ITUJournal
Future and evolving technologies

ITU Journal on Future and Evolving Technologies, Volume 6, Issue 4, December 2025

ARTICLE

Enhancing multiuser scheduling in massive MIMO mobile channels

Sara Al-kokhon[1,2], Hossein Bijanrostami[1], Elaheh Bassak[1], Brad Stimpson[2], Elvino Sousa[1]

[1] Department of Electrical and Computer Engineering, University of Toronto, Canada, [2] Bell Canada

Corresponding author: Sara Al-kokhon, sara.al.kokhon@utoronto.ca

Massive Multiple-Input Multiple-Output (MIMO) is a key enabler of 5G and beyond mobile networks, significantly improving spectral efficiency through multiuser beamforming. However, in massive MIMO systems, the multiuser scheduling problem, selecting which users to serve concurrently on the same time-frequency resources, remains a critical challenge. Due to potential channel correlation among users, suboptimal multiuser scheduling can lead to inter-symbol interference and throughput degradation. Additionally, the scheduler must balance the achieved spectral efficiency with user fairness. While the Optimal Proportional Fair (Opt-PF) scheduler seeks to achieve this balance, applying it to the massive MIMO scheduling problem leads to an NP-hard optimization problem. Although existing approximation algorithms can reduce the computational complexity of the Opt-PF multiuser scheduler, they often fail to provide adequate fairness or adapt to fast varying channels, making them impractical for real-world deployment. As an alternative, Machine Learning (ML)-based methods, particularly Deep Reinforcement Learning (DRL) models, have shown promise in addressing this problem. To further foster innovation in this area, the International Telecommunication Union (ITU) AI/ML in 5G Challenge hosted a competition focused on enhancing the performance of a DRL-based multiuser scheduler. The provided baseline scheduler employed a user-grouping algorithm to cluster users with low channel correlation and a Soft Actor-Critic (SAC) DRL framework for user selection. This paper presents the winning solution to the ITU competition, which proposes two approaches to enhance the performance of the baseline scheduler. The first approach redefines the baseline's SAC DRL framework and redesigns the underlying neural network architecture, whereas the second approach applies an ML-based clustering algorithm, specifically the Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN), to the user-grouping problem. Both approaches outperformed the baseline scheduler in terms of user fairness, sum rate, and computational complexity.

Keywords: Deep Reinforcement Learning (DRL), HDBSCAN, massive MIMO, ML-based clustering, multiuser scheduling, Soft Actor-Critic (SAC)

1. INTRODUCTION

5G networks are designed to deliver significantly higher data rates and greater capacity to meet the growing demands of mobile users and IoT devices. A foundational technology enabling these enhancements is Multiple-Input Multiple-Output (MIMO). MIMO systems use multiple antennas at both the Base Station (BS) and the User Equipment (UE) to improve link robustness through transmit and receive diversity, increase data rates via spatial multiplexing, and boost network capacity through multiuser beamforming. To further accommodate the increasing traffic and user density, massive MIMO has been adopted as a key 5G technology. By significantly increasing the number of antennas at the BS, massive MIMO enables more efficient multiuser beamforming and allows a larger number of users to be served concurrently on the same time-frequency resources.

A critical challenge in massive MIMO systems is multiuser scheduling, which involves determining the optimal set of users for simultaneous transmission over the same radio resource. Since scheduling decisions directly impact massive MIMO system performance, it has been a central focus of research. In particular, achievable system throughput and

capacity gains can degrade significantly if the selected users exhibit high correlation in their wireless channels, as this impairs beamforming efficiency in two fundamental ways: (i) in the downlink, constructing orthogonal beams toward the scheduled users becomes difficult due to their channel correlation; (ii) in the uplink, accurately recovering the users' transmitted signals becomes challenging, as correlated user channels increase the likelihood of inter-user interference.

Furthermore, in rapidly changing environments, such as those experienced by highly mobile users, the user's channel conditions can vary rapidly across frequency resources. This dynamic behavior makes optimal user scheduling significantly challenging, demanding efficient and adaptive techniques to maximize spectral efficiency and overall network performance.

A well-known optimal scheduler is the Optimal Proportionally Fair (Opt-PF) scheduler. This scheduler seeks to maximize the spectral efficiency while ensuring user fairness. However, applying this scheduler to the massive MIMO scheduling problem leads to an NP-hard optimization problem [1], making it computationally intractable for real-time or large-scale deployments. To address this challenge, various approaches have been proposed in the literature. These approaches can be broadly categorized into two groups: heuristic-based methods and AI-based methods.

Heuristic-based scheduling methods, [2], [3], [4], approximate the Opt-PF scheduler by reducing the computational overhead while maintaining high spectral efficiency. Although these methods aim to balance performance and complexity, they often lack formal fairness guarantees and scalability, limiting their applicability in practical massive MIMO systems. These limitations have motivated the exploration of AI-based solutions, particularly those based on Deep Reinforcement Learning (DRL). DRL-based scheduling approaches [5], [6], [7], [8], [9], model the multiuser scheduling problem as a Markov Decision Process (MDP) and learn proportional fair scheduling policies by interacting with the environment. This model-free learning framework enables adaptive decision-making under dynamic network conditions while capturing long-term trade-offs between spectral efficiency and user fairness.

In AI-based scheduling methods, the choice of the underlying DRL model plays a critical role, as the scheduling problem inherently involves a discrete action space. In general, DRL models for this problem can be classified into two categories:

(i) **DRL models with discrete action spaces**: These models naturally align with the discrete nature of the user selection problem without requiring structural modifications to the model's output. Examples of such models include Deep Q-Networks (DQN) [10], Double DQN [11], Advantage Actor-Critic (A2C) [12], Asynchronous Advantage Actor-Critic (A3C) [13], Actor-Critic with Experience Replay (ACER) [14], and Proximal Policy Optimization (PPO) [15]. While these models offer direct compatibility with discrete action settings, they struggle to scale in scenarios with large discrete action spaces (equivalently large number of users), which naturally arise in real-world user scheduling problems.

(ii) **DRL models with continuous action spaces**: These models can be adapted to address the discrete nature of the user selection problem. Examples include Deep Deterministic Policy Gradient (DDPG) [16] and Soft Actor-Critic (SAC) [17]. These models offer a potential solution to scalability challenges by leveraging continuous-action formulations; however, their application to inherently discrete scheduling tasks remains an active area of research.

In [1], the authors employ a continuous-action SAC DRL framework to solve the multiuser scheduling problem in massive MIMO networks. To convert the SAC's output to discrete actions, they combine SAC with the K-Nearest Neighbors (KNN) algorithm [18] to generate discrete actions corresponding to user scheduling decisions. Additionally, they propose a dimension division strategy that maps the discrete action sets to multiple dimensions to enable robust scalability. However, the use of KNN introduces additional computational overhead, increasing the scheduler's runtime. On the other hand, the authors attempt to reduce the complexity of the scheduler by reducing the dimensionality of the SAC input space. To capture the inter-user channel correlation, they use a threshold-based user grouping algorithm that clusters uncorrelated users and assigns user grouping labels, which are further used in the model's state input instead of the raw Channel State Information (CSI) matrix. While this approach reduces state dimensionality, it incurs information loss and additional runtime due to the inefficiency of the user grouping process.

In this paper, we propose two approaches to enhance the performance of the multiuser scheduler proposed in [1]. In the first approach, we redesign the SAC DRL framework and the underlying neural networks architecture. The proposed SAC DRL framework incorporates new state and reward definitions and minor modifications to the baseline's discretization method. The new state representation omits the baseline user grouping, and instead, leverages a compact and informative representation of the full channel correlation matrix. Specifically, we construct the state space using the non-diagonal elements of the upper triangular part of the correlation matrix, combined with the ℓ_2 norms of each user's channel vec-

tor. This compact formulation retains crucial information about the inter-user correlation while avoiding the dimensionality explosion of using the full channel matrix. In addition, we replace the baseline's KNN action discretization module with a more efficient approach. Furthermore, we redesign the critic's neural network architecture to enhance the actor's performance, and we take a novel approach to incorporating the user fairness criteria in our SAC model. Unlike the baseline model [1], which explicitly incorporates a fairness term into the reward function, our approach exploits the stochastic nature of the SAC policy to implicitly promote fairness among users.

