communication in various ways. It enhances computer vision, image processing, parallel processing, distributed processing, analytics, and prediction capabilities. In machine learning, models are extensively trained using datasets to ensure they perform well across a variety of examples. This is particularly crucial for tasks such as image classification and sentiment analysis . The use of machine learning in controlling communication systems has seen significant growth recently. Key references cover a range of applications, including source and channel coding waveform design signal detection resource allocation and channel estimation among others. This expanding field, highlights the effectiveness of a machine-learning approach in automatically finding optimal solutions based on training data.

communication using intelligent reflective surfaces (IRS) in innovative solution, tackling both AI's data demands and the

aleadingmethod in machine learning.Exploringthesynergy between machine learning and wireless communication system design, another study provides three practical examples In the first example, machine learning optimizes reflection coefficients, enhancing performance by bypassing channel estimation. The second example explores distributed source coding in massive Multiple Input Multiple Output (MIMO), emphasizing the feasibility of short block-length code design for significant performance gains and the third example illustrates machine learning's role in navigating the optimization landscape for millimeter-wave initial alignment in a complex sequential learning problem. Furthermore, in the context of the B5G network, ML-enabled scheduling is highlighted for its crucial role in reducing queuing latency and ensuring reliable services. In a reinforcementlearning-based framework is presented for wireless channel access mechanisms in IEEE 802.11 standards, particularly in the context of Massive Internet of Things (mIoT). Reference classifies application scenarios, including strengthened eMBB/mMTC/uRLLC and novel scenarios like space–air–ground–sea integrated networks and AI-enabled networks. These scenarios illustrate the integration of AI and big data techniques with key technologies and applications, improving their comprehensive utility. A different perspective is explained by a study that outlines ten key roles for machine learning in joint sensing and communication (JSC), sensing-aided communication, and communicationaided sensing. Additionally, proposes an RL-based model for RADAR operation prediction, facilitating the identification of unused communication channels. Reference recommends the development of ad hoc AI/ML models to enhance their practical usability. Reference clarifies the potential of machine learning in the link-to-link aspects of communication systems, with a

focus on neural-networkbased reinforcement learning algorithms and on-orbit testing.

In addition, beamforming as a technique recognized for its capacity to enhance wireless communication performance, holds promise for increasing 6G internet capabilities. In

Qi et al. present a novel 6G IoT network with UAVs and Intelligent Reflective Surfaces (IRS) for efficient data transmission by backscattering communication (BackCom). IRS, using beamforming, enhances signal energy, improving BackCom system performance and range. In Ihsan and et al. aim to improve energy efficiency in 6G wireless a non-orthogonal multiple access beamforming (NOMA-BF) system. It optimizes beamforming, power allocation, and performance while keeping complexity low. In Jiang and et al. present a novel initial beamforming approach that uses complementary beams to achieve an equal gain in all directions, ensuring comprehensive coverage. Numerical results confirm its potential for significantly enhancing 6G internet performance.

This article investigates the practical application of machine learning, especially reinforcement learning and, for comparison, a Monte Carlo approach. Ray tracing simulations are used, for detecting the best angles for beamforming in transmitters and receivers, thereby aiming to achieve the maximum total channel capacity. To our knowledge, no article has employed classic machine learning methods to determine optimal angles for antennas on transmitters and receivers using ray tracing in a indoor room environment and subsequently compared these approaches based on channel capacity.

In Section II, an overview of machine learning, with a particular focus on reinforcement learning, is provided. Furthermore, it is also briefly explained how the Monte Carlo method is applied. Section III offers extensive information on the experimental setup, clarifying the methodology employed for antenna pattern generation and detailing the classic reinforcement learning methods and Monte Carlo method utilized. The ensuing section, Section IV, presents the results of our simulations and of our analytical efforts. In Section V, we provide conclusions from our findings. Additionally, we outline potential areas for future research in this field.

II. OVERVIEW OF MACHINE LEARNING WIRELESS NETWORK OPERATION

Machine Learning (ML) is highly useful for solving complex problems with intricate patterns, like those found in tasks such as network density and traffic load estimation. ML techniques are typically classified into supervised learning (SL), unsupervised learning (uSL), and Reinforcement Learning (RL) . In supervised learning, the agent learns from labeled data with clear input-output pairs. Unsupervised learning, on the other hand, does not require labeled data and relies on the inherent structure of the data. Reinforcement Learning (RL), the main approach in this article, involves a dynamic balance between exploration and exploitation in an environment, using both labeled and unlabeled input data. In RL, the goal is to maximize expected rewards by learning the best policies and actions that connect current states to unknown future states in the environment. This learning process involves states, actions, rewards, and state-transition probabilities, which together define the new environment. RL-aware frameworks are well-suited for next-generation wireless communications because of their adaptability and effectiveness. Notably, RL algorithms have lower computational complexity compared to other supervised and unsupervised techniques, as they learn from real-time experiences rather than relying on preexisting datasets. This approach involves a balance between exploration (randomly selecting actions with a probability 'ε') and exploitation (choosing actions with the highest value function with a probability of '1 - ε'). This exploration-exploitation trade-off is pivotal in determining the optimal solution . Fig. 1 offers an overviewofvariousMachineLearning(ML)algorithms,with aspecificfocusonReinforcementLearning(RL)-algorithms that we will examine more in the future.

