ANN-Based Downlink Resource Allocation for 5G MIMO Networks

Chrysostomos-Athanasios Katsigiannis
Computer Engineering and Informatics Department
University of Patras
Patras, Greece
Email: up1072490@upnet.gr

Konstantinos Tsachrelias
Computer Engineering and Informatics Department
University of Patras
Patras, Greece
Email: up1096511@upatras.gr

Vasileios Kokkinos
Computer Engineering and Informatics Department
University of Patras
Patras, Greece
Email: kokkinos@cti.gr

Apostolos Gkamas
Department of Chemistry
University of Ioannina
Ioannina, Greece
Email: gkamas@uoi.gr

Christos Bouras
Computer Engineering and Informatics Department
University of Patras
Patras, Greece
Email: bouras@upatras.gr

Philippos Pouyioutas
Computer Science Department
University of Nicosia
Nicosia, Cyprus
Email: pouyioutas.p@unic.ac.cy

***Abstract*- This paper investigates a supervised Artificial Neural Network (ANN)-based approach for downlink resource allocation in a simplified single-cell 5G MIMO network with one central gNodeB. The MIMO physical layer is represented through the effective user-side Signal-to-Noise Ratio (SNR), which captures channel quality, propagation conditions, and link-level processing without explicitly modeling antenna arrays, beamforming, or precoding. The ANN uses traffic demand, effective SNR, and Quality-of-Service (QoS) requirements to predict one of four allocation-priority classes, which are mapped to normalized allocation factors affecting user throughput. The proposed scheduler is compared with Random, Round-Robin, and Max-SNR methods under multiple synthetic traffic and channel scenarios. Performance is evaluated in terms of throughput, delay, QoS satisfaction, and Jain's fairness index. Results show that the ANN-based scheduler improves throughput, delay, and QoS satisfaction while maintaining competitive fairness, supporting more adaptive downlink scheduling in controlled 5G MIMO environments.**

***Keywords*- *5G Networks, Multiple Input Multiple Output (MIMO),* Artificial Neural Network, Resource Allocation.**

I. INTRODUCTION

Machine learning has been increasingly considered for radio resource management, since scheduling decisions depend on traffic conditions, channel quality, and heterogeneous QoS requirements [1], [2]. Conventional scheduling methods, including Round-Robin and channel-aware approaches, remain useful reference strategies, although their limitations become more evident under heterogeneous 5G traffic conditions [3]. Massive MIMO introduces additional resource-management challenges due to the larger number of users and the strong dependence of scheduling decisions on channel conditions [4]. More recent studies have investigated learning-based scheduling in 5G NR environments, including DQN- and PPO-based methods for heterogeneous eMBB traffic [5]. Deep learning has also been studied across several 5G communication and resource-management problems [6], while ANN-based methods have been applied to wireless network optimization and resource allocation [7]. Recent work has further addressed explicit 5G resource-block scheduling using deep reinforcement learning and transformer-based processing [8]. In contrast, the present work considers a lightweight supervised MLP that maps traffic demand, effective SNR, and QoS information into four allocation-priority classes in a controlled single-cell environment, without explicit PRB allocation, multi-cell interference, or online reinforcement learning.

This paper evaluates a supervised ANN-based downlink allocation-priority mechanism in a simplified single-cell 5G MIMO scenario, where physical-layer effects are represented through effective user-side SNR values. The simulation pipeline includes synthetic traffic and channel generation, ANN training, and comparison with Random, Round-Robin, and Max-SNR schedulers. Rather than proposing a new neural architecture, the study examines how traffic demand, SNR, and QoS conditions can be mapped to discrete allocation levels. Performance is evaluated using throughput, delay, QoS satisfaction, and fairness. A brief sensitivity discussion also examines the effect of selected hyperparameters, including learning rate, training epochs, and batch size, on training stability and classification performance. The results show that the ANN-based scheduler generally outperforms the selected baselines while highlighting the importance of appropriate hyperparameter configuration.

The remainder of this paper is organized as follows: Section II presents the system model and formulates the problem addressed in this study. Section III describes the proposed methodology and algorithmic framework. Section IV details the simulation setup and the performance evaluation metrics. Section V presents and discusses the simulation results. Finally, Section VI concludes the paper and outlines directions for future work.

979-8-3195-1282-6/26/$31.00 ©2026 IEEE

II. Mathematical Model

The implemented resource allocation problem is formulated for a controlled and simplified single-cell downlink 5G MIMO scenario. Let $U = \{u_1, u_2, \dots, u_n\}$ denote the set of users served by one central gNodeB. Since only one serving gNodeB is considered, the problem does not involve multi-BS association. In this model, MIMO-related physical-layer effects are not explicitly represented through antenna arrays, channel matrices, beamforming vectors, or precoding schemes. Instead, they are abstracted through the effective SNR observed by each user, which serves as a compact representation of link quality after propagation and link-level processing. The objective is to assign each user an allocation-priority class that improves throughput and QoS satisfaction while controlling delay and fairness [9].