In the second approach, we enhance the user grouping mechanism by introducing a new user grouping technique based on the Machine Learning (ML)-based Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) algorithm [19]. Unlike the baseline method, the HDBSCAN-based user grouping does not require a predefined correlation threshold or a fixed number of clusters, as is typical in conventional clustering algorithms. Instead, it constructs a hierarchy of clusters based on varying density levels and identifies the most stable configuration. This allows it to adapt more effectively to inter-user correlation structure variations and form more robust user groupings. By applying HDBSCAN to the inter-user channel correlation matrix, our model forms more coherent user groups and identifies outlier users as noise, enabling better input representations and improved scheduling performance, without requiring modifications to the SAC architecture. Additionally, the efficient implementation of HDBSCAN [20] helps reduce the overall computational complexity of the scheduler.

Challenge description: Given the important role of DRL methods in solving the multiuser scheduling problem in massive MIMO systems and to further foster innovations in this field, the International Telecommunication Union (ITU) AI/ML in 5G Challenge (Fifth edition, 2024) hosted a competition [21] focused on improving the performance of the SAC-based multiuser scheduler presented in [1]. The competition emphasized optimizing three key performance metrics: sum rate, runtime, and user fairness. In this paper, we present the winning solution announced in [22] to the ITU competition described in [21], which builds upon the baseline's solution by exploring two independent enhancement approaches.

The remainder of this paper is organized as follows. Section 2 reviews the multiuser scheduling problem in massive MIMO systems and presents the baseline scheduling solution. Section 3 presents the two enhancement approaches proposed in the winning solution. Approach I redefines the baseline's SAC DRL framework and the underlying neural network architecture, while Approach II improves the user-grouping strategy using the HDBSCAN algorithm. Section 4 reports the simulation setup and the results. Finally, Section 5 concludes the paper and outlines directions for future work.

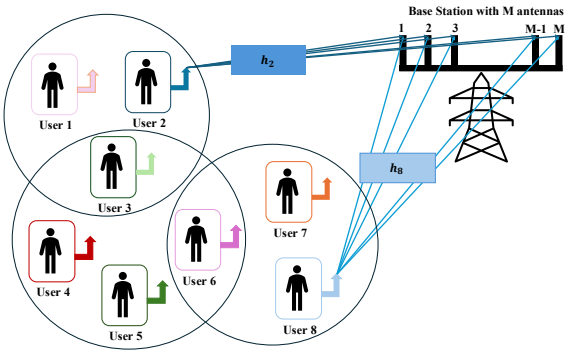

Figure 1 – Illustration of a massive MIMO cellular system with a single BS equipped with M antennas and $L = 8$ single-antenna users. Users with correlated channels are grouped into clusters, and the BS simultaneously serves uncorrelated User 2 and User 8 over the same time-frequency resources using multiuser beamforming.

2. BACKGROUND

In this section, we first provide an overview of the multiuser scheduling problem in massive MIMO systems, and then describe the baseline solution provided in the ITU AI/ML in 5G challenge [21] that the participants were asked to enhance in terms of the sum rate, fairness, and runtime (inference time).

2.1 Multiuser scheduling problem in massive MIMO systems

To illustrate the multiuser scheduling problem in massive MIMO systems, as in [1], we consider an OFDM system with a single BS equipped with M antennas serving L single antenna users. In each Transmission Time Interval (TTI) t, the BS selects N users, where $N <= L$ and $N <= M$, for simultaneous transmission over the same time-frequency resources. In 5G, the smallest transmission unit in the frequency domain is known as a Physical Resource Block (PRB), consisting of 12 subcarriers.

To enable multiuser transmissions and receptions over the same PRB, multiuser beamforming is applied at the BS. In multiuser beamforming, the channel matrix is used to recover the transmitted signals in the uplink and to form orthogonal beams in the downlink. As the selection of the N users shapes the channel matrix, their selection directly impacts the performance of the multiuser beamforming applied at the BS. More specifically, selecting users with correlated channels impairs the beamforming performance, impacting the massive MIMO gains. Therefore, the BS must carefully select uncorrelated users for simultaneous uplink and downlink transmissions. Fig. 1 illustrates a cell with a massive MIMO setup.

We further illustrate the multiuser scheduling problem mathematically below. We consider uplink transmission; however, the same approach can be applied to downlink transmission. Let the uplink channel between user i and the M BS antennas be represented by

$$\mathbf{h}_i = \left[h_{1,i}, h_{2,i}, h_{3,i}, \ldots, h_{M,i}\right]^T, \quad (1)$$

where $\mathbf{h}_i \in \mathbb{C}^{M\times 1}$ is the channel vector of user i, and $h_{m,i} \in \mathbb{C}$ denotes the channel coefficient between receive antenna m and user i. The channel correlation coefficient between users (i, j) is denoted by $c_{i,j}$ and is calculated as

$$c_{i,j} = \left\langle \frac{\mathbf{h}_i}{\|\mathbf{h}_i\|_2}, \frac{\mathbf{h}_j}{\|\mathbf{h}_j\|_2} \right\rangle = \frac{\mathbf{h}_i^H \mathbf{h}_j}{\|\mathbf{h}_i\|_2 \|\mathbf{h}_j\|_2}. \quad (2)$$

At the BS, the received signal vector over a single subcarrier is given by

$$\mathbf{y} = \sum_{i=1}^{N} x_i \mathbf{h}_i + \mathbf{n}, \quad (3)$$

where $\mathbf{y} \in \mathbb{C}^{M\times 1}$ is the received signal vector, $x_i \in \mathbb{C}$ is the symbol transmitted by user i, and $\mathbf{n} \in \mathbb{C}^{M\times 1}$ is the receiver noise vector with distribution $\mathbf{n} \sim \mathcal{CN}(\mathbf{0}, \sigma^2 \mathbf{I}_M)$. Eq. *(3)* can be expressed compactly as

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n}, \quad (4)$$

where $\mathbf{H} \in \mathbb{C}^{M\times N}$ is the channel matrix with columns $\mathbf{h}_i$, and $\mathbf{x} \in \mathbb{C}^{N\times 1}$ is the transmitted symbol vector.

To recover $\mathbf{x}$ from $\mathbf{y}$, the zero-forcing beamforming (ZF-BF) receiver is employed at the BS. The ZF-BF weight matrix is calculated using the estimated channel matrix $\hat{\mathbf{H}} \in \mathbb{C}^{M\times N}$ as

$$\mathbf{W} = \hat{\mathbf{H}} \left(\hat{\mathbf{H}}^H \hat{\mathbf{H}}\right)^{-1}, \quad (5)$$

where $\mathbf{W} \in \mathbb{C}^{M\times N}$ is the ZF-BF weight matrix.

Applying ZF-BF to the received vector yields

$$\hat{\mathbf{x}} = \mathbf{W}^H \mathbf{y} = \left(\hat{\mathbf{H}}^H \hat{\mathbf{H}}\right)^{-1} \hat{\mathbf{H}}^H (\mathbf{H}\mathbf{x} + \mathbf{n}), \quad (6)$$

where $\hat{\mathbf{x}} \in \mathbb{C}^{N\times 1}$ is the ZF-BF estimate of the transmitted symbol vector $\mathbf{x}$.

From Eq. *(6)*, it follows that for $\left(\hat{\mathbf{H}}^{\mathrm{H}} \hat{\mathbf{H}}\right)^{-1}$ to exist and for the ZF-BF receiver to be numerically stable, $\hat{\mathbf{H}}$ must have full column rank ($\mathrm{rank}(\hat{\mathbf{H}}) = N$) and the Gram matrix $\hat{\mathbf{H}}^{\mathrm{H}} \hat{\mathbf{H}}$ must be well-conditioned (i.e., it has a small condition number). This is promoted by selecting users with low channel correlation, e.g., enforcing

$$c_{i,j} \le \tau, \quad \text{for all } i \ne j,$$

where $\tau \in [0, 1)$ is a design threshold.

In addition to managing channel correlation, the multiuser scheduler must also account for user fairness and computational complexity, both of which are essential for practical and scalable deployment in real-world systems.

2.2 Baseline solution

To solve the multiuser scheduling problem in massive MIMO systems, the authors in [1] propose a dynamic AI-based scheduler called SMART. SMART serves as the baseline solution in the ITU AI/ML in 5G Challenge, and is based on the SAC DRL model. It also incorporates the following techniques:

- **User grouping algorithm**: Clusters users with low channel correlation ($c_{i,j} \le$ *Threshold*), and assigns a group label to each cluster. These group labels are further used in the SAC state input.
- **Output discretization**: Maps continuous proto-actions generated by the SAC policy to the discrete scheduling space.
- **K-Nearest Neighbors (KNN)**: Selects the closest discrete actions to mitigate precision loss from discretization.
- **Dimension division**: Splits the discrete action space into multiple lower-dimensional subspaces to improve resolution and scalability in large action spaces.