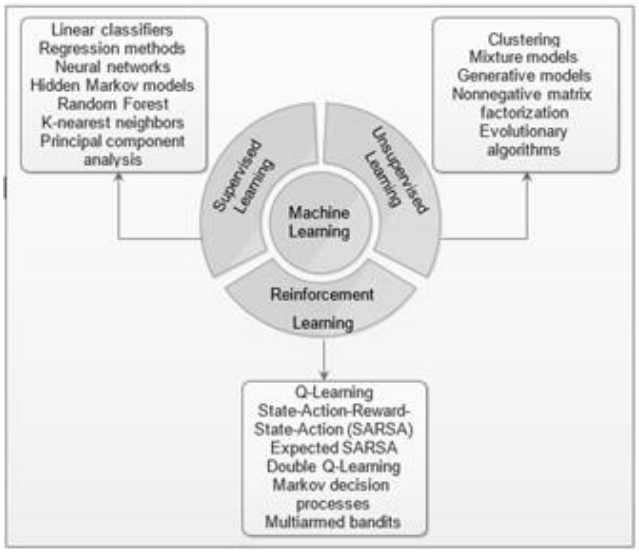

FIGURE 1. An overview of machine learning algorithms.

Reinforcement Learning (RL) is a crucial part of machine learning that focuses on creating smart agents capable of making a series of decisions. These agents aim to maximize a total reward, representing rational and goal-oriented behavior .

Initially, we explored supervised learning methods in our research. However, we faced challenges due to nonlinear relationships and complex interactions among antenna angles and maximum total channel capacity. The complexity of the data made regression techniques impractical. Consequently, we changed our focus to classic reinforcement learning (RL) and Monte Carlo as a stochastic method. RL, known for effectively handling nonlinearity and decoding complex relationships in data, became a more suitable and effective approach for covering our specific problem. In addition, in a scenario involving a massive number of combinations, evaluating all possible configurations exhaustively may not be feasible.

Monte Carlo methods allow us to explore a representative subset of the solution space by randomly sampling configurations . This approach provides a reasonable approximation of the optimal solution while significantly reducing the computational burden associated with an exhaustive search . The motivation behind comparing reinforcement learning (RL) with Monte Carlo methods stems from their distinct approaches to problem-solving and decision-making. Each methodology possesses distinct strengths and weaknesses, prompting a comparative analysis to evaluate which approach proves more effective within a specific context. In the following discussion, we will briefly introduce the Monte Carlo method and some potentially suitable RL methodologies.

A. MONTE CARLO

Monte Carlo methods are a general class of computational algorithms that rely on random sampling to obtain numerical results. They are especially suitable for tasks structured in episodes, where the agent interacts with the environment and gathers returns. In such episodic tasks, the agent interacts with the environment over a sequence of episodes, with each episode consisting of a series of steps or time-steps. In mathematical terms, they approximate the action-value function Q (s, a) by averaging returns obtained from multiple episodes:

$$Q(s,a)=1/N\Sigma^{N}{}_{i=1}G_i(s,a)$$

The averaging process involves sampling and averaging the returns achieved by taking a specific action in a particular state across different episodes. This allows for a more robust estimation of the action's value in that state, as it considers the variability in returns across different episodes .

B. Q-LEARNING

Q-learning is a fundamental reinforcement learningalgorithm that trains an agent to make sequential decisions to maximize cumulative rewards in an environment. It is model-free, requiring no prior knowledge of the environment's dynamics, and is off-policy, allowing it to learn from past experiences . The central concept in Q-learning is the Q-value, denoted as Q (s, a), which signifies the expected cumulative reward when taking action 'a' in state 's' and subsequently following the optimal policy. Q-values are iteratively updated using the Bellman equation :

$$Q(s,a) \leftarrow Q(s,a) + \alpha\,[R + \gamma.\max[Q\ s',a' - Q(s,a)]]$$

where:

- $Q(s,a)$ is the Q-value for state-action pair (s,a).
- α is the learning rate, determining the step size in updates.
- R is the immediate reward received after taking action 'a' instate 's'.
- γ is the discount factor that balances immediate and future rewards.
- $Q\ s',a'$ represents the Q-value for the next state-action pair after the action 'a' is taken in state 's'.

Through a process of exploration and exploitation, Q-learning guides the agent to refine its Q-values over time, enabling it to make better decisions. Q-learning is particularly effective in solving problems where agents must learn to navigate uncertain environments to maximize their cumulative rewards.

C. DOUBLE Q-LEARNING

Double Q-learning is a sophisticated technique within the domain of reinforcement learning. It was developed to adjust a common challenge known as overestimation bias, which can lead to inaccuracies in estimating the values of actions. This method extends the conventional Q-learning framework by introducing two distinct Q-value functions and employs an alternating approach during the learning process. In mathematical terms, Double Q-learning involves the following steps:

1) Initialization of two separate sets of Q-values, denoted as $Q_1(s,a)$ and $Q_2(s,a)$ where s signifies the state, and a represents the action.

2) During each learning iteration, a decision is made between these two sets of Q-values with a 50% probability:

- *if rand ()* $< 0.5, Q_1$ is updated using the Bellman equation:

$$Q_1(s,a) \leftarrow Q_1(s,a) + \alpha[R + \gamma.Q_2(s',argmax_a Q_1\ s',a - Q_1(s,a)]$$

- *if rand ()* $\geq 0.5, Q_2$ is updated similarly:

$$Q_2(s,a) \leftarrow Q_2(s,a) + \alpha.[R + \gamma.Q_1(s',argmax_a Q_2(s',a) - Q_2(s,a)]]$$

3) The agent consistently alternates between Q_1 and Q_2 during learning iterations, which effectively decreases the problem of overestimation bias.

Double Q-learning finds particular relevance in scenarios where accurately estimating action values is critical, as it helps prevent potential inaccuracies that can adversely affect decision-making in dynamic environments.