Each user u_i is described by a traffic demand T_i, an effective SNR value SNR_i, and a QoS requirement Q_i. In the simulation, T_i is interpreted as the generated traffic demand in Mbits for the considered scheduling interval. These values form the ANN input vector $X_i = [T_i, SNR_i, Q_i]$. The decision variable is the predicted allocation-priority class $c_i \in \{0, 1, 2, 3\}$, where each class corresponds to a normalized resource allocation level within the serving cell.

$$c_i \in \{0, 1, 2, 3\}, \forall i \tag{1}$$

where each class corresponds to a normalized resource allocation level within the serving cell. The four classes are mapped to normalized allocation factors $w^c = \{0.25, 0.50, 0.75, 1.00\}$. Therefore, the effective throughput of user u_i is computed as:

$$R_i = \frac{B \cdot w_i^c \log_2(1 + \gamma_i)}{10^6} \tag{2}$$

where R_i denotes the estimated user throughput in Mbps, B is the reference system bandwidth in Hz, w_i^c is the allocation factor associated with the predicted allocation-priority class c, and γ_i is the effective SNR in linear scale. In this simulation, a reference bandwidth of $B = 1$MHz is assumed. Therefore, the numerical throughput values are equivalent to $w_i^c \log_2(1 + \gamma_i)$ and are reported in Mbps.

The total system throughput is then obtained as the sum of the effective throughput values of all users:

$$T_{\text{total}} = \sum_{i=1}^{N} R_i \tag{3}$$

Since the implemented setup contains only one serving gNodeB, base-station association is not part of the optimization problem. All users are assumed to be connected to the same central gNodeB, and the allocation process focuses only on predicting the allocation-priority class for each user within that cell.

$$c_i \in \{0,1,2,3\}, \quad \forall u_i \in U \tag{4}$$

In the implemented simulation, the allocation class does not represent an explicit physical resource block assignment. Instead, it controls the normalized throughput factor applied to each user. Therefore, feasibility is represented through the bounded allocation factors $0.25 \le w^c \le 1.00$.

Fairness is evaluated using Jain's fairness index, computed over the achieved user throughput values as:

$$J = (\textstyle\sum_i R_i)^2 / (N \sum_i R_i^2). \tag{5}$$

where higher values indicate a more balanced throughput distribution among users.

Delay is computed from the ratio between traffic demand and achieved throughput as

$$D_i = T_i / (R_i + \varepsilon) \tag{6}$$

where D_i denotes the estimated delay of user u_i in seconds, T_i is the generated traffic demand in Mbits, R_i is the estimated throughput in Mbps, and ϵ is a small constant used to avoid division by zero. The present evaluation focuses on throughput, delay, QoS satisfaction, and Jain's fairness index, which are the metrics directly computed in the simulation. The ANN-based allocation mechanism is implemented as a supervised Multilayer Perceptron (MLP) classifier. The input vector is $X_i = [T_i, SNR_i, Q_i]$, where T_i is the traffic demand, SNR_i is the signal-to-noise ratio, and Q_i is the QoS requirement of user u_i. The model consists of three hidden layers with [128, 64, 32] neurons using ReLU activation, while the softmax output layer predicts one of four allocation-priority classes $c_i \in \{0, 1, 2, 3\}$. The model is trained using Sparse Categorical Crossentropy and the Adam optimizer with a learning rate of 0.0005. For comparison, the ANN-based scheduler is evaluated against Random, Round-Robin, and Max-SNR allocation methods. The proposed model is evaluated under three controlled synthetic scenarios. Scenario A uses exponential traffic, normally distributed SNR, and uniformly distributed QoS values. Scenario B uses Weibull-distributed traffic, lognormal SNR, and exponential QoS values. Scenario C uses a linear traffic trend with Gaussian noise, sinusoidal SNR variation, and uniformly distributed QoS values. These scenarios are used to evaluate the ANN under different traffic and channel conditions, without claiming to reproduce a complete 3GPP channel model [10]. This design follows the broader practice of using representative 5G traffic profiles and learning-based scheduling evaluations [11], [12].

III. Algorithm Analysis

The proposed Algorithm 1 leverages a trained Multilayer Perceptron (MLP) to map each user's downlink context—traffic demand, effective SNR, and QoS requirement—to one of four allocation-priority classes. The input vector is $X_i = [T_i, SNR_i, Q_i]$, and the features are standardized using the same scaler applied during training. Since a single-gNB setup is considered, the predicted class does not represent base-station association but the user's resource-allocation level within the serving cell. Once the inputs are preprocessed, MLP inference is executed per user, producing a class label that represents the user's allocation-priority level within the serving cell. The predicted class is then mapped to an allocation factor used in the throughput calculation.

The MLP output is a probability distribution over four classes {0, 1, 2, 3}. These classes are mapped to normalized allocation factors $w^c = \{0.25, 0.50, 0.75, 1.00\}$, respectively, and the effective throughput is computed as $R_i = w^c \log_2(1 + SNR_i)$. When the resource allocation for each user is determined, the algorithm checks the sum of all assigned fractions to ensure it does not exceed the overall system budget. If it does, a proportional scaling is performed across all allocations so that the feasible constraint is met while preserving the relative ratios assigned by the neural network.

Algorithm 1. Proposed ANN-Based Resource Allocation Algorithm.

1. **Step 1: Initial Setup:**
   The algorithm reads the global configuration, defining the number of users, simulation length, cell radius, and traffic/channel parameters. These values are stored in memory, and each user is assigned to an initial location within the single-cell area, establishing the network's starting state.
2. **Step 2: Topology and Data Structures:**
   It builds the internal single-cell topology by calculating each user's distance from the central gNodeB and the corresponding SNR value. Arrays or tables are created to store time-varying metrics such as throughput, delay, and fairness, updated after each scheduling round.
3. Step 3: Feature Collection and Normalization:
   The algorithm gathers each user's signal quality, traffic demand, and QoS indicators, forming an input vector per user. Features are normalized using the same strategy as during neural network training to ensure consistency.
4. **Step 4: Neural Network Inference:**
   The trained neural network processes the normalized features for each user, outputting one of four allocation-priority classes. This class determines the allocation factor used in the throughput calculation.
5. **Step 5: Resource Adjustment:**
   The predicted class is mapped to the corresponding normalized allocation factor used in the throughput calculation.
6. **Step 6: Performance Calculation:**
   Throughput and delay are derived from signal quality and allocated resources, then used to update metrics on fairness, QoS satisfaction, and overall efficiency.
7. **Step 7: Iteration and Logging:**
   Finally, the algorithm advances the simulation interval, updates the traffic and QoS values, and repeats inference for the next round until completion.