An overview of the baseline SAC model and its key techniques is presented in the following subsections.

2.2.1 Baseline SAC model

SAC is an off-policy deep reinforcement learning algorithm designed for learning stochastic policies with continuous action spaces [23]. Unlike traditional DRL methods that solely aim to maximize the expected return, SAC introduces an entropy term into its objective function. This encourages the agent to explore the action space more thoroughly and avoid premature convergence to suboptimal policies. SAC is known for its sample efficiency, stability, and robustness.

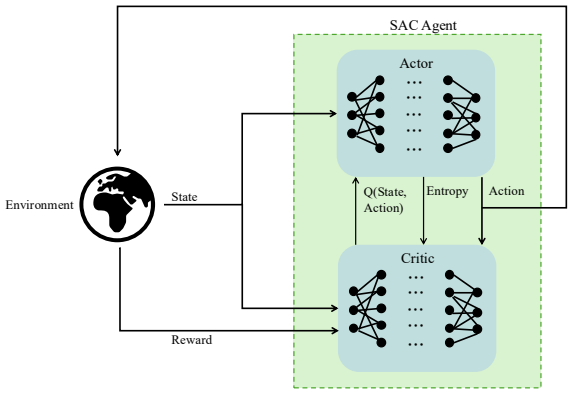

Figure 2 – SAC framework. The actor receives the current environment state and samples an action from the learned stochastic policy. The critic evaluates the Q-value of state-action pairs, guiding the actor during training to improve policy performance.

In SAC, the agent consists of two main components: the **actor** and the **critic**. The actor employs a neural network to learn a stochastic policy, i.e., a probability distribution over actions, by maximizing the following objective function [23]:

$$\pi^* = \arg\max_{\pi} \mathbb{E}_{(s_t,a_t)\sim\rho_t}\left[\sum_t R(s_t,a_t) + \alpha\mathcal{H}\Big(\pi(\cdot \mid s_t)\Big)\right], \quad (7)$$

where:

- s_t is the state at time t,
- a_t is the action taken at time t,
- ρ_t is the state–action distribution induced by the policy π,
- $R(s_t, a_t)$ is the reward received at time t,
- $\alpha \in \mathbb{R}_+$ is the temperature parameter that controls the trade-off between reward maximization and entropy, and
- $\mathcal{H}(\pi(\cdot|s_t))$ denotes the entropy of the policy at state s_t.

The entropy term is defined as

$$\mathcal{H}(\pi(\cdot|s_t)) = -\sum_a \pi(a|s_t)\log\pi(a|s_t), \quad (8)$$

where $\pi(a|s_t)$ is the probability of selecting action a under policy π given state s_t.

As shown in Fig. 2, in SAC, the actor receives the current state of the environment and outputs an action that is sampled from the learned stochastic policy π. In contrast, the critic takes as input the state-action pair and uses a neural network to estimate the corresponding Q-value. The critic is used only during the actor's training phase, where the estimated Q-values are used in the actor's policy update. During the critic's training, both the observed reward and the entropy of the policy are used in its network update.

The baseline SMART scheduler [1] employs the SAC algorithm by formulating the multiuser scheduling problem in massive MIMO systems as a Markov Decision Process (MDP). This MDP consists of three fundamental components: the state space, the action space, and the reward function. The baseline MDP is defined as follows.

- **State space**: The state $s_{i,t}$ for user i at TTI t is defined as

$$s_{i,t} = [\gamma_{i,t}, f_{i,t}, g_{i,t}] \in \mathcal{S} := [\mathcal{T}, \mathcal{F}, \mathcal{G}], \quad (9)$$

  where $\gamma_{i,t}$ denotes the maximum achievable single-user MIMO spectral efficiency for user i at TTI t, and $\mathcal{T} \subseteq \mathbb{R}_+ = [0, \infty)$ represents the set of all feasible values of $\gamma_{i,t}$; $f_{i,t}$ is the cumulative amount of data transmitted by user i up to TTI t, and $\mathcal{F} \subseteq \mathbb{R}_+ = [0, \infty)$ is the set of feasible $f_{i,t}$; and $g_{i,t}$ is the group label associated with user i at TTI t, with $\mathcal{G} := \{1, 2, \ldots, L\}$ denoting the finite set of possible user group labels. The total state input size is equal to $3 \times L$.

- **Action space**: The action at TTI t is represented by

$$\mathbf{a}_t = [a_{1,t}, \ldots, a_{i,t}, \ldots, a_{L,t}] \in \mathcal{A} := \{0, 1\}^L, \quad (10)$$

  where $\mathbf{a_t}$ represents the binary user selection vector at TTI t with $a_{i,t} = 1$ indicating the selection of user i for transmission in TTI t and $a_{i,t} = 0$ indicating otherwise. The action space $\mathcal{A}$ comprises all binary vectors $\mathbf{a}_t$ that satisfy the scheduling constraint $|\mathbf{a}_t| \leq N_{\max}$ with $|\mathbf{a}_t|$ denoting the total number of scheduled users in TTI t and $N_{\max}$ representing the maximum number of users to be scheduled in one TTI. This constraint ensures that no more than $N_{\max}$ users are selected for transmission at any given t.

- **Reward**: The reward function r_t is designed to balance system throughput and user fairness. r_t is defined as

$$r_t = c \cdot \left[\beta\gamma_t^{\text{total}} + (1-\beta)JFI_t\right], \quad (11)$$

  where γ_t^{total} represents the total spectral efficiency at TTI t, JFI_t denotes the Jain's Fairness Index at TTI t, $\beta \in [0, 1]$ is a weight parameter controlling the trade-off between fairness and throughput, and c is a reward scaling constant. γ_t^{total} is calculated as the sum of the spectral efficiencies of the scheduled users, normalized by the sum of the maximum achievable rates of the $N_{\max}$ users. JFI_t is calculated as:

$$JFI_t = \frac{\left(\sum_{i=1}^{L} f_{i,t}\right)^2}{L\sum_{i=1}^{L}(f_{i,t})^2}, \quad (12)$$

  Incorporating Jain's Fairness Index into the reward function encourages the scheduler to allocate resources evenly across users over time.

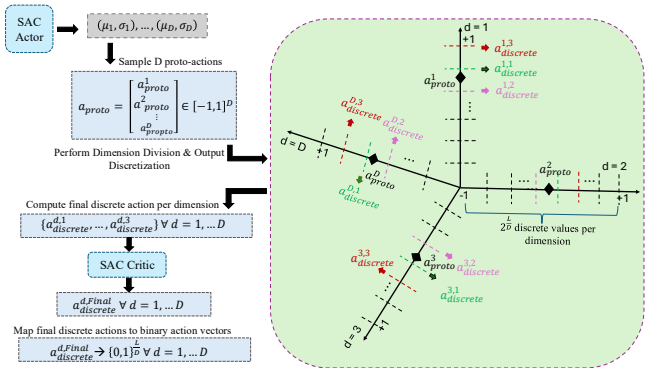

Figure 3 – Baseline discretized SAC architecture using KNN ($K = 3$). The nearest three discrete actions to the proto-action in each dimension are selected for further evaluation.

2.2.2 User grouping algorithm

DRL methods for massive MIMO scheduling often face convergence challenges as the dimensionality of the neural network input increases, an issue commonly referred to as the curse of dimensionality. To address this, SMART replaces the full complex channel matrix input $H \in \mathbb{C}^{M \times L}$, which is typically used to capture inter-user correlations, with scalar group labels derived from the inter-user channel correlation matrix [1]. The proposed user grouping algorithm clusters users with correlation coefficient $c_{i,j} < Threshold$ into the same group and assigns them a shared group label. These labels are then included in the SAC model's state input, effectively reducing the dimensionality of the state space and the overall model complexity.

Although the user grouping method succeeds in reducing the state space size, it introduces information loss that can negatively affect the model's ability to select optimal actions. In particular, the correlation distances between users within the same cluster and across different clusters are not preserved. Another limitation of this approach lies in the computational complexity of the grouping algorithm, which significantly increases the model's inference time, and consequently, the scheduler's runtime.