D.SARSA(STATE-ACTION-REWARD-STATE-ACTION)

SARSA, or State-Action-Reward-State-Action, is a reinforcement learning technique where an agent learns to make decisions within an environment. It associates states (environmental conditions), actions (choices), and rewards (immediate outcomes). SARSA aims to optimize cumulative rewards over time . In SARSA, the agent estimates Q-values for state-action pairs ($Q(s,a)$). It updates these estimates using a formula:

$$Q(S,A) \leftarrow Q(S,A) + \alpha.[R + \gamma.Q\ S',A' - Q(S,A)]$$

(3)

where:

- α (alpha) is the learning rate, controlling update size.
- R is the immediate reward.
- γ (gamma) is the discount factor, valuing future rewards.
- (S', A') represents the next state-action pair based on the agent's policy.

In summation, SARSA is a basic algorithm for reinforcement learning, emphasizing on-policy learning, where decisions are made in line with the current policy. Through the estimation of Q-values, SARSA prepares agents to make progressively informed decisions within a dynamic and uncertain environment, ultimately optimizing their long-term rewards.

E. EXPECTED SARSA

Expected SARSA is a reinforcement learning algorithm that estimates the expected value of action-values under the current policy. It's less sensitive to noise in the environment compared to SARSA.

The algorithm updates its estimates using the formula:

$$Q(s,a) - Q(s,a) + \alpha[R + \Upsilon\Sigma_{a'}\pi\,(a'/s') \text{ -Q(s,a)} \qquad (4)$$

In this formula:

- $Q(s,a)$ represents the estimated action-value.
- α is the learning rate.
- R is the immediate reward.
- γ is the discount factor.
- $\pi\, a' \,|s'$ is the probability of taking action a' in the next state s'.

Expected SARSA calculates the expected value by combining Q-values for all possible next-state actions, weighted by their probabilities according to the current policy 'π'. This reduces sensitivity to noisy rewards, making it a robust choice for uncertain environments.

Moreover, it's important to note that the computational complexity of the four learning methods—Q-Learning, Double Q-Learning, SARSA, and Expected SARSA—is O(S∗A), where S represents the number of states in the problem and A represents the number of possible actions in each state. Additionally, for the Monte Carlo Method, the complexity is O(S∗A∗T), with T denoting the episode length.

A more detailed discussion on these ML categories and techniques is out of the scope of this article, and we suggest readers refer to for further details.

1

III. EXPERIMENTAL SETUP

In this section, the focus is on maximizing the total channel capacity by exploring transmitter (Tx) and receiver (Rx) beamforming angles. A wide range from -60 to +60 degrees in a 5-degree step size is covered, totaling 25^4 = 390,625 combinations. Fig. 2 shows the setup of a 3D room model ("office.stl") measuring 8m × 5m × 2.75m. In our model, the mmWave communication nodes are configured to operate at a frequency of 60 GHz with a bandwidth of 2 GHz. The transmit power is 0.5 W. The gain of each antenna varies based on its beamforming angle and is not constant. However, the maximum antenna gain reaches 15.95 dBi. The four nodes deploy antennas arranged in a 8 × 4 Uniform Rectangular Array (URA) each, with half-wavelength spacing between elements. The exact locations of transmitters (Tx) and receivers (Rx) are specified as:

Tx_1: (0.02, 8, 2)
Tx_2: (5, 8, 2)
Rx_1: (4.5, 3.5, 0.85)
Rx_2: (2, 4, 0.85)

It's important to note that the setup we're discussing here doesn't involve any unique or specific room conditions. Instead, it's based on a standard layout that's readily available within the MATLAB environment. We've opted for a general room setup and layout with a generalized configuration.

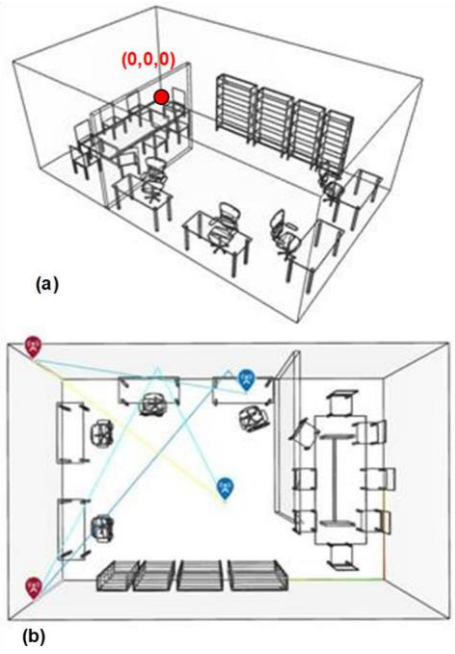

FIGURE 2. (a) The 3D room layout model. (b) The antenna placement in the model room.

Additionally, the choice of antenna locations isn't tied to any particular scenario. We anticipate that the conclusions drawn won't be affected even if we were to alter the rooms and locations of the antennas.

Raytracing is employed for channel modeling, allowing one maximum reflection with a "concrete" surface material to evaluate the channel capacity across various Tx and Rx angles. Antenna patterns are calculated within our computational framework using MATLAB's Phased Array System Toolbox, employing these four key steps:

1. Creating a steering vector to represent the antennaarray's spatial response.
2. Defining desired scan angles for the main lobe beam.
3. Calculating beamforming weights.
4. Using the pattern function to compute the antennaarray's radiation pattern.

