An Optimized Hybrid Bio-Inspired Framework for Machine Learning-Based 5G Resource Allocation

Esraa Taha Heussien*, Walled Khalid Abdulwahab

Department of Computer Science, College of Computer Science and Information Technology,
University of Kirkuk, Kirkuk, Iraq

Received 19 July 2026; received in revised form 21 August 2026; accepted 24 August 2026
DOI: https://doi.org/10.46604/peti.2026.16627

Abstract

This study presents a lightweight hybrid framework for optimizing radio resource allocation in 5G networks, enabling high prediction accuracy with low computational complexity for edge deployment. A binary grey wolf optimization (BGWO) algorithm is used for feature selection, while a continuous GWO algorithm is used for hyperparameter adjustment of nine regression models trained using machine learning on a pre-processed public dataset for 5G networks. To ensure evaluation integrity, the dataset is segmented for training, validation, and testing before feature selection and expansion, preventing data leakage. Model performance is assessed using mean squared error (MSE) and the coefficient of determination (R^2). The optimized decision tree achieves the best overall performance, improving R^2 from 46.32% to 99.44% and reducing MSE from 0.00470 to 0.00005, while random forest and support vector machine (SVM) exceed 99% R^2. These results demonstrate that the proposed framework achieves accurate and computationally efficient resource allocation for 5G networks.

Keywords: 5G networks, resource allocation, machine learning, grey wolf optimizer

1. Introduction

5G mobile networks must simultaneously support enhanced mobile broadband (eMBB), ultra-reliable low-latency communications (URLLC), and massive machine-type communications (mMTC), each involving a large number of devices generating significant data volumes with varying quality of service (QoS) requirements [1-2]. Consequently, resource allocation becomes a complex optimization problem that must continuously adapt to changing channel conditions, traffic demands, and network slicing constraints [2-3].

Recent studies have demonstrated that machine learning (ML) techniques can significantly improve network performance through intelligent user clustering and resource management in 5G environments [4-5]. However, directly applying these regression models to raw network telemetry data often yields limited predictive accuracy due to feature redundancy, data noise, and mixed data types. Furthermore, high computational requirements can prevent such models from meeting the strict timing constraints of near-real-time radio access network (RAN) control [6-7].

To address these challenges, this paper proposes a lightweight hybrid optimization framework that integrates binary grey wolf optimization (BGWO) for feature selection with continuous GWO for hyperparameter tuning. The proposed framework minimizes the number of required features while maximizing prediction accuracy, thereby maintaining low computational complexity suitable for edge deployment.

* Corresponding author.stcm24010@uokirkuk.edu.iq

This paper is organized as follows. Section 2 reviews the related literature. Section 3 presents the materials and methods. Section 4 presents results. Section 5 discusses the overall findings. Section 6 discusses computational complexity and practical deployment. Section 7 addresses threats to validity and reviewer-ready improvements, and Section 8 concludes the paper with future work.

2. Related Work and Research Gap

The existing works in Table 1, on 5G resource allocation based on ML can be categorized into four groups: supervised learning, deep learning, reinforcement learning, and metaheuristic optimization. More comprehensive Artificial Intelligence (AI)-powered systems were also proposed [8]. Supervised regression models, such as support vector regression (SVR), decision tree (DT), random forest (RF), and gradient boosting (GB), are widely used because of their low inference cost and suitability for offline training; they have been applied to traffic prediction, internet of things resource allocation, spectrum allocation based on traffic forecasts [9-10], and user clustering [11]. These models depend heavily on the quality and correct hyperparameter tuning. In addition, previous studies proposed a deep reinforcement learning-based user association and load framework for heterogeneous 5G networks, which demonstrates improved cumulative transmission rates under different network scenarios [12].

Deep learning and reinforcement learning can model complex networks and adaptive allocation of resources [13-14]; however, they are often limited by large dataset requirements, extensive training periods, and heavy computational resources demands [15], making them challenging for near-real-time edge environments [16]. Multi-agent reinforcement learning extends this capability to distributed scheduling consider scenarios, but introduces convergence and exploration overhead [17]. Regarding metaheuristic algorithms, optimization using metaheuristic algorithms, specifically GWO has been effectively applied to non-differentiable objective functions for both feature selection and model parameter tuning. For instance, a closely related study utilized feature selection alongside greylag goose optimization (GGO) to enhance 5G performance [18], where the reported coefficient of determination (R^2) for resource allocation prediction exceeds 99% [19]. Furthermore, edge- and fog-based deployments rely on lightweight resource-allocation strategies to minimize online computational cost [20].

In most existing studies, the optimization of the feature space or the learning model is conducted separately. To overcome this limitation, this paper proposes a unified framework combining BGWO for feature selection with continuous GWO. This integrated framework requires less memory, performs more efficiently, and forms an intelligent 5G resource optimization system.