Over successive scheduling intervals, this procedure adapts to new network states. The user SNR or traffic demands may fluctuate due to mobility, and the updated feature vectors are repeatedly fed into the MLP. In doing so, the system can respond to load imbalances, channel variations, and QoS requirements in near real time. While the MLP inference step is computationally lightweight, the model itself is assumed to have been trained offline using a large dataset of channel samples, user demands, and labeled rule-based allocation-priority decisions generated from traffic, SNR, and QoS conditions. Consequently, the online overhead of the algorithm is modest: it primarily involves feature acquisition, normalization, and a forward pass through the MLP.

Such a data-driven strategy stands in contrast to manually derived solutions or purely game-theoretic schedulers that rely on iterative local updates. Here, the ANN effectively encodes a policy that aims to maximize throughput, minimize delay, and achieve better fairness or QoS satisfaction. Additional modifications, such as combining this method with a multi-cell coordination or advanced beamforming, can further enhance performance.

IV. Simulation Environment

The simulation is conducted in a controlled single-cell downlink layout implemented in Python. One gNodeB is fixed at the centre of a circular service area with a 500 m radius, and 1,000 user terminals are uniformly placed inside the cell. Therefore, all users are associated with the same serving gNodeB, and inter-cell interference or multi-gNB coordination is not considered in the current setup. This controlled topology allows the evaluation to focus on the ANN-based allocation-priority mechanism under distance-dependent SNR conditions. The simulated SNR is treated as an effective link-quality indicator that abstracts the impact of propagation and MIMO-related link processing. Therefore, the simulation does not include explicit antenna-array modelling, beamforming optimization, spatial multiplexing, or precoder design. A representative realization of the controlled single-gNB topology is shown in Fig. 1.

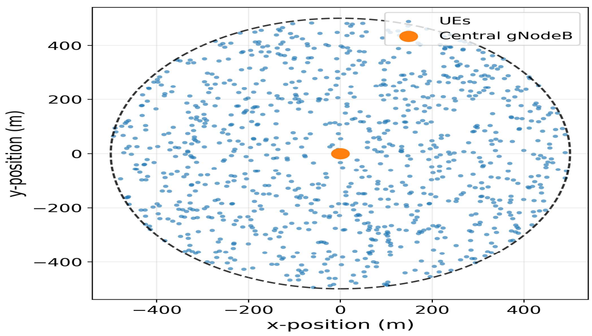

Fig. 1. Representative Single-gNB Network Topology.

For every scheduling instant the simulator synthesises a paired traffic-demand / channel-quality vector (Ti,SNRi) for each user, drawing from three stochastic scenarios. In Scenario A the offered load follows an exponential law, and the channel is modelled as a narrow normal cloud around 15 dB; Scenario B combines Weibull traffic with log-normal fading so as to emulate heterogeneous ranges; Scenario C superimposes a slow linear drift on the demand and drives the SNR with a sinusoid to emulate time-varying channel conditions. Each scenario generator produces 2,000 samples, which are split into training and testing sets for supervised ANN evaluation.

Raw features are standardized to zero mean and unit variance before ANN training. The supervised targets are generated by a deterministic rule that combines traffic demand, effective SNR, and QoS thresholds into four allocation-priority classes. The purpose of this labeling procedure is not to approximate a globally optimal scheduler, but to encode a joint decision rule that considers the three input conditions and can subsequently be reproduced by a lightweight MLP with a single forward pass. The ANN is therefore evaluated as a learned approximation of a predefined allocation policy.

The neural scheduler itself is a feed-forward Multilayer Perceptron with three hidden layers of 128, 64 and 32 ReLU units. Batch-normalisation and 30% dropout are interleaved after every dense block to stabilise convergence and guard against over-fitting, while the softmax output layer maps activations to the four discrete allocation-priority classes.

Training proceeds for at most 100 epochs with mini-batches of 64 samples, under the Adam optimiser at a learning rate of 0.0005; A Reduce-LR-on-Plateau scheduler (factor = 0.5) cut short any plateau in validation loss.

On an ordinary 12-thread desktop CPU a single epoch over Scenario A completes in roughly three seconds—fast enough to allow the extensive hyper-parameter sweep reported later.

To provide meaningful yardsticks, the same traces are fed through three reference schedulers: random assignment, round-robin cycling and a greedy max-SNR rule. Because these baselines share the topology, traffic traces and post-processing code with the ANN, any gap in throughput, delay, Jain fairness or QoS satisfaction can be ascribed purely to the decision logic itself. More details about the simulation parameters are shown in Table I.

TABLE 1. Simulation Parameters

Parameter	Value
Network topology	Simplified single-cell downlink scenario with one central gNodeB
Cell radius	500 m
Number of UEs	1,000
Final simulation duration	20 timeslots
Samples per scenario	2,000
Train-test split	80% / 20%
ANN input vector	$X_i = [T_i, SNR_i, Q_i]$
Allocation-priority classes	$c_i \in \{0, 1, 2, 3\}$
Allocation factors	$w^c = \{0.25, 0.50, 0.75, 1.00\}$
Scenario A	Exponential traffic and normally distributed SNR
Scenario B	Weibull traffic and lognormal SNR
Scenario C	Linear traffic trend and sinusoidal SNR variation
Hidden layers	[128, 64, 32]
Hidden / output activation	ReLU / Softmax
Dropout rate	0.3
L2 regularization	10^{-3}

V. Performance Evaluation

This section first reports the ANN classification diagnostics and then focuses on the network-level scheduling results. Classification accuracy and confusion matrices verify that the ANN learns the rule-based allocation-priority labels, while throughput, delay, QoS satisfaction, and fairness measure the actual scheduling impact.