Various radiation patterns are created by customizing scan angles and weights, considering antenna element options such as "patchMicrostrip" and "phased.CosineAntenna Element". Custom antenna elements are generated based on these patterns, and transmitter ("Tx") and receiver ("Rx") objects are constructed accordingly. Path characteristics for selected rays, including path loss, phase, angle of departure (AoD), and angle of arrival (AoA), are then calculated. The Signal-to-Noise and Interference Ratio (SNIR) depends on the specific interference generated by each transmitter for the others. This variability is influenced by the particular radiation pattern employed in the system.

IV. SIMULATION RESULTS

In the subsequent section, our approach will cover two distinct implementation phases. In the initial phase, emphasis will be placed on evaluating the precision and accuracyofthelearningmethods. Subsequently,inthesecond phase, when the computational demands of exhaustive search become impractical, according to the accuracy determined in Phase 1 will be utilized to predict optimal transmit-receive angles and calculate the maximum attainable channel capacity using these learning methods.

PHASE I: Validation Using 2 Tx and 2 Rx with Exhaustive Search Results

Recognizing the computational complexity involved in processing all 25^4 = 390,625 combinations of beam patterns, which necessitates a simulation time of 3 days, we decided to use classic reinforcement learning methods and the Monte Carlo method for finding the optimal solution within a reasonable time. Subsequently, a comparison is performed between the results derived from an exhaustive search and those obtained through machine learning. The exhaustive search, which demanded around 3 days, was performed using a machine equipped with the specification: 72 cores, 4 E7-4880 CPUs operating at a speed of 2.5 GHz, and 1TB RAM.

This evaluation aims to measure the time required for ascertaining the optimal antenna angles and estimating channel capacity. The primary objective of this comparison is to evaluate the accuracy of the classic reinforcement learning-based approach in contrast to the exhaustive search method.

Fig. 3 offers a comparison of different classic reinforcement learning methods and Monte Carlo, focusing on the average achieved channel capacity after 10 execution cycles for each method. The decision to conduct 10 execution cycles was made to ensure that the results were not influenced by random variations. The average of these runs provides a more reliable assessment. Additionally, the figure provides

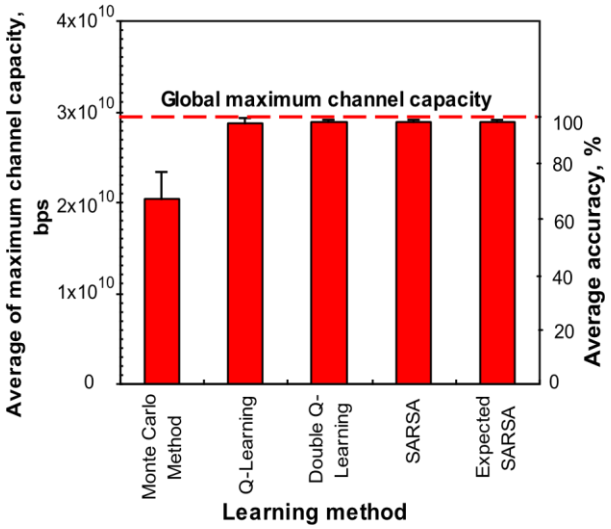

FIGURE 3. Comparison of reinforcement learning methods for channel capacity and accuracy for 2 Tx and 2 Rx.

the associated error values and accuracy averages for each method. The left axis of the figure represents channel capacity, measured in bits per second (bps), while the right axis illustrates the accuracy of the learning methods, expressed as a percentage. Among the methods, SARSA has the best performance in both channel capacity and accuracy. It achieves an impressive 99.88% accuracy and presents superior channel capacity results. Following SARSA, Expected SARSA obtains the second-highest accuracy at 99.85%. It shows great performance in terms of channel capacity, though it doesn't perform as well as SARSA. The remaining methods are ranked based on their accuracy, with Double Q-learning achieving 99.76%, Q-learning at 99.09%, and Monte Carlo at 70.14%. While these methods show varying levels of accuracy, they still provide reasonable solutions for enhancing channel capacity.

Table 1 displays the maximum total channel capacity achieved through the learning process, along with the corresponding optimal angles for all four antennas for each learning method. It is noteworthy that most methods converged to similar angle configurations for all four antennas and most of them have been able to find the maximum total channel capacity which is 29 Gbps.

TABLE 1. Maximum total channel capacity and optimal antenna angles.

Learning method	Tx-1 (o)	Tx-2 (o)	Rx-1 (o)	Rx-2 (o)	Channel capacity (Gbps)
Monte Carlo	-40	5	5	35	25.4
Q-Learning	-40	-40	45	55	29.0
Double Q-Learning	-40	-40	45	55	29.0
SARSA	-40	45	45	55	29.0
Expected SARSA	-40	-60	45	55	29.0

According to Table 1, Fig. 4 provides a schematic representation of the optimal positioning of transmitting and receiving

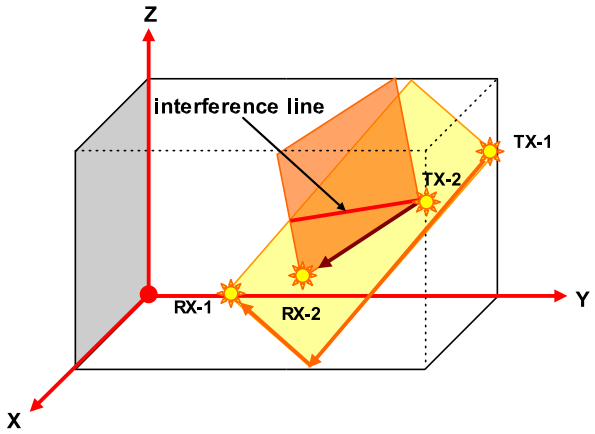

FIGURE 4. Schematic representation of maximum interference points according to 4 antennas placement in 2D space for 2 Tx and 2 Rx.

antennas to achieve maximum total channel capacity. To simplify the display in a 2D format, we assume that all antennas have a parallel viewing direction aligned with the Y-axis.