Table 1 The proposed framework with AI-based 5G

Study group	Typical methods	Strength	Limitations addressed by this paper
Supervised ML [8-11]	SVR, K-nearest neighbors (KNN), DT, RF, GB	Fast offline training and transparent evaluation	Often sensitive to raw noisy features and hyperparameter tuning
Deep Learning [13-15]	Convolutional neural network (CNN), Long short-term memory (LSTM), Transformer, and multi-layer perceptron (MLP) variants	Strong non-linear modeling capacity	Higher training/inference complexity and large data requirements
Reinforcement Learning [12, 16-17]	Deep Q-learning (DQN), State–action–reward–state–action (SARSA), DDQN-LSTM, Multi-agent DRL	Adaptive control under dynamic network states	Online convergence and exploration overhead can be risky for live RAN control
Metaheuristic ML [18-20]	GWO, BGWO, GGO	Efficient feature selection and hyperparameter optimization	Requires separation between offline optimization and online inference

3. Materials and Methods

3.1. Dataset and preprocessing protocol

The experimental dataset is a public dataset for 5G resource deployment, representing the network as in Fig. 1. It consists of 400 records containing variables related to the physical layer and services. Each record corresponds to a user session and includes properties such as signal strength, latency, requested bandwidth, allocated bandwidth, application type, and continuous resource allocation goal. Because many values are stored as character strings in geometric units, direct modeling requires preprocessing.

As summarized in Table 2, the preprocessing protocol includes decomposition, digitization, categorical encoding, normalization, and expansion. Specifically, decomposition converts raw entries—such as signal power in dBm/milliwatts, latency in milliseconds, and bandwidth in megabits per second into numerical values. The application type is converted into numerical labels suitable for regression models. The numerical features are scaled subsequently using min-max normalization.

Furthermore, since the available dataset contains 400 records, Gaussian noise is added exclusively to the training data to increase training samples. This data augmentation is applied only to the training portion, leaving validation and test portions untouched. Finally, the dataset is partitioned into training, validation, and test sets using a 70/20/10% ratio. The validation set is used during optimization, while the test set is kept aside for the final evaluation.

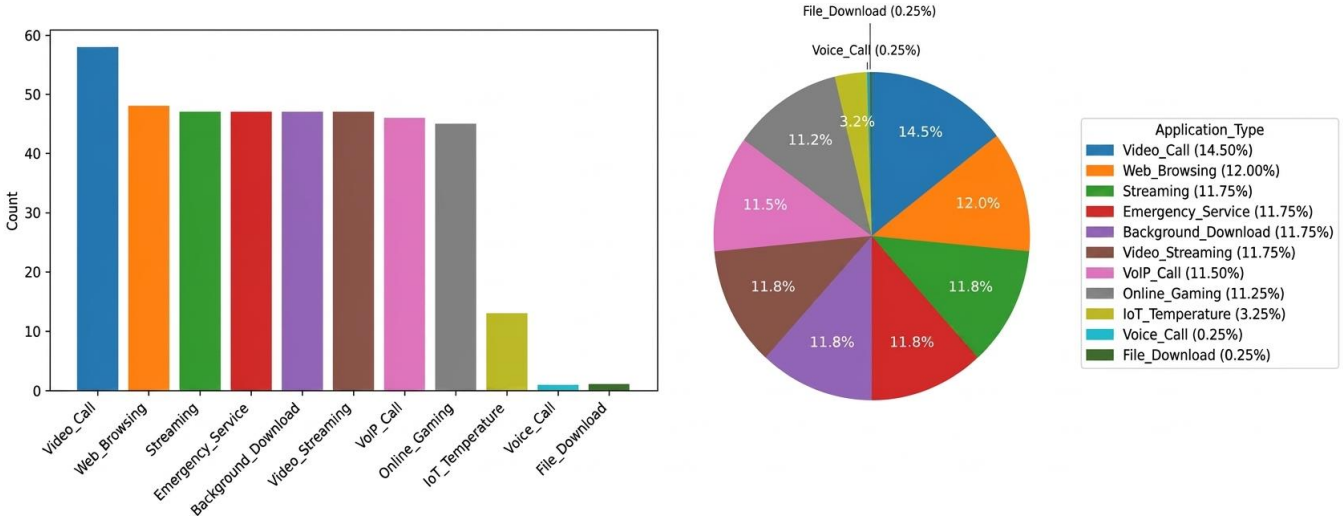

Fig. 1 Distribution of application types

Table 2 Description of the 5G resource allocation dataset

Parameter	Description
Dataset type	Public 5G resource allocation dataset
Number of records	400
Number of variables	8
Observation unit	User session
Timestamp	Time of the recorded user session
User ID	Unique identifier for each user
Application type	Type of application/service associated with the user
Signal strength	Received signal-strength measurement
Latency	Network latency measurement
Required bandwidth	Bandwidth required by the user/application
Allocated bandwidth	Bandwidth allocated to the user
Resource allocation	Resource-allocation target
Data format	Mixed numerical and categorical/string values before preprocessing