A. ANN Variant Study

The model performance was evaluated through the analysis of accuracy and loss during training and validation. The graphs below show the evolution of these quantities for the three scenarios. A separate ANN was trained for each scenario. Since Scenario A achieved the strongest validation performance, it was selected for the detailed baseline comparison presented in Section V-B. Training/validation curves for the three scenarios are shown in Figs. 2–7, while their numerical scores appear in Table 1. In Scenario A (Figs. 2-3), the ANN rapidly reaches a high level of accuracy, surpassing 97% validation accuracy within just 15 epochs. Both accuracy curves stabilize by epoch 34, and the validation loss converges at around 0.15. The small difference between training and validation curves suggests that the ANN generalizes well without significant overfitting.

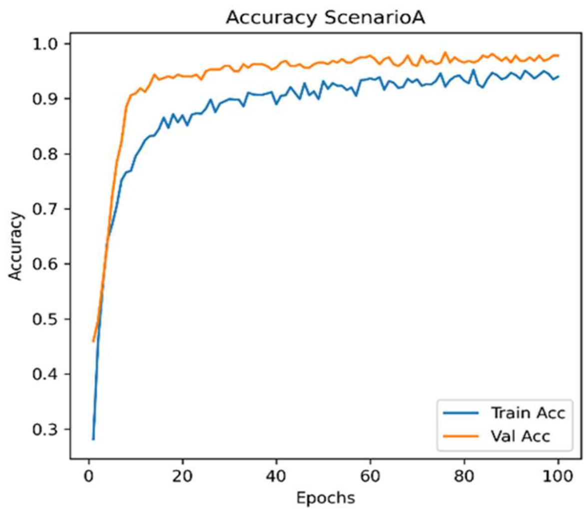

Fig. 2. Scenario A Accuracy Graph.

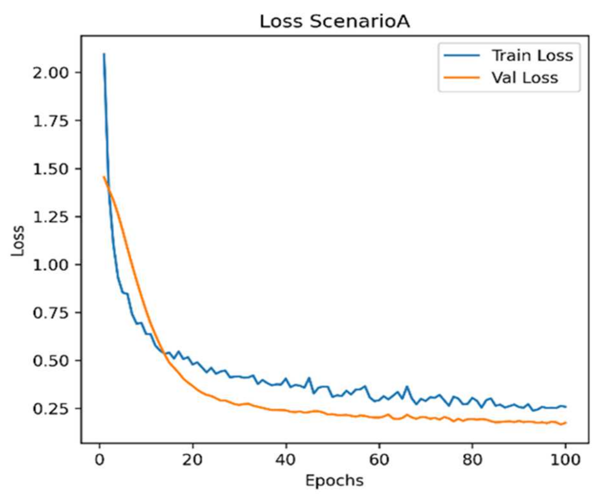

Fig. 3. Scenario A Loss Graph.

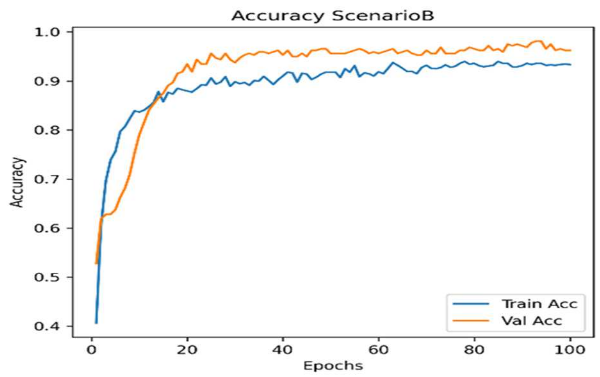

Fig. 4. Scenario B Accuracy Graph.

Scenario B (Figs. 4-5) shows a different pattern: validation accuracy increases more gradually and reaches a plateau near epoch 55, achieving a slightly lower maximum of 94%. Correspondingly, the validation loss settles higher, around 0.25, reflecting Scenario B's inherently more complex conditions. The wider spread of SNR and log-normal fading characteristics present in Scenario B cause greater difficulty for the ANN, resulting in longer training times and higher residual errors.

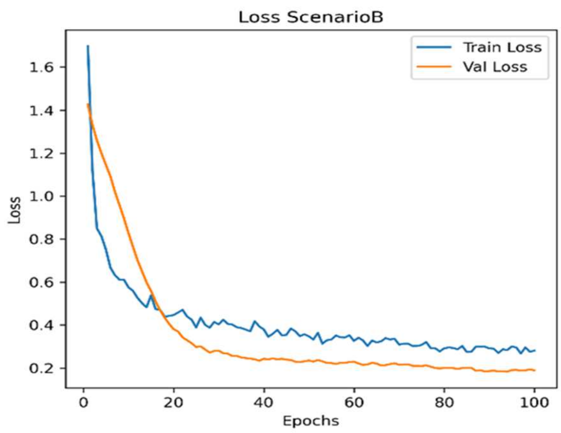

Fig. 5. Scenario B Loss Graph.