2.2.3 Output discretization

The massive MIMO user scheduling problem inherently involves selecting a subset of users for each transmission, making the action space discrete and combinatorial. While this might suggest using DRL models with discrete action spaces, such as DQN, the exponential growth of the action space with the number of users (2^L combinations) renders traditional discrete DRL models infeasible.

To address this challenge, the approach in [1] employs the SAC algorithm, which is designed for continuous action spaces. SAC generates a continuous proto-action $a_{\text{proto}} \in [-1, 1]$. To map this continuous output to a valid scheduling decision $\mathbf{a}_t \in \{0,1\}^L$, the range $[-1, 1]$ is discretized into 2^L intervals, with each interval corresponding to one of the possible binary user selection vectors. The proto-action a_{proto} is then mapped to the nearest interval, and the corresponding discrete scheduling action $\mathbf{a}_t$ is selected. However, this approach suffers from accuracy loss due to quantization; as L increases, the number of intervals grows exponentially while the size of each interval shrinks, leading to finer quantization and higher sensitivity to small perturbations in a_{proto}.To mitigate this loss of precision and improve action resolution, the baseline further employs K-Nearest Neighbors (KNN) and dimension division techniques, which are described in the following subsections.

2.2.4 K-Nearest Neighbors (KNNs)

To improve the accuracy of the SAC continuous-to-discrete action mapping, the baseline solution [1] employs the KNN technique. Rather than selecting the single discrete action closest to the proto-action a_{proto}, the algorithm first identifies the K-nearest discrete-valued actions. These candidate actions are then evaluated by the critic network, which estimates their corresponding Q-values. Finally, the discrete action with the highest Q-value is selected. This technique helps reduce the quantization error by minimizing the sensitivity to small perturbations in a_{proto}. However, this added precision comes at the cost of increased computational complexity, since the critic must evaluate multiple candidate actions per dimension at each decision step.

2.2.5 Dimension division

To enhance the scheduler's scalability, i.e., to accommodate larger action spaces, the approach in [1] introduces a *dimension division* strategy, which leverages SAC's capability to handle high-dimensional tasks. Instead of outputting a single proto-action, the actor generates D proto-actions, where each is mapped to a separate discrete action space.

By partitioning the original discrete action space into D dimensions, the number of discrete actions per dimension is reduced from 2^L to $2^{L/D}$, thereby improving the mapping resolution from $\frac{2}{2^L}$ to $\frac{2}{2^{L/D}}$. This mitigates the risk of the required resolution falling below the network's output precision. As a result, the SAC-generated proto-actions can more effectively distinguish between discrete actions, reducing ambiguity and improving both decision accuracy and training stability in large-scale systems.

The baseline output discretization mechanism, along with the KNN technique and the dimension division strategy, is illustrated in Fig. 3.

3. PROPOSED SOLUTION

To enhance the performance of the baseline SMART scheduler, the proposed solution introduces two independent enhancement approaches.

Approach I proposes a redefined SAC DRL framework, which incorporates novel state and reward definitions and minor modifications to the baseline's discretization method. It also incorporates new neural network architecture designs. The proposed SAC DRL framework is presented in Section 3.1.1, and the proposed neural network architecture is described in Section 3.1.2.

Approach II introduces a novel user grouping technique, which leverages the ML-based HDBSCAN clustering algorithm. It also introduces targeted modifications to the SAC DRL framework, which incorporates a new state representation and modified action representation. This approach maintains the baseline's neural network architecture. The HDBSCAN-based user grouping methodology is detailed in Section 3.2.1, and the SAC DRL modifications with the new state representation are outlined in Section 3.2.2.

3.1 Approach I: Proposed SAC DRL framework and neural network architecture

3.1.1 Proposed SAC DRL framework

The proposed SAC DRL framework introduces three key modifications compared to the baseline scheduler: (i) a novel state representation, (ii) a refined action representation with an improved discretization strategy, and (iii) a redesigned reward formulation. These changes are designed to enhance scheduling performance while simultaneously reducing inference time.

3.1.1.1 **State space**

The proposed state at TTI t is defined as

$$\mathbf{s_t} = \left[\mathbf{U}(\mathbf{C_t}), \left\|\mathbf{h}_{1,t}\right\|_2, \ldots, \left\|\mathbf{h}_{L,t}\right\|_2\right], \qquad (13)$$

where $\mathbf{C}_t \in \mathbb{R}^{L\times L}$ is the channel correlation matrix, $\mathbf{U}(\mathbf{C}_t)$ denotes the set of non-diagonal elements of $\mathbf{U}(\mathbf{C}_t)$, i.e., $\mathbf{U}(\mathbf{C}_t) = \{c_{i,j,t} \mid i < j,\ 1 \le i, j \le L\}$, and $\|\mathbf{h}_{i,t}\|_2$ is the ℓ_2-norm of the channel vector of user i. The size of $\mathbf{U}(\mathbf{C}_t)$ is equal to $\frac{L(L-1)}{2}$, and the total state input size is equal to $\frac{L(L-1)}{2} + L$.

By comparing Eq. *(13)* with the baseline state in Eq. *(9)*, the following distinctions emerge:

(i) The proposed state eliminates user grouping and, instead, directly incorporates inter-user channel correlations by using $\mathbf{U}(\mathbf{C}_t)$. This design choice preserves complete channel correlation information while avoiding the computational overhead of processing the full channel correlation matrix.
(ii) The ℓ_2-norm of each user's channel vector is used instead of the maximum achievable user spectral efficiency $\gamma_{i,t}$. This reduces computational complexity while maintaining relevant information.
(iii) The proposed state omits the use of the user's data transmission history $f_{i,t}$ and instead leverages the stochastic nature of the SAC output to promote fairness among users, further reducing computational complexity.

As mentioned in Section 2.2.2, the baseline user grouping algorithm introduces information loss. To mitigate this limitation without resorting to the use of the full channel matrix, the proposed model redesigns its state space and incorporates the non-diagonal elements of the upper triangular channel correlation matrix in it. Since the correlation matrix is symmetric and the diagonal entries do not contain inter-user correlation information, only the coefficients $c_{i,j,t}$ for $i < j$, where $1 \le i, j \le L$, are included in the state space. This new state space captures the full inter-user channel correlation information included in $\mathbf{C}_t$, while reducing the state space input size by a factor of half compared to using the full C_t.

Another novel contribution is the elimination of the user transmission history from the state input and, instead, incorporating the stochasticity of the SAC output to incorporate fairness among the selected users. This is further explained in Section 3.1.1.3.

In addition, the proposed model replaces $\gamma_{i,t}$ with $\|\mathbf{h}_{i,t}\|_2$. Since channel strength and throughput are correlated, capturing the channel strength reduces the preprocessing required at the state while maintaining relevant information.

Although the proposed model has a larger state input size compared to the baseline, the computational overhead reduction and the information gain achieved by the proposed model design results in an overall model performance enhancement as shown in Section 4.4.

3.1.1.2 **Action space**

The proposed action at TTI t is defined as in the baseline formulation in Eq. *(10)*. Unlike the baseline model, the proposed action does not include a constraint on N, thus, $|\mathbf{a}_t| \le L$.

To discretize the output of the SAC DRL model and to allow for action space scalability, the proposed framework adopts the baseline's output discretization and dimension

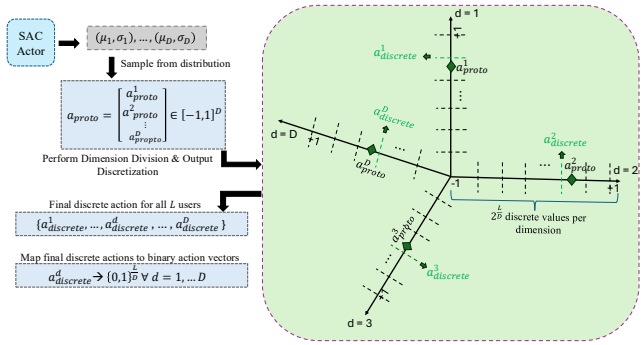

Figure 4 – Proposed discretized SAC architecture. Instead of using KNN with $K = 3$ as in the baseline, the proposed method selects only the nearest discrete action per dimension. This is enabled by increasing the proto-action dimensionality from $d = 8$ to $d = 32$, allowing for more precise and efficient action selection without KNN.

division methods. However, to reduce the computational complexity of the baseline scheduler while preserving the precision of the output, the proposed framework eliminates the baseline KNN technique and, instead, increases the number of discrete action space dimensions (D) from 8 to 32 and selects the nearest discrete value in each dimension. Consequently, the number of discrete actions in each dimension is reduced from $2^{\frac{L}{8}}$ to $2^{\frac{L}{32}}$, enlarging the spacing between discrete-valued actions and improving the accuracy of continuous-to-discrete action mapping while eliminating the computational overhead of the KNN technique. The proposed discretized strategy is illustrated in Fig. 4.