Arectangularcubeisdepicted, proportionallyscaledtomatch the dimensions of the room under examination, and antennas are positioned accordingly. The figure integrates data derived from the Q-learning method to determine emission angles and the line of rays. The intersection line between these planes which has been indicated by the red line, describes the location characterized by maximum interference. This intersection signifies the interference point in a scenario involving two transmitters and two receivers aligned in a single line, effectively representing the interference of two planes.

Fig. 5, displays the maximum standard deviation values for the implemented methods, which are the square root of their variances and it shows how much the values of each method vary from their averages. Monte Carlo exhibits the highest deviation at 2.93E+9 bps, while Q-Learning follows at 5.25E+9 bps. Double Q Learning shows a lower deviation of 1.1E+9 bps, SARSA at 8.5E+8 bps, and Expected SARSA with the lowest deviation at 8.3E+8 bps. This plot highlights Expected SARSA's superior consistency and accuracy compared to other methods, emphasizing its stability in optimizing channel capacity. In contrast, Monte Carlo exhibits higher variability, showing reduced reliability in channel capacity optimization.

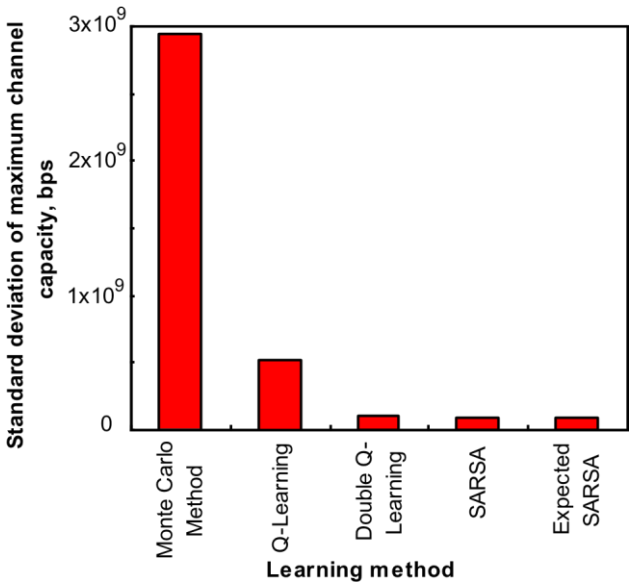

FIGURE 5. Standard deviation analysis of reinforcement learning methods for 2 Tx and 2 Rx.

Fig. 6 summarizes the analysis of how long it takes and the errors observed in 10 runs for each method. This comparison allows us to evaluate the efficiency of these methods, taking into account their varying execution times. Among the implemented methods, Monte Carlo demands the most significant computational time, which may appear surprising considering its lower accuracy. In contrast, SARSA demonstrates

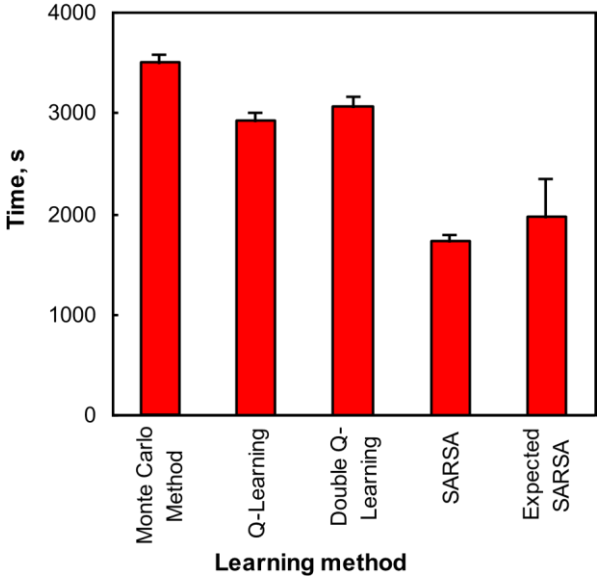

FIGURE 6. Computation time of learning methods for 2 Tx and 2 Rx.

the most efficient execution time, approximately 1724 s, making it an efficient choice. Expected SARSA shows the second-longest execution time, approximately 1971 s while presenting favorable accuracy. Q-learning ranks next in terms

of execution time, followed by Double Q-learning and Monte Carlo. The extended computational time for the Monte Carlo method can be attributed to its inherent reliance on random sampling and statistical approximation. Monte Carlo needs a large number of tries to make sure the result is close enough to the true solution with a good level of confidence. This high number of attempts contributes to the longer computational time.

Combining accuracy and complexity (execution time) we attempt to generate a single figure of merit for the different ML-techniques and Monte Carlo method. According to Equation 5, Fig. 7 uses effect sizes of 0.7 for accuracy and 0.3 for execution time, providing an evaluation of their combined impact on the observed results. Selecting the effect size is contingent on the preference. In our case, prioritizing accuracy instead of time led us to assign a larger coefficient to it. Fig. 7 shows that SARSA, Expected SARSA and Q-Learning methods exhibit superior performance, achieving a balance between time efficiency and accuracy with scores of 55%, 53%, and 44%, respectively. These methods deliver accurate results within shorter computational times, making them efficient choices for optimization tasks. In contrast, Double Q-Learning and the Monte Carlo demonstrate lower weighted efficiency scores.