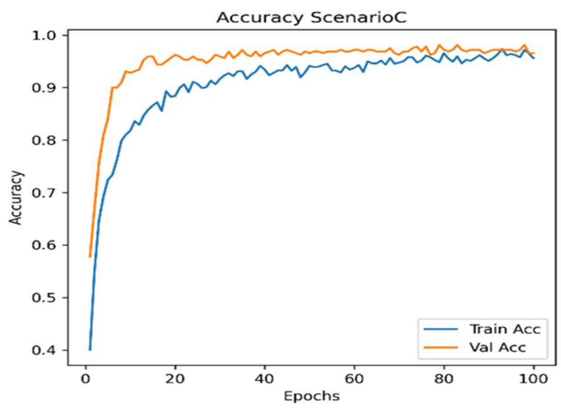

Fig. 6. Scenario C Accuracy Graph.

In Scenario C (Figs. 6-7), the training behavior exhibits intermediate complexity. Accuracy rapidly approaches 96% by epoch 30, and gradually increases to 97%, with validation loss stabilizing at around 0.18 after approximately 45 epochs. This scenario combines moderate complexity stemming from a linear traffic trend and sinusoidal SNR variation, indicating the ANN's adaptability to time-varying data conditions.

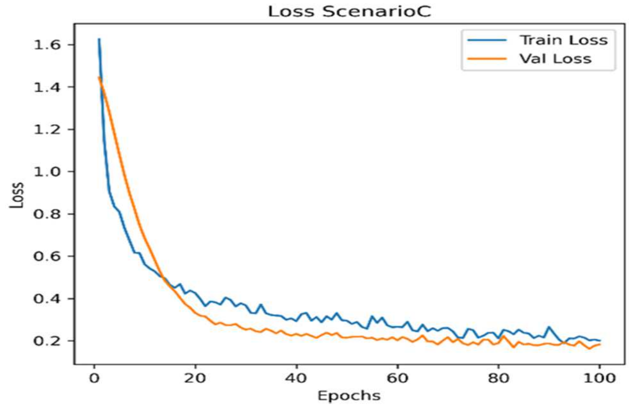

Fig. 7. Scenario C Loss Graph.

Table II complements these visual observations with concrete figures. It confirms Scenario A's superior final accuracy (validation accuracy 98.0%, validation loss 0.15), clearly distinguishing it as the simplest among the three. Scenario B, conversely, shows the lowest accuracy (94%) and highest validation loss (0.25), reflecting its increased complexity. Scenario C sits between these two extremes, achieving 97% accuracy and a moderate loss of 0.18. Across all scenarios, the negligible gap (less than 1%) between training and validation accuracies indicates robust model training without substantial overfitting, reinforcing the effectiveness of the selected dropout parameters.

TABLE II. TRAINING COMPARISON BETWEEN SCENARIOS.

Scenario	Train Accuracy	Validation Accuracy	Train Loss	Validation Loss
A	98.5%	98.0%	0.10	0.15
B	95.7%	94.0%	0.20	0.25
C	98.0%	97.0%	0.15	0.18

Classifier performance at the individual class level was assessed through confusion matrices and summarized using aggregate metrics.

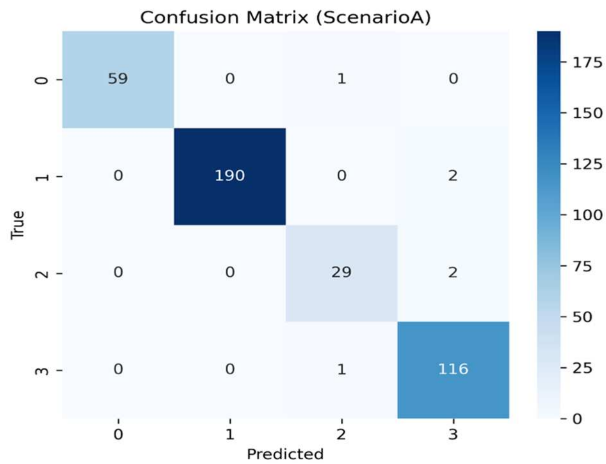

Fig. 8. Scenario A Confusion Matrix.

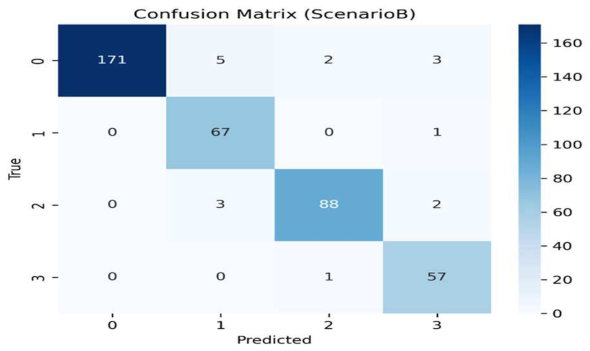

Fig. 9. Scenario B Confusion Matrix.

Figs. 8-10 show Scenarios A to C, where in Scenario A classification errors are minimal, totaling only six false predictions overall. Class 2 is slightly misclassified twice as class 3, while class 1 sees minimal confusion. This high accuracy across all classes leads to strong macro-precision and recall values around 97.5%, as seen in Table 2.

Scenario B shows noticeably more misclassification, primarily within classes 0 and 2. Class 0 instances are occasionally misclassified as class 1, and class 2 is confused multiple times with adjacent classes (class 0 and class 3). Consequently, precision and recall slightly decline, leading to lower macro-F1 scores (approximately 95.2%), with total false predictions rising to 17, the highest among the three scenarios.