3.1.1.3 **Reward**

The reward at TTI t is defined as

$$r_t = \gamma_t^{\text{total}}, \tag{14}$$

where γ_t^{total} denotes the normalized sum rate of the scheduled users. Unlike the baseline model, the proposed reward does not include *JFI*. Instead, fairness is implicitly promoted through the stochastic nature of the SAC policy. In SAC, the proto-actions are sampled from a learned probability distribution; hence, users are randomly selected. Moreover, the entropy maximization in the SAC DRL model encourages continued exploration, leading to diverse user selections rather than repeatedly favoring a subset of users and thereby promoting more diverse user selection over time.

In summary, the proposed SAC DRL framework eliminates the computational overhead associated with user grouping and the KNN technique, reduces preprocessing requirements for the state input, and removes explicit *JFI* computation from the reward function. It further leverages the stochastic nature of SAC and enriches the state representation by incorporating full inter-user channel correlation information. Although this design increases the state input size compared to the baseline, it achieves improved overall performance in terms of sum rate, fairness and runtime, as validated in Section 4.4.

3.1.2 Proposed neural network architecture

The SAC DRL model consists of two main components: the policy network (actor) and the value network (critic). The actor learns a stochastic policy from which actions are sampled, while the critic evaluates the quality of selected actions by estimating their corresponding Q-values. These Q-value estimates play a critical role in guiding the actor toward high-reward actions during training. Consequently, the accuracy of the critic's Q-value predictions directly impacts the efficiency and stability of the learned policy.

In the proposed Approach I, we focused on redesigning the critic's network to learn more accurate Q-value estimates. Since these estimates are used in the actor's gradient update during training, inaccuracies may lead to suboptimal or unstable policy updates, which can further result in the selection of poor or suboptimal actions. Conversely, more accurate Q-value estimation leads to more reliable policy updates, enabling the actor to assign higher probabilities to genuinely high-value actions. This results in improved learning stability, faster convergence, and better overall performance of the SAC model.

In the baseline model, both the actor and critic are implemented using identical neural networks. The baseline's neural network is composed of two fully connected hidden layers, each with 256 neurons. In contrast, the proposed model adopts distinct neural network designs for the actor and the critic, as detailed below.

- **Actor**: As illustrated in Fig. 5, the actor network consists of two fully connected hidden layers, similar to the baseline design. However, since the proposed SAC framework employs a larger state input, the number of neurons in the first hidden layer is increased from 256 to 512 to support richer feature extraction. The second hidden layer remains unchanged with 256 neurons. The input layer has a size of $\frac{L(L-1)}{2} + L$, while the output layer has a size of L.

  In neural networks, there is a fundamental trade-off between information capacity and computational complexity. Increasing the number of neurons enhances the network's representation capacity, but also increases computational costs. Conversely, reducing the number of neurons decreases complexity at the expense of reduced capacity and limited learning ability. For the case of $L = 64$ users and a state input size of 2080, we doubled the number of neurons in the first hidden layer to achieve a more balanced design. However,

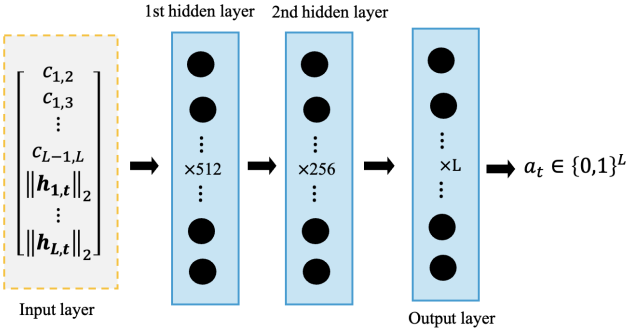

Figure 5 – Proposed actor neural network architecture.

tuning this parameter has not been fully explored and is left for future work. Although larger architectures may yield additional performance gains, they come with higher risks of overfitting and increased training overhead.

- **Critic**: As illustrated in Fig. 6, the critic processes the state and action inputs separately through dedicated input branches. The state input passes through two fully connected hidden layers with 512 and 256 neurons, respectively. The action input passes through a single hidden layer with 256 neurons. The outputs of both branches are then concatenated and passed through a common hidden layer with 256 neurons (referred to as the state-action hidden layer), which has a combined input size of 512. The output from the state-action hidden layer is further input into the output layer, which outputs the Q-value estimate of the state-action input pair.

  The separation of the state input and the action input in the critic allows it to perform different levels of feature extraction for each input type before combining them for Q-value estimation. This architectural separation enables the network to learn rich, task-specific representations of the state space independently from the action space. As a result, the critic can better model the interaction between state features and action choices, leading to more accurate Q-value estimates.

As the actor and critic input dimensions depend on the variable L, the networks are designed to accommodate a predefined maximum number of users per cell. When the number of users per cell is less than L, the inputs corresponding to inactive users can be muted by assigning negative placeholder values. The agent can then be trained on such data to learn an adaptive policy. This approach has not yet been evaluated and is left for future work.

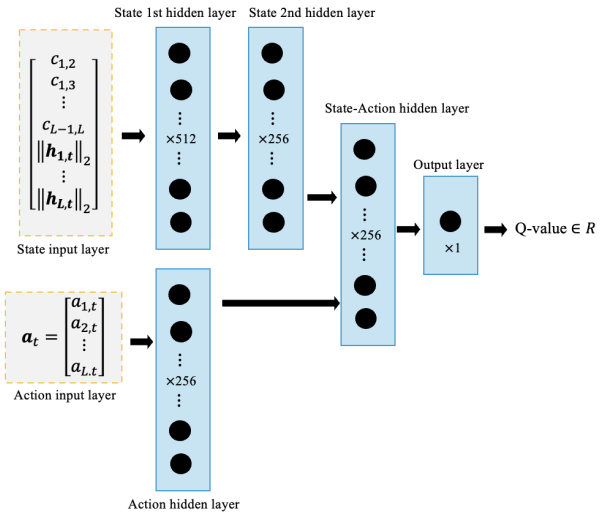

Figure 6 – Proposed critic neural network architecture.

3.2 Approach II: User grouping optimization

3.2.1 HDBSCAN-based user grouping

During the evaluation of the baseline model, it was observed that the user grouping algorithm represented a significant portion of the overall runtime , more specifically around 32% of the total step runtime. To improve the efficiency of the multiuser scheduler, we introduce a novel user grouping algorithm based on HDBSCAN. HDBSCAN is an unsupervised density-based clustering algorithm that automatically identifies clusters of varying densities without requiring the number of clusters to be specified [19]. It handles noise effectively by labeling outliers and is well-suited for high-dimensional or complex datasets where the underlying cluster structure is unclear.

For the HDBSCAN implementation, the function in [19, 20] is used. The function is fed with two inputs: the distance matrix D', representing the distance between each pair of data elements, and the minimum cluster size n, defining the minimum number of elements required to form a cluster. In this approach, n is set to 2 and D' to the inter-user channel correlation matrix C after setting its diagonal elements to 0. Since the inter-user channel correlation represents the distance, small $c_{i,j}$ would be interpreted as a small inter-element distance. However, this represents a logical distance between users and should not be confused with the physical inter-user distance. Therefore, logically, users with low correlation are considered close together, and those with high correlation are considered farther apart. The HDBSCAN function is then applied to the logical distance matrix D' to form clusters of uncorrelated users based on their pairwise correlation.

Unlike the baseline user grouping algorithm, HDBSCAN identifies natural groupings in the data by locating high

density regions and labeling users that do not belong to any group as noise. This leads to more coherent group formation and reduces the likelihood of assigning correlated users to clusters intended for uncorrelated ones. Furthermore, HDBSCAN enhances clustering efficiency by eliminating the need for manual correlation threshold tuning and exhaustive pairwise comparisons. Instead, it relies on density-based criteria to identify meaningful clusters, enabling the clustering process to scale more effectively with the number of users.