$$\textit{Weighted efficiency} = (\textit{Normalized Accuracy of Algorithms} * 0.7) - (\textit{Normalized Execution Time of Algorithms} * 0.3) \quad (5)$$

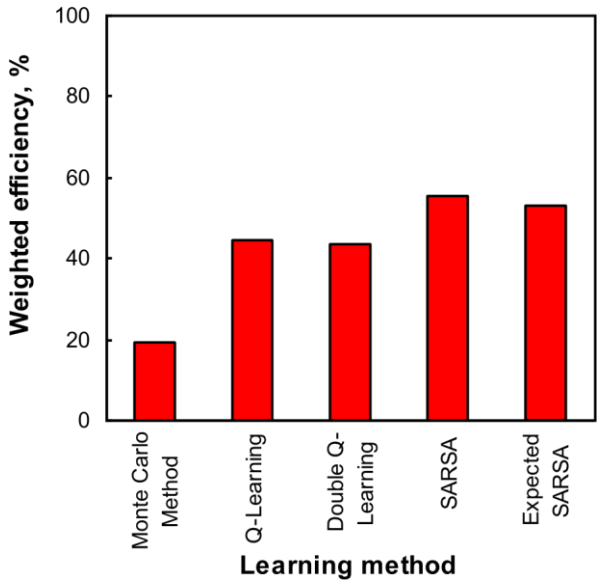

FIGURE 7. Weighted efficiency scores of RL methods for 2 Tx and 2 Rx.

The choice of effect size depends on the specific preferences and priorities of the analysis. In our case, our priority was accuracy rather than time efficiency, which led us to assign a larger coefficient to accuracy in Equation 5. We prioritize accuracy in our results, even if it means taking extra time for computations.

PHASE II: Prediction using 3 Tx and 3 Rx without Exhaustive Search Results

In the next step, after our evaluation and accuracy assessment of the classic reinforcement learning methods in the prior step, the configuration is extended by adding one additional-pair of transmitter and receiver, creating a scenario with 3 transmitters and 3 receivers. This expansion introduces a total of 25^6 = 244,140,625 distinct states, exceeding the computational capabilities of our system for exhaustive search.

Consequently, machine learning techniques and the Monte Carlo method are exclusively employed to estimate optimal angles and calculate the maximum total channel capacity in this context. To enhance the accuracy of the result and ensure convergence, the machine learning code execution comprises 20,000 iterations. The positioning of the new antennas within the room is outlined below:

Tx_3: (1,2.75, 2)
Rx_3: (2.5,6.5,1)

Fig. 8 compares different methods, focusing on the average channel capacity after 10 runs. The measurements are in bits per second (bps). Double Q-learning stands out as the top performer with a channel capacity of 2.47E+10 bps. Following closely is Q-learning, achieving the second-highest channel capacity at 2.40E+10 bps, showing commendable performance but slightly trailing Double Q-learning. Expected SARSA reaches a channel capacity of 2.39E+10 bps,

SARSA attains 2.38E+10 bps, and Monte Carlo records

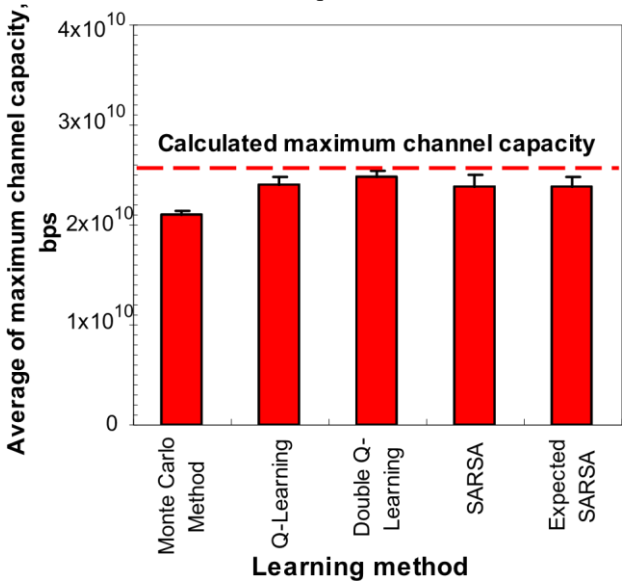

FIGURE 8. Comparative analysis of reinforcement learning methods for channel capacity (average after 10 runs) for 3 Tx and 3 Rx.

2.10E+10 bps. These rankings are based on their respective channel capacity measurements, offering insights into their comparative performance.

Table 2 presents the maximum total channel capacity attained via the learning process, in addition to the optimal antenna angles for all six antennas, as per each learning method. It's worth mentioning that several of these methods tend to converge on similar angle configurations for particular antennas. Moreover, the majority of them have approximated the total channel capacity to be around 2.5E+10 bps.

TABLE 2. Prediction of maximum total channel capacity and optimal antenna angles FOR 3 Tx and 3 Rx.