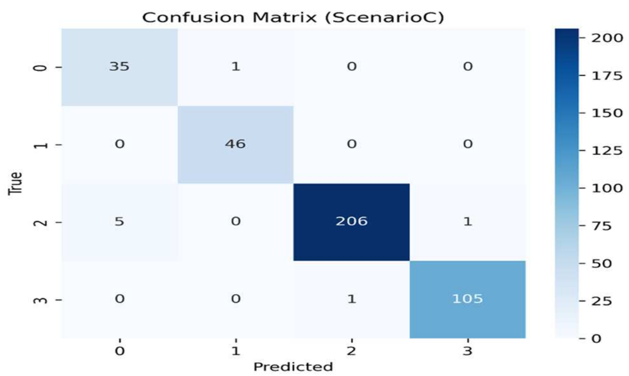

Fig. 10. Scenario C Confusion Matrix.

Scenario C shows minor confusion mainly in class 2, where several samples are misclassified into class 0. Apart from this, performance resembles Scenario A closely, maintaining a low total error count of 8 false predictions. Table 2 highlights this scenario's robust performance, with macro-precision around 96% and recall at approximately 98.4%.

Overall, the class-level analysis from Table 2 confirms that Scenario B presents the most challenging classification task, while Scenarios A and C display high classification accuracy and balanced performance across all classes.

TABLE 2. CONFUSION MATRIX COMPARISON ACROSS SCENARIOS.

Scenario	**Accuracy (%)**	**Precision (macro)**	**Recall (macro)**	**F1-score (macro)**	**False Predictions**
A	98.5	0.9755	0.9750	0.9752	6
B	95.7	0.9413	0.9648	0.9518	17
C	98.0	0.9599	0.9836	0.9710	8

Finally, Figs. 11-13 track predicted versus actual rates; Scenario A aligns tightly along the diagonal, whereas the scatter widens in Scenario B, echoing the 'Moderate' agreement level of Table 3.

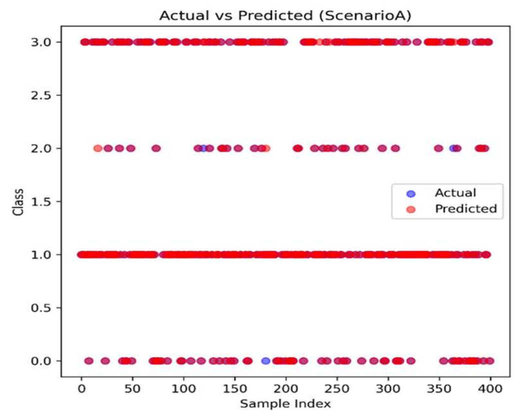

Fig. 11. Actual vs. Predicted Classes in Scenario A.

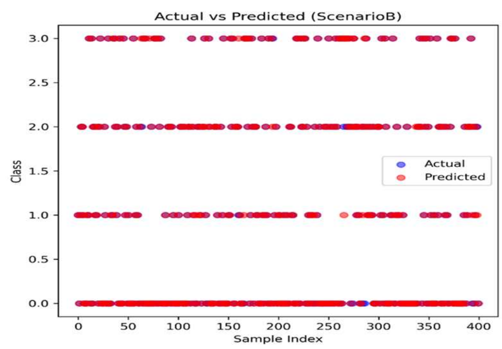

Fig. 12. Actual vs. Predicted Classes in Scenario B.

In Scenario A, predictions tightly overlap actual classes, indicating excellent classifier accuracy with minimal deviation. Almost every prediction aligns perfectly with the corresponding actual data points, confirming the "Very High" agreement level described in Table 3.

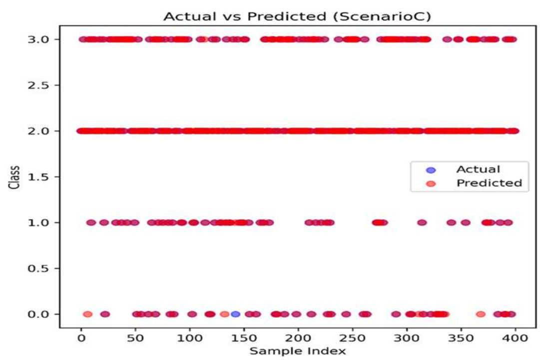

Fig. 13. Actual vs. Predicted Classes in Scenario C.

In contrast, Scenario B exhibits increased scattering. Several predictions deviate noticeably from their true labels, particularly in classes 0 and 2. This visual dispersion directly matches the "Moderate" accuracy rating provided in Table 3, highlighting that Scenario B experiences more classification challenges, likely stemming from more complex data conditions and greater class overlap. Scenario C demonstrates mostly accurate predictions, closely resembling Scenario A, though with occasional minor deviations in class 2. The scattered misclassifications are fewer than Scenario B, earning Scenario C a "High" accuracy rating in Table 3, with partial confusion specifically noted for class 2.

TABLE 3. ACCURACY COMPARISON ACROSS SCENARIOS.

Scenario	Agreement Level	Deviations	Class-wise Performance	Possible Issues
A	Very High	Minimal	All classes	—
B	Moderate	Several	Class 0 & 2	Higher class overlap
C	High	Few	Class 2	Partial confusion

These observations reinforce that Scenario A provides the strongest predictive accuracy, Scenario B presents notable classification challenges, and Scenario C achieves reliable predictions despite limited confusion in one class.

B. Baseline Comparison

The trained scheduler was evaluated over the final simulation timeslots using 1,000 users distributed inside the single-cell area. Four headline metrics were logged per run: average user throughput, mean one-way delay, Jain's fairness index J, and QoS-satisfaction ratio (percentage of users whose instantaneous rate met or exceeded their target demand). Because the ranking of the candidate schedulers remained unchanged across all three traffic/channel profiles, the discussion below quotes the representative figures obtained in Scenario A, where the ANN achieved its best overall balance. Key distributions and averages are collected in Figs. 14 to 16 and Table 4.