In terms of computational complexity, the baseline threshold-based user grouping requires approximately 0.01127 seconds per execution, whereas the HDBSCAN-based grouping requires only 0.0013 seconds. This corresponds to nearly a 9× reduction in runtime (about 88% faster). The improved efficiency arises from eliminating iterative pairwise searches and manual threshold checks, while simultaneously providing more coherent clusters.

3.2.2 SAC design with HDBSCAN grouping

To improve the convergence and performance of the SAC model, the following modifications were made to the baseline SAC model.

- **State space**: Compared to the baseline, a new state representation is defined. This state incorporates standardized features to improve learning stability. Similar to Approach I, it uses the L2 norm of the user channel vector $\left\|\mathbf{h}_{i,t}\right\|_2$ instead of the maximum achievable user spectral efficiency $\gamma_{i,t}$. The state $s_{i,t}$ for user i at TTI t is defined as

$$s_{i,t} =< g_{i,t}, \tilde{h}_{i,t}, \tilde{f}_{i,t} >\in \mathcal{S} := [\mathcal{G}, \mathcal{H}, \mathcal{F}], \quad (15)$$

  where $g_{i,t}$ represents the user group ID computed by the HDBSCAN function, $\tilde{h}_{i,t}$ represents the standard L2 norm of the i^{th} user channel vector, and $\tilde{f}_{i,t}$ represents the standard user's data transmission history $f_{i,t}$.

  $\tilde{h}_{i,t}$ is computed as

$$\tilde{h}_{i,t} = \frac{\left\|\mathbf{h}_{i,t}\right\|_2 - \mu_h}{\sigma_h + 1e\text{-}8}, \quad (16)$$

  where μ_h and σ_h are the mean and standard deviation of $\left\|\mathbf{h}_{i,t}\right\|_2$ over all users, and $1e$-8 is added for numerical stability.

  Similarly, $\tilde{f}_{i,t}$ is computed as

$$\tilde{f}_{i,t} = \frac{f_{i,t} - \mu_f}{\sigma_f + 1e\text{-}8}, \quad (17)$$

  where μ_f and σ_f are the mean and standard deviation of $f_{i,t}$ over all users.

  The SAC's state input at time t is defined as

$$\mathbf{s_t} =< \mathbf{g_t}, \tilde{\mathbf{h}}_\mathbf{t}, \tilde{\mathbf{f}}_\mathbf{t} >\in \mathcal{S}, \quad (18)$$

  where $\mathbf{g_t} = [g_{1,t}, \ldots, g_{L,t}]$, $\tilde{\mathbf{h}}_\mathbf{t} = [\tilde{h}_{1,t}, \ldots, \tilde{h}_{L,t}]$, and $\tilde{\mathbf{f}}_\mathbf{t} = [\tilde{f}_{1,t}, \ldots, \tilde{f}_{L,t}]$. The dimensionality of $\mathbf{s_t}$ is equal to $3 \times L$, which is unchanged compared to the baseline model.

- **Action space**: The action at TTI t is defined in Eq. *(10)*. However, the constraint over the maximum number of scheduled users N_{max} is omitted, i.e. $|\mathbf{a}_t| \leq L$.
  To convert the continuous SAC output into a binary action vector, Approach II adopts the baseline's output discretization and dimension division methods. However, similar to Approach I, the KNN step is omitted. Additionally, the number of proto-actions (or action dimensions) is increased from 8 to 16. As a result, the number of discrete actions per dimension is reduced from $2^{\frac{L}{8}}$ to $2^{\frac{L}{16}}$.

- **Reward**: The baseline's reward presented in Eq. *(11)* is adopted.

  Similar to the baseline, this approach employs $f_{i,t}$ in its state representation. However, practical Proportional-Fair (PF) schedulers typically compute throughput using a finite moving window in order to capture recent service conditions and short-term fairness. In contrast, $f_{i,t}$ grows monotonically with time and becomes increasingly insensitive to recent performance fluctuations. This reduces the ability of the SAC agent to accurately capture recent scheduling dynamics, as the input feature becomes dominated by long-term history rather than meaningful short-term information. We acknowledge this limitation in the current model design and note that extending the state to include a windowed throughput history measure is a promising direction for future work

4. EVALUATION

4.1 Simulation setup

To evaluate the performance of the proposed solution, challenge participants received two channel datasets representing low-mobility and stationary scenarios, along with the baseline solution code [24]. Both datasets are based on a single 64-antenna massive MIMO BS serving 64 single-antenna users and a channel bandwidth of 52 frequency subcarriers. Each dataset contains 500 channel realizations, where each realization represents the uplink channel between the 64 users and the 64 BS antennas across all 52 subcarriers. Further details on the datasets and baseline implementation can be found in [24]. The SAC model parameters of the proposed approaches are presented in Table 1, and the simulation parameters are summarized in Table 2.

In the baseline simulation, each frequency subcarrier is treated as a PRB, and user-scheduling decisions are performed independently per PRB. The same method-

Table 1 – SAC model parameters for proposed approaches.

Parameter	Approach I	Approach II
User Grouping	None	HDBSCAN
State	$[\mathbf{U}(\mathbf{C_t}), \left\|\mathbf{h}_{1,t}\right\|_2, \ldots, \left\|\mathbf{h}_{L,t}\right\|_2]$	$< \mathbf{g}_t, \tilde{\mathbf{h}}_t, \tilde{\mathbf{f}}_t >$
State Dimension	2080	192
Final Action	$a_t \in [0,1]^{64}$	$a_t \in [0,1]^{64}$
Final Action dimension	64	64
No. of Proto-Actions	32	16
No. of Discrete Actions per Dimension	2^2	2^4
Reward	γ_t^{total}	$(\beta)\gamma_t^{\text{total}} + (1-\beta)JFI_t) * c$
Reward Control Parameter β	None	0.5
Reward Scaling Parameter c	None	2

ology was adopted in our approach. Additionally, the baseline checkpoint file was provided to facilitate performance comparisons and final score computation. The results and the final score calculation of our proposed approaches are presented in Section 4.4.

4.2 Model training

In the baseline code, the 21st subcarrier of the low-mobility dataset is used for training. However, using only one subcarrier in training may lead to overfitting and reduced generalization to other subcarriers and different mobility scenarios. To enhance the model's generalization capability, we adopt an improved training strategy.

Specifically, instead of training on a single subcarrier from one scenario, we utilize the first 40 subcarriers from both the low-mobility and stationary datasets, resulting in 80 subcarriers in total. During each training episode, one of the two datasets is randomly selected with a probability of 0.8 for the low-mobility dataset and 0.2 for the stationary dataset. Subsequently, one subcarrier from the selected dataset is chosen uniformly at random.

Biasing the sampling towards the low-mobility scenario ensures that the model is trained on a larger proportion of samples with faster channel variations. Since the low-mobility dataset exhibits more rapid temporal fluctuations compared to the stationary case, frequent exposure to such conditions enables the model to learn a more adaptive and generalizable scheduling policy. In addition, training on multiple subcarriers exposes the model to frequency-selective fading effects, thereby improving robustness across the frequency domain.

However, it is worth noting that the velocity difference between the low-mobility and stationary datasets is relatively small. Consequently, when the sampling probability was set equally between the two datasets, only minor performance differences were observed. Table 3 compares the sum rate and fairness achieved by Approach II when the scenario sampling probability, P_{sample}, is set to 0.5 and 0.8. It can be noted that both probabilities achieve relative equal fairness in both scenarios; however, $P_{sample} = 0.8$ achieves a relatively small improvement in terms of the sum rate. Due to the limited scope of the available datasets, a more extensive exploration of this sampling strategy and its hyperparameter was not possible and is left for future work.

Table 2 – Simulation parameters for proposed approaches.

Parameter	Approach I	Approach II
No. of BS antennas	64	
No. of UEs	64	
Carrier Frequency	3.6 GHz	
UE Speed	0 and 2.8 m/s	
TTI Duration	1 ms	
Modulation	16 QAM	
Training Episodes	1400	
TTIs per Episode	500	
Alpha Learning Rate	3×10^{-5}	
Actor Learning Rate	3×10^{-5}	
Critic Learning Rate	3×10^{-5}	
Automatic Entropy Tuning	True	
Optimizer	Adam	
Batch Size	256	
Replay Buffer Size	10^6	
Scenario Sampling Probability	0.8	

4.3 Evaluation framework

The performance of the trained models is assessed using the official challenge evaluation framework [21], [24]. In this framework, each model is evaluated over 200 steps or TTIs using the following performance metrics.