Learning method	Tx-1 (o)	Tx-2 (o)	Tx-3 (o)	Rx-1 (o)	Rx-2 (o)	Rx-3 (o)	Channel capacity (Gbps)
Monte Carlo	-50	-55	-30	-50	0	-30	17.8
Q-Learning	-40	-30	-5	45	55	-30	25.5
Double Q-Learning	-40	30	-5	45	55	45	25.5
SARSA	-40	-15	-5	45	55	15	25.2
Expected SARSA	-40	-15	-5	45	55	15	25.2

Fig. 9 presents a schematic diagram, similar to Fig. 4, with 3 transmitters and 3 receivers. In this figure, we have incorporated the angles derived from the Q-learning method to show the positions of six antennas. Consequently, the intersection points among these three planes, representing each pair of transmitters and receivers, signify areas with the highest interference levels. It's essential to note that these points do not form an interference line; rather, they make an interference plane. Any ray path traversing

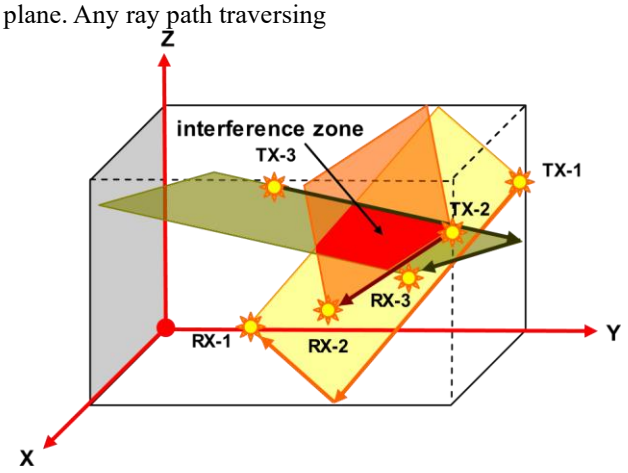

FIGURE 9. Schematic representation of maximum interference points according to 6 antennas placement in 2D space for 3 Tx and 3 Rx.

this plane will experience increased interference, resulting in reduced channel capacity compared to alternative pathways.

Fig. 10 presents the maximum standard deviation values for the maximum channel capacity of various reinforcement learning methods and Monte Carlo Method, showing their respective values in bits per second (bps). SARSA has the highest deviation at 1.1E+9 bps, followed by Expected SARSA at 9.5E+8 bps. On the other hand, Q-Learning shows a lower deviation of 7.6E+8 bps, and Double Q Learning has even less deviation at 6.7E+8 bps. Notably, Monte Carlo exhibits the lowest deviation at 3.2E+8 bps. The difference in standard deviation between Monte Carlo and SARSA is attributed to their exploration strategies. The Monte Carlo method completes episodes before updating, resulting in more stable output. In contrast, SARSA's on-policy approach

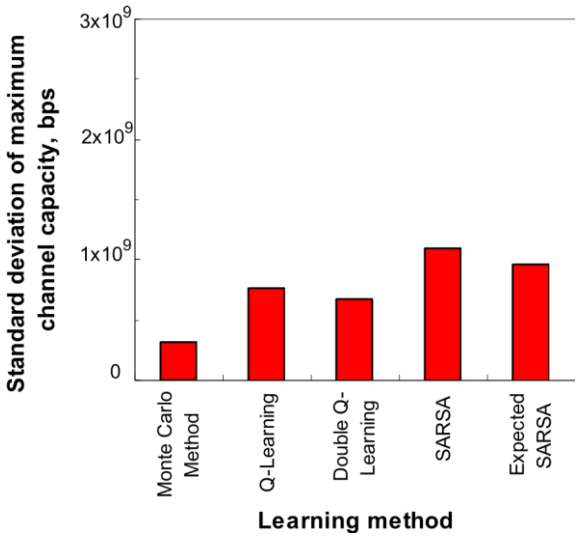

FIGURE 10. Standard deviation analysis of reinforcement learning methods for 3 Tx and 3 Rx.

introduces variability due to exploration, leading to a higher standard deviation. In addition, Monte Carlo method in prediction phase has less standard deviation than validation phase because the implementation covers a broader range of scenarios, accounting for various configurations. This diversity helps mitigates the influence of outlier scenarios and making the results more stable and less variable.

Fig. 11 provides an analysis of computational time versus error values across 10 runs for each implemented method. This plot highlights the crucial role of machine learning in handling demanding computational tasks. In our scenario, performing an exhaustive search would have demanded around 1875 days, rendering it practically unfeasible, despite utilizing a powerful machine with 72 cores. However, by employing reinforcement learning techniques, we can achieve (near-)optimal solutions in less than 15 minutes. Among the methods, Double Q-learning demands the most extensive computational time, approximately 8985 s. SARSA follows with the second-longest execution time, roughly 8740 s. Expected SARSA ranks third in execution time, around 6556 s. Q-learning occupies the next position, with an execution time of 4156 s, followed by Monte Carlo, which, despite its shorter execution time of 3390 s,

yielded a lower maximum total channel capacity

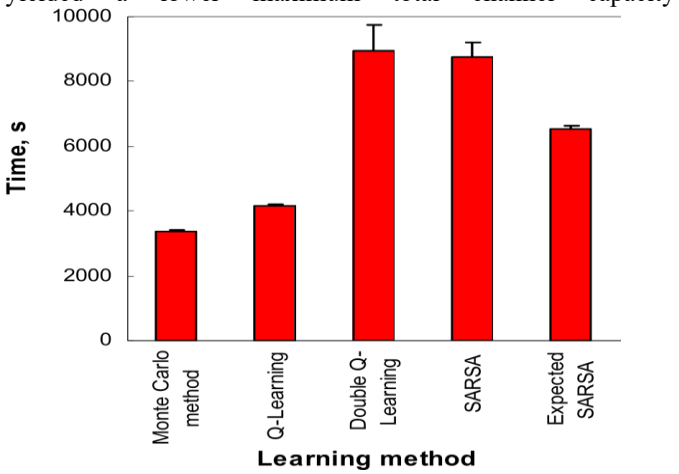

FIGURE 11. Computational time of learning methods for 3 Tx and 3 Rx.