Fig. 14 compares the throughput distributions across the tested methods. The ANN scheduler significantly outperforms the conventional methods, achieving a median user rate of approximately 4 Mbps and a broad distribution extending from roughly 1 to 10 Mbps. In contrast, Round-Robin and Random scheduling yield medians near 3.3 Mbps, while Max-SNR clusters around approximately 1.4 Mbps. When averaging across the final simulation samples, Table 4 shows the ANN achieving an average throughput of 4.45 Mbps, clearly surpassing Random (3.48 Mbps), Round-Robin (3.49 Mbps), and Max-SNR (1.41 Mbps). This demonstrates the ANN's capability to exploit channel variations effectively, translating into higher data rates for users.

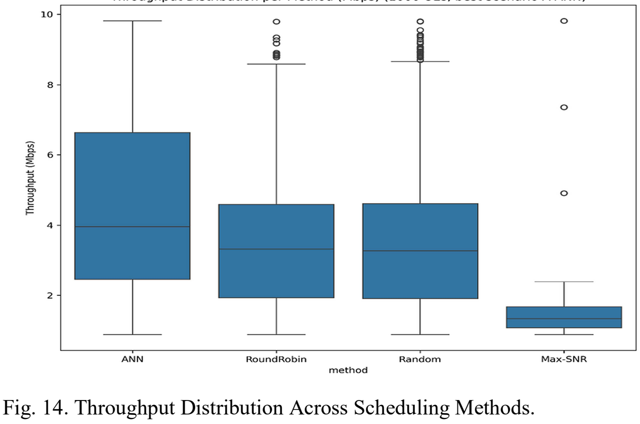

Fig. 14. Throughput Distribution Across Scheduling Methods.

Fig. 16 illustrates the QoS satisfaction rates across the four schedulers. The ANN achieves the highest QoS adherence, fulfilling target demands for approximately 89.4% of the user samples. This outcome represents a clear improvement of around 15 percentage points over Random (74.8%) and Round-Robin (75.6%) and nearly doubles the QoS adherence of Max-SNR, which only meets user expectations in approximately half (49.6%) of the cases. Such superior QoS results directly stem from the ANN's optimized balance of high throughput and low delay.

The delay metric, shown in Fig. 15, further highlights ANN's robustness. The ANN consistently maintains user delays tightly within a narrow interval (0.47 s–0.52 s) and achieves an average delay of just 0.492 s. In comparison, Random and Round-Robin scheduling exhibit noticeably higher delays around 0.8 s, while Max-SNR suffers substantial queue buildups, averaging delays as high as 1.53 s. The ANN scheduler's stable, predictable delay performance confirms its resilience and stability under diverse traffic conditions.

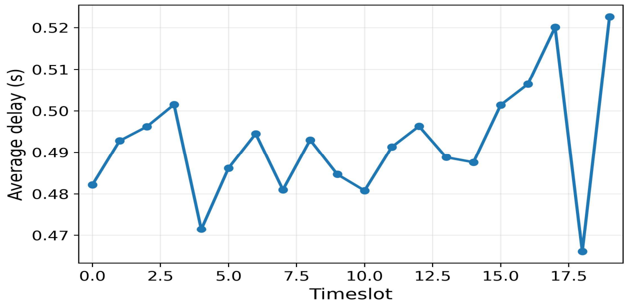

Fig. 15. Average ANN Delay per Timeslot in Scenario A.

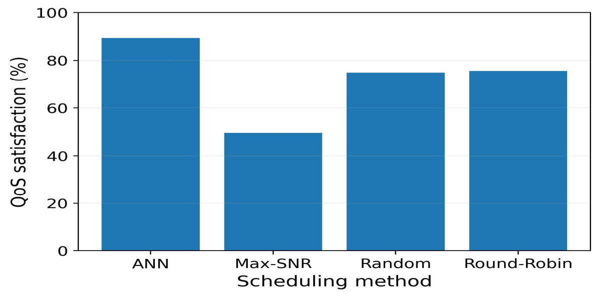

Fig. 16. QoS Satisfaction Across Scheduling Methods.

Regarding fairness, as reported in Table 4, the ANN achieves a Jain fairness index of 0.787, which is slightly higher than Random (0.776) and Round-Robin (0.775), while remaining below Max-SNR (0.886). However, Max-SNR obtains this fairness value together with much lower throughput, higher delay, and weaker QoS satisfaction. Therefore, the ANN provides a more balanced trade-off across throughput, delay, QoS satisfaction, and fairness.

A brief hyperparameter sensitivity discussion was conducted to support the selected ANN configuration. The final model uses hidden layers [128, 64, 32], dropout 0.3, L2 regularization 10^{-3}, batch size 64, and learning rate 0.0005. These settings provided stable training and validation behaviour in the evaluated scenarios. Since a full hyperparameter optimization study is outside the scope of this paper, the results are presented as configuration support rather than proof of global optimality. In summary, the ANN scheduler demonstrates significant and consistent performance improvements: approximately 30% higher average throughput, around 40% reduced delay, and roughly 15 percentage points higher QoS adherence compared to conventional schedulers, with only a marginal trade-off in fairness. These findings underline the efficacy of employing a simple, feed-forward ANN architecture as a practical and adaptive solution for dynamically managing 5G MIMO resources, clearly surpassing legacy scheduling heuristics that rely solely on cyclic allocation or instantaneous channel conditions.

TABLE 4. FINAL METHOD COMPARISON.