- **Sum rate (R):** The average sum rate achieved over 200 *TTIs*. R is defined as

$$R = \frac{1}{200} \sum_{t=1}^{200} R_t, \tag{19}$$

  where R_t denotes the sum rate achieved by the N scheduled users at TTI t.

- **Fairness (F):** The user fairness, measured using *JFI* at TTI 200. F is defined as

$$F = \frac{\left(\sum_{i=1}^{L} f_{i,200}\right)^2}{L \sum_{i=1}^{L} \left(f_{i,200}\right)^2}, \tag{20}$$

Table 3 – Approach II performance relative to different scenario sampling probabilities P_{sample}.

P_{sample}	Low-mobility Scenario		Stationary Scenario	
	Sum Rate	Fairness	Sum Rate	Fairness
0.5	1971.74	0.952	1247.99	0.956
0.8	1984.13	0.949	1256.56	0.952

where $f_{i,200}$ is the total data transmitted by user i up to TTI 200.

- **Runtime (T):** The average inference time of the actor, measured in seconds. It represents the mean time required for the actor to generate an action during a single inference step.

These metrics are further combined to compute the evaluation score set by the ITU challenge [21], [24] for low-mobility and stationary scenarios. The score for the low-mobility scenario is calculated as

$$S_{low} = 0.4\left(\frac{R_{s,low}}{R_{b,low}}\right) + 0.2\left(\frac{F_{s,low}}{F_{b,low}}\right) + 0.4\left(\frac{T_{b,low}}{T_{s,low}}\right), \qquad (21)$$

where the metric subscript s refers to the submitted model and b to the baseline model. That is, $R_{s,low}$ denotes the sum rate achieved by the proposed model and $R_{b,low}$ refers to the sum rate of the baseline model, all achieved in a low-mobility scenario.

Similarly, the score for the stationary scenario is calculated as

$$S_{stat} = 0.4\left(\frac{R_{s,stat}}{R_{b,stat}}\right) + 0.2\left(\frac{F_{s,stat}}{F_{b,stat}}\right) + 0.4\left(\frac{T_{b,stat}}{T_{s,stat}}\right). \qquad (22)$$

The final evaluation score presented in Table 4 is calculated as

$$Score = \frac{S_{low} + S_{stat}}{2}. \qquad (23)$$

The weights of the evaluation metrics (0.4, 0.2, 0.4) in *(21)* and *(22)* are set by the challenge organizers and are not part of the solution design. Since the sum rate and runtime components are given a higher weight (0.4) compared to the fairness component (0.2), the competition places greater emphasis on the scheduler's sum rate and runtime performance than on its fairness.

4.4 Results

Table 4 presents the evaluation results of the baseline scheduler and the two proposed approaches for the low-mobility and stationary scenarios. In both scenarios, the proposed methods outperform the baseline across all three metrics: sum rate, fairness, and runtime. Approach I achieves the highest improvements in sum rate and fairness, while Approach II delivers the most significant runtime reduction.

Table 4 – Evaluation results of proposed models.

Model	Low-mobility Scenario			Stationary Scenario			Score
	R	F	T	R	F	T	
Baseline	1907.82	0.8382	0.00908	1218.64	0.8688	0.00949	1.00
Approach I	2128.75	0.9807	0.00394	1289.52	0.9779	0.00306	1.74
Approach II	1984.12	0.9491	0.00131	1256.56	0.9520	0.00137	3.41

In the **low-mobility scenario**, Approach I increases the baseline sum rate by 11.6%, compared to a 4.0% improvement achieved by Approach II. In terms of fairness, Approach I achieves near-perfect fairness with 17.0% gain, while Approach II achieves a 13.2% improvement relative to the baseline.

In the **stationary scenario**, Approach I achieves a 5.8% increase in the sum rate compared to the baseline, while Approach II provides a smaller gain of 3.1%. For fairness, Approach I yields a 12.6% improvement, while Approach II achieves a 9.6% gain.

In both scenarios, Approach II achieves the most significant reduction in runtime, with an average improvement of approximately 85% compared to the baseline, while Approach I achieves an average runtime reduction of around 62%.

From the above results, it can be seen that both approaches yield larger improvements in the low-mobility scenario compared to the stationary scenario. This highlights the ability of the proposed approaches to better adapt to varying channels compared to the baseline model.

Finally, the evaluation score, which aggregates throughput, fairness, and runtime, highlights the overall advantage of the proposed approaches. Approach I achieves a score of 1.74, corresponding to a 74% improvement over the baseline. Approach II achieves the highest score of 3.41, more than tripling the baseline. This substantial gain is primarily driven by its drastic runtime reduction, while still maintaining competitive performance in sum rate and fairness.

These results reveal a clear trade-off between throughput and fairness maximization versus computational efficiency. Approach I emphasizes higher sum rate and fairness but with a modest runtime improvement compared to the baseline. In contrast, Approach II achieves the best overall score by prioritizing runtime efficiency while maintaining competitive throughput and fairness.

5. CONCLUSION

In this paper, we presented the winning solution to the ITU AI/ML in 5G Challenge, which focused on enhancing an ML-based multiuser massive MIMO scheduler. We proposed two approaches for improving the baseline scheduler's performance. Both approaches outperformed the baseline solution in terms of sum rate, fairness, and runtime. In Approach I, we demonstrated how the stochastic nature of the SAC output can be leveraged to promote user fairness, representing a novel contribution to DRL-based schedulers. We also emphasized the importance of balancing the SAC model's complexity with the richness of the input state information. Although the state space in Approach I was larger than in the baseline, the omission of user grouping and output KNN, combined with an increase in the information provided to the SAC model and optimizations in neural network design, led to improved scheduler performance. In Approach II, we showcased the effectiveness of applying ML-based clustering techniques, specifically HDBSCAN, for user grouping. By improving both the accuracy and the efficiency of the clustering process and standardizing the state input, Approach II outperformed the baseline scheduler by more than threefold. The combination of using HDBSCAN-based user grouping and leveraging the stochastic nature of SAC output for fairness, i.e., omitting explicit fairness criteria from the reward function, was not explored in this study and is left as an avenue for future work.

REFERENCES

[1] Qing An, Santiago Segarra, Chris Dick, Ashutosh Sabharwal, and Rahman Doost-Mohammady. "A Deep Reinforcement Learning-Based Resource Scheduler for Massive MIMO Networks". In: *IEEE Transactions on Machine Learning in Communications and Networking* 1 (2023), pp. 242–257. ISSN: 2831-316X. DOI: 10.1109/tmlcn.2023.3313988. URL: http://dx.doi.org/10.1109/TMLCN.2023.3313988.

[2] Chia-Mu Chen, Qian Wang, Abdelrahman Gaber, Andrea P. Guevara, and Sofie Pollin. "User Scheduling and Antenna Topology in Dense Massive MIMO Networks: An Experimental Study". In: *IEEE Transactions on Wireless Communications* 19.9 (2020), pp. 6210–6223. DOI: 10.1109/TWC.2020.2994760.

[3] Hongyang Liu, Hui Gao, Sheng Yang, and Tao Lv. "Low-Complexity Downlink User Selection for Massive MIMO Systems". In: *IEEE Systems Journal* 11.2 (2017), pp. 1072–1083. DOI: 10.1109/JSYST.2015.2479585.

[4] Yuchen Chen, Yingying Wu, Y. Thomas Hou, and Wenjing Lou. "mCore: Achieving Submillisecond Scheduling for 5G MU-MIMO Systems". In: *IEEE INFOCOM 2021 - IEEE Conference on Computer Communications*. 2021, pp. 1–10. DOI: 10.1109/INFOCOM42981.2021.9488725.

[5] Xiaojun Guo, Ziyi Li, Pengyu Liu, Rudan Yan, Yingying Han, Xiaojun Hei, and Guohui Zhong. "A Novel User Selection Massive MIMO Scheduling Algorithm via Real Time DDPG". In: *2020 IEEE Global Communications Conference (GLOBECOM)*. 2020, pp. 1–6. DOI: 10.1109/GLOBECOM42002.2020.9347805.