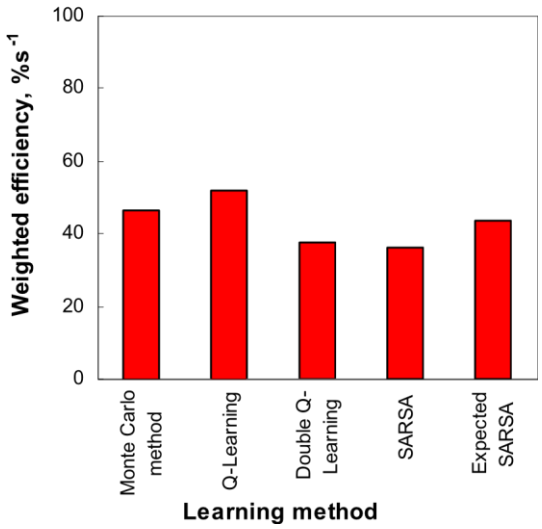

FIGURE 12. Relative efficiency of channel capacity optimization methods with computational time considerations for 3 Tx and 3 Rx.

It involves comparing the maximum channel capacity achieved by each method after 10 runs to the maximum channel capacity determined by all learning models and Monte Carlo method, which is attributed to the Q-learning method. This comparison is achieved by dividing the former by the latter and multiplying the result by 0.7, emphasizing the importance of capacity optimization.

Furthermore, we normalize the average execution time of each method after 10 runs by dividing it by the average of the longest execution time, associated with the Double Q-learning method. Finally, we subtract the result of the execution time normalization from the capacity optimization factor, providing the relative efficiency (or 'figure of merit') of the implemented methods in optimizing channel capacity by considering the computational time. It should be mentioned that the increase from 4 nodes to 6 nodes notably benefited the Monte Carlo approach. This can be attributed to its inherent scalability, efficient handling of increased combinatorial complexity, and the provision of realistic uncertainty quantification. The stochastic nature of Monte Carlo simulations proved particularly advantageous in capturing the complexities of larger communication systems.

V. CONCLUSION

In this investigation, we compare different techniques of machine learning (ML) for application in wireless communications. The research involved two main phases: initially validating various learning methods and the Monte Carlo method through an exhaustive search and then applying classic reinforcement learning techniques and the Monte Carlo technique to optimize total channel capacity, determining optimal transmitter and receiver beamforming antenna angles. Due to the impractical computational complexity of exhaustive search, classic reinforcement learning techniques are essential. Our results highlighted the accuracy of SARSA, Expected SARSA, Q-Learning, and Double Q-Learning methods, all achieving 99% accuracy with two transmitters and two receivers. Particularly, when dealing with configurations involving three transmitters and three receivers, these methods outperform the Monte Carlo approach. Among them, Double Q-learning appears with a higher average channel capacity, being the best method evaluated. It's noteworthy that the Q-learning family, overall, outperformed the SARSA family, though Double Q-learning required a more significant computational time investment. Choosing between these methods depends on application-specific priorities and constraints. However, our work emphasizes that using classic reinforcement learning can compete with exhaustive search results in much shorter execution time periods. The investigated reinforcement learning techniques show promising results in handling complex tasks in wireless communications that involve large amounts of data. As a future work, our model could be extended beyond current limitations, exploring the application of machine learning to optimize antenna configurations in a dynamic real urban environment.

REFERENCES

[1] [1] B. Fong, H. Kim, A. C. M. Fong, G. Y. Hong, and K. F. Tsang,

"Reliability optimization in the design and implementation of 6G vehicleto-infrastructure systems for emergency management in a smart city environment,"*IEEECommun.Mag.*,vol.61,no.8,pp. 148–153,Jul.2023. [2] N. Haider, M. Zeeshan Baig, and M. Imran, "Artificial intelligence and machine learning in 5G network security: Opportunities, advantages, and future research trends," 2020, *arXiv:2007.04490*.

[3] A. Nauman, T. N. Nguyen, Y. A. Qadri, Z. Nain, K. Cengiz, and S. W. Kim, "Artificial intelligence in beyond 5G and 6G reliable communications," *IEEE Internet Things Mag.*, vol. 5, no. 1, pp. 73–78, Mar. 2022.

[4] J. Mills, J. Hu, and G. Min, "Client-side optimization strategies for communication-efficient federated learning," *IEEE Commun. Mag.*, vol. 60, no. 7, pp. 60–66, Jul. 2022.

[5] W. Yu, F. Sohrabi, and T. Jiang, "Role of deep learning in wireless communications," *IEEE BITS Inf. Theory Mag.*, vol. 2, no. 2, pp. 56–72, Nov. 2022.

[6] T. O'Shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Trans. Cognit. Commun. Netw.*, vol. 3, no. 4, pp. 563–575, Dec. 2017.

[7] Z. Qin, H. Ye, G. Y. Li, and B. F. Juang, "Deep learning in physical layer communications," *IEEE Wireless Commun.*, vol. 26, no. 2, pp. 93–99, Apr. 2019, doi: 10.1109/MWC.2019.1800601.

[8] M. Chen, U. Challita, W. Saad, C. Yin, and M. Debbah, "Artificial neural networks-based machine learning for wireless networks: A tutorial," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 4, pp. 3039–3071, 4th Quart., 2019.

[9] Y. C. Eldar, A. Goldsmith, D. Gunduz, and H. V. Poor., *Machine Learning and Wireless Communications*. Cambridge, U.K.: Cambridge Univ. Press, 2022, doi: 10.1017/9781108966559.

[10] E. Bourtsoulatze, D. Burth Kurka, and D. Gündüz, "Deep joint sourcechannel coding for wireless image transmission," *IEEE Trans. Cognit. Commun. Netw.*, vol. 5, no. 3, pp. 567–579, Sep. 2019.