Method	Throughput (Mbps)	Delay (s)	QoS Satisfaction (%)	Fairness
ANN	4.45	0.4922	89.36	0.7869
Max-SNR	1.41	1.5313	49.55	0.8861
Random	3.48	0.8053	74.83	0.7757
Round-Robin	3.49	0.7924	75.55	0.7753

VI. CONCLUSION AND FUTURE WORK

This study evaluated a lightweight feed-forward ANN scheduler for dynamic downlink resource allocation in a simplified single-cell 5G MIMO simulation environment, where MIMO-related physical-layer effects were abstracted through effective user-side SNR values. Using the Scenario A ANN, which achieved the strongest validation performance among the separately trained scenario-specific models, the proposed supervised classifier outperformed the baseline policies (Random, Round-Robin, Max-SNR). In the reported evaluation, it achieved approximately 30% higher throughput, 40% lower mean queuing delay, and 15 percentage points higher QoS satisfaction. Fairness, measured through Jain's index, remained competitive and was slightly higher than Random and Round-Robin, although lower than Max-SNR. These gains show that the ANN can exploit joint patterns in traffic demand and channel conditions, outperforming the implemented heuristic baselines under the evaluated setup.

Several research directions emerge. Extending the current single-gNB setup to multi-cell environments would allow the study of inter-cell interference, handover, and coordinated scheduling, requiring richer input features and interference-aware ANN design. Transitioning from offline to online learning could improve adaptability through incremental updates, reducing retraining costs. Integrating interpretability techniques (e.g., SHAP values, attention) would enhance transparency, trust, and compliance. Hardware-level evaluation on FPGAs or low-precision accelerators is also critical for quantifying real-time complexity, latency, and energy use. Practical deployment demands robust performance under diverse network conditions. This includes testing real traces, optimizing inference for low latency, and distributing ANN architecture across edge nodes to align with 5G's edge-centric design. Hybrid frameworks combining ANN predictions with heuristic fallbacks could ensure stability under rare or unforeseen conditions. Finally, safeguarding ANN schedulers against adversarial interference or anomalous inputs requires anomaly detection and robust training.

By addressing these challenges, ANN-based scheduling can progress from simulation results to reliable deployment in real-world 5G and beyond-5G networks.

REFERENCES

[1] I. A. Bartsiokas, P. K. Gkonis, D. I. Kaklamani, and I. S. Venieris, "ML-Based Radio Resource Management in 5G and Beyond Networks: A Survey," IEEE Access, vol. 10, pp. 83507–83528, 2022.

[2] M. A. Kamal, H. W. Raza, M. M. Alam, and A. A. B. Sajak, "Resource Allocation Schemes for 5G Network: A Systematic Review," Sensors, vol. 21, no. 19, Art. no. 6588, 2021.

[3] A. Mamane, M. Fattah, M. E. Ghazi, M. E. Bekkali, Y. Balboul, and S. Mazer, "Scheduling Algorithms for 5G Networks and Beyond: Classification and Survey," IEEE Access, vol. 10, pp. 51643–51661, 2022, doi: 10.1109/ACCESS.2022.3174579.

[4] R. Chataut and R. Akl, "Massive MIMO Systems for 5G and beyond Networks—Overview, Recent Trends, Challenges, and Future Research Direction," Sensors, vol. 20, no. 10, Art. no. 2753, 2020, doi: 10.3390/s20102753.

[5] H. E. Benmadani, M. A. Ouamri, M. Azni, and T. E. Alharbi, "Dynamic eMBB scheduling strategy for GBR and NGBR in Non Standalone 5G NR: A deep reinforcement learning approach", Computer Networks, vol. 272, Art. no. 111692, 2025.

[6] A. Ly and Y.-D. Yao, "A Review of Deep Learning in 5G Research: Channel Coding, Massive MIMO, Multiple Access, Resource Allocation, and Network Security," IEEE Open Journal of the Communications Society, vol. 2, pp. 396–408, 2021, doi: 10.1109/OJCOMS.2021.3058353.

[7] M. Chen, U. Challita, W. Saad, C. Yin, and M. Debbah, "Artificial Neural Networks-Based Machine Learning for Wireless Networks: A Tutorial," IEEE Communications Surveys & Tutorials, vol. 21, no. 4, pp. 3039–3071, 2019.

[8] S. Nérondat, X. Leturc, P. Ciblat, and C. J. Le Martret, "Efficient 5G Resource Block Scheduling Using Action Branching and Transformer Networks", in Proc. 2025 IEEE International Conference on Machine Learning for Communication and Networking (ICMLCN), Barcelona, Spain, pp. 1–6, 2025.

[9] S.-C. Tseng, Z.-W. Liu, Y.-C. Chou, and C.-W. Huang, "Radio Resource Scheduling for 5G NR via Deep Deterministic Policy Gradient," in Proc. 2019 IEEE International Conference on Communications Workshops (ICC Workshops), Shanghai, China, pp. 1–6, 2019.

[10] 3GPP TR 38.901, "Study on Channel Model for Frequencies from 0.5 to 100 GHz," 3rd Generation Partnership Project, Release 15, 2020.

[11] J. Navarro-Ortiz, P. Romero-Diaz, S. Sendra, P. Ameigeiras, J. J. Ramos-Munoz, and J. M. Lopez-Soler, "A Survey on 5G Usage Scenarios and Traffic Models," IEEE Communications Surveys & Tutorials, vol. 22, no. 2, pp. 905–929, 2020.

[12] S. Xu, P. Liu, R. Wang, and S. S. Panwar, "Realtime Scheduling and Power Allocation Using Deep Neural Networks," in Proc. 2019 IEEE Wireless Communications and Networking Conference (WCNC), Marrakesh, Morocco, pp. 1–5, 2019, doi: 10.1109/WCNC.2019.8886140.