[6] Hongchao Chen, Yupu Liu, Zhe Zheng, Huiyang Wang, Xiaowei Liang, Yuhang Zhao, and Jian Ren. "Joint User Scheduling and Transmit Precoder Selection Based on DDPG for Uplink Multi-User MIMO Systems". In: *2021 IEEE 94th Vehicular Technology Conference (VTC2021-Fall)*. 2021, pp. 1–5. DOI: 10.1109/VTC2021-Fall52928.2021.9625344.

[7] Junchao Shi, Wenjin Wang, Jiaheng Wang, and Xiqi Gao. "Machine Learning Assisted User-Scheduling Method for Massive MIMO System". In: *2018 10th International Conference on Wireless Communications and Signal Processing (WCSP)*. 2018, pp. 1–6. DOI: 10.1109/WCSP.2018.8555722.

[8] X. Yu, Z. Zhang, and H. Wang. "Deep learning based user scheduling for massive MIMO downlink system". In: *Science China Information Sciences* (2021). [Online]. Available: http://scis.scichina.com/en/2021/182304.pdf.

[9] Q. An, B. Wang, and W. Wang. "A Deep Reinforcement Learning-Based Resource Scheduler for Massive MIMO Networks". In: *IEEE Transactions on Mobile Computing* (Mar. 2023). [Online]. Available: https://arxiv.org/abs/2303.00958.

[10] Jianqing Fan, Zhaoran Wang, Yuchen Xie, and Zhuoran Yang. "A theoretical analysis of deep Q-learning". In: *Learning for dynamics and control*. PMLR. 2020, pp. 486–489.

[11] Hado Van Hasselt, Arthur Guez, and David Silver. "Deep reinforcement learning with double q-learning". In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 30. 1. 2016.

[12] Mohammad Babaeizadeh, Iuri Frosio, Stephen Tyree, Jason Clemons, and Jan Kautz. "Reinforcement learning through asynchronous advantage actor-critic on a gpu". In: *arXiv preprint arXiv:1611.06256* (2016).

[13] Mohit Sewak and Mohit Sewak. "Actor-critic models and the A3C: The asynchronous advantage actor-critic model". In: *Deep reinforcement learning: frontiers of artificial intelligence* (2019), pp. 141–152.

[14] Ziyu Wang, Victor Bapst, Nicolas Heess, Volodymyr Mnih, Remi Munos, Koray Kavukcuoglu, and Nando De Freitas. "Sample efficient actor-critic with experience replay". In: *arXiv preprint arXiv:1611.01224* (2016).

[15] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. "Proximal policy optimization algorithms". In: *arXiv preprint arXiv:1707.06347* (2017).

[16] Chengrun Qiu, Yang Hu, Yan Chen, and Bing Zeng. "Deep deterministic policy gradient (DDPG)-based energy harvesting wireless communications". In: *IEEE Internet of Things Journal* 6.5 (2019), pp. 8577–8588.

[17] Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, et al. "Soft actor-critic algorithms and applications". In: *arXiv preprint arXiv:1812.05905* (2018).

[18] Pádraig Cunningham and Sarah Jane Delany. "k-Nearest Neighbour Classifiers - A Tutorial". In: *ACM Computing Surveys* 54.6 (July 2021), pp. 1–25. ISSN: 1557-7341. DOI: 10.1145/3459665. URL: http://dx.doi.org/10.1145/3459665.

[19] Leland McInnes, John Healy, and Steve Astels. "hdbscan: Hierarchical density based clustering". In: *The Journal of Open Source Software* 2.11 (2017), p. 205.

[20] Leland McInnes and John Healy. "Accelerated Hierarchical Density Based Clustering". In: *Data Mining Workshops (ICDMW), 2017 IEEE International Conference on*. IEEE. 2017, pp. 33–42.

[21] International Telecommunication Union. *Match Item 93 – ITU AI/ML in 5G Challenge*. https://challenge.aiforgood.itu.int/match/matchitem/93. Accessed: 2025-09-29. 2024.

[22] International Telecommunication Union. *The 5th edition of the ITU AI/ML in 5G Challenge: A year of competitions in review*. https://aiforgood.itu.int/the-5th-edition-of-the-itu-ai-ml-in-5g-challenge-a-year-of-competitions-in-review/. 2024.

[23] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. *Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor*. 2018. arXiv: 1801.01290 [cs.LG]. URL: https://arxiv.org/abs/1801.01290.

[24] 3DML-Wireless. *Smart-SCHEDULER/ML-Challenge/ML_Challenge_Code at master · 3DML-wireless/smart-scheduler*. https://github.com/3DML-Wireless/SMART-Scheduler/tree/master/ML-Challenge/ML_Challenge_Code.

AUTHORS

Sara Al-kokhon is currently pursing her Ph.D. in Electrical and Computer Engineering at the University of Toronto. She is also a Cloud Architect at Bell Canada. She completed her M.A.Sc. in Electrical and Computer Engineering from the University of Toronto in 2017, and joined Bell as a Technology Specialist in 2018. Her M.A.Sc. research focused on enhancing channel estimation in OFDM-based MIMO systems, while her current research focuses on the application of Deep Reinforcement Learning (DRL) in mobile wireless networks. Her industry experience spans multiple domains, including network virtualization, transport network design, cloud architecture, open-source RAN, and end-to-end network observability. She received the IEEE Best Student Paper Award in 2013 for a paper published in the IEEE Standards Education e-Magazine, vol. 3, no. 1 (1st Quarter). Her research interests include multi-agent DRL, resource allocation in wireless networks, and autonomous networks.

Hossein Bijanrostami received his B.Sc. and M.Sc. in Electrical Engineering (communications) from the University of Tehran in 2017 and 2020, respectively. He is currently pursuing a Ph.D. degree in Electrical and Computer Engineering at the University of Toronto. His research interests include wireless communications and networking, semantic communications, mobile edge computing, and wireless artificial intelligence.

Elaheh Bassak received her B.Sc. in Electrical Engineering from Alzahra University, Tehran, Iran, in 2020, and her M.Sc. in Electrical Engineering from Sharif University of Technology, Tehran, Iran, in 2022. During her master's studies, she worked on sparse signal processing and published her work in the IEEE Transactions on Signal Processing. She is currently pursuing a Ph.D. degree in Electrical and Computer Engineering at the University of Toronto. Her current research interests include the application of machine learning in communications networks.

Brad Stimpson is the Director, Wireless Technology at Bell Canada, specializing in radio innovation and network transformation. With 20+ years of experience in wireless technology, Brad has a proven track record of driving business growth, leading successful product launches, managing large-scale projects, and team transformation. Before joining Bell, Brad served as VP of Engineering at Redline Communications, where he led RD, Product Line Management, and System Engineering. Brad has vast experience across the entire technology supply chain having worked for Academia, MNOs, OEMs, Software Developers, and Consultancies. Brad holds a M.Eng. in Electrical Engineering from McGill University in 2009 and he is a licensed professional engineer. He is passionate about radio and network innovation and regularly participates with the Open Compute Project as the co-chair of OpenRU strategic initiatives.

Elvino Sousa received his B.A.Sc. in engineering science, and the M.A.Sc. in Electrical Engineering from the University of Toronto in 1980 and 1982 respectively, and his Ph.D. in electrical engineering from the University of Southern California in 1985. Since 1986 he has been with the department of Electrical and Computer Engineering at the University of Toronto. He has performed research in cellular wireless systems since 1983. His current interests are in the areas of broadband wireless systems, smart antenna systems, autonomous infrastructure wireless networks, self-configurable wireless networks, user deployed networks, adaptive unicast/broadcast networks, and the use of AI techniques in the optimization of wireless networks. He was the founder of wireless communications at the University of Toronto and is the director of the wireless lab, which has undertaken research in wireless systems for the past 35 years. He has been invited to give lectures and short courses on spread spectrum, CDMA, and wireless systems in many countries, and has been a consultant to industry and Governments internationally in the area of wireless systems. He was the technical program chair for PIMRC 95, vice-technical program chair for Globecom '99, Co-Technical Program Chair for WPMC 2010 and for PIMRC 2011, and has been involved in the technical program committee of numerous international conferences. He is a past chair of the IEEE Technical committee on Personal Communications. He has spent sabbatical leaves at Qualcomm and Sony CSL/ATL, where he was the holder of the Sony sabbatical chair. He is a Fellow of the IEEE and has been awarded the Queen Elizabeth II Golden Jubilee Medal.