The main numerical variables in the dataset are presented in a Pearson correlation matrix in Fig 2. The matrix is analyzed to evaluate feature engineering and the relationships between the variables. There is a strong positive correlation between the required_bandwidth and allocated_bandwidth. Signal_strength is also positively correlated with both bandwidth variables, while other variables have relatively weak relationships with Latency. Correlation values of the input features with resource allocation are relatively low. Hence, the correlation results do not pinpoint any one feature as being enough to predict resource allocation. To address this concern, feature selection is also carried out with BGWO in the next optimization stage.

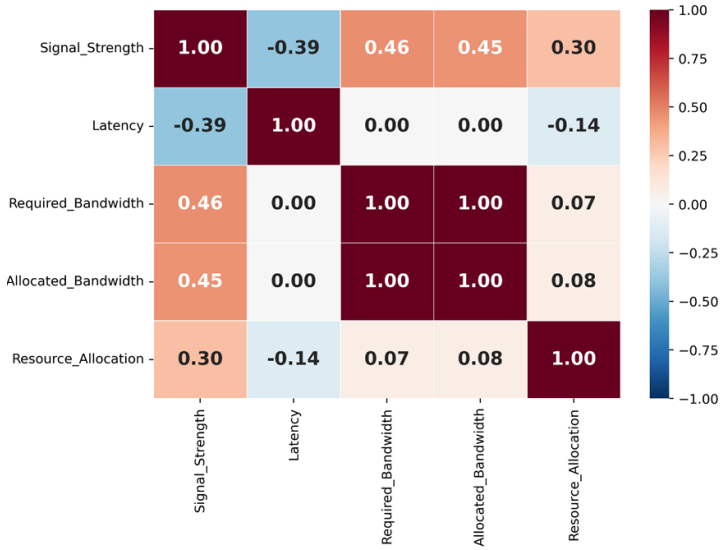

Fig. 2 Correlation heatmap of variables

3.2. *System scenario and resource-allocation target*

The experimental scenario considers 400 user-session records obtained from a public 5G dataset. Each record is associated with a user_id and an application type and contains network and resource-related attributes, including signal strength, latency, required bandwidth, and allocated bandwidth. The dataset also provides a resource allocation ratio, which is treated as the target variable for the machine learning models. Accordingly, the proposed framework learns the relationship between the observed user-session characteristics and the corresponding resource allocation ratio. The resource allocation ratio is used as provided by the dataset rather than being recalculated by the proposed model. Therefore, the experimental objective is to predict the resource-allocation ratio from the available 5G service and network characteristics.

3.3. *Feature engineering*

The basic representation consists of thirteen filter variables and derived terms, as shown in Fig. 3. This expanded feature space contains useful but also redundant information, making it susceptible to the high-dimensional problem. Consequently, the proposed framework creates a compact set. Table 3 summarizes the four feature sets, including the calibrated signal power, the required bandwidth, the allocated bandwidth, and the signal-latency interaction term. Signal power provides a direct measure of the radio link quality. The required bandwidth represents the demand imposed by the user application. Allocated bandwidth captures the resource-mapping response network. The signal-latency interaction term is designed to express cross-layer degradation when a weak radio link coincides with increased delay. The normalized value of a numerical variable x is computed as [20]:

$$x' = \frac{(x - x_{\min})}{(x_{\max} - x_{\min})} + \varepsilon \tag{1}$$

where x' denotes the normalized (res3caled) value, x is the original raw data value, x_{min} and x_{max} represent the minimum and maximum values within the dataset feature range, and ε is a small positive constant added to prevent division-by-zero or numerical instability.

Furthermore, the interaction term is computed [17] as:

$$\text{sig\_lat} = \text{signal strength normalized} \times \text{latency normalized} \tag{2}$$

The purpose of this interaction is to give the model a compact non-linear descriptor without forcing the deployed estimator to learn every interaction implicitly from the raw feature space. The feature-selection objective can be expressed as minimizing validation MSE while penalizing unnecessary feature cardinality [17]:

$$f(S,\theta) = \text{MSE}_{\text{valid}}(M\theta(X_S)y) + \lambda \mid S \mid D \tag{3}$$

where S is the selected feature mask, θ denotes model hyperparameters, λ is a feature-count penalty, and D is the number of available features. MSE_{valid} represents the mean squared error on the validation set, $M\theta$ is the predictive model parameterized by θ, X_S denotes the selected feature subset, and y is the target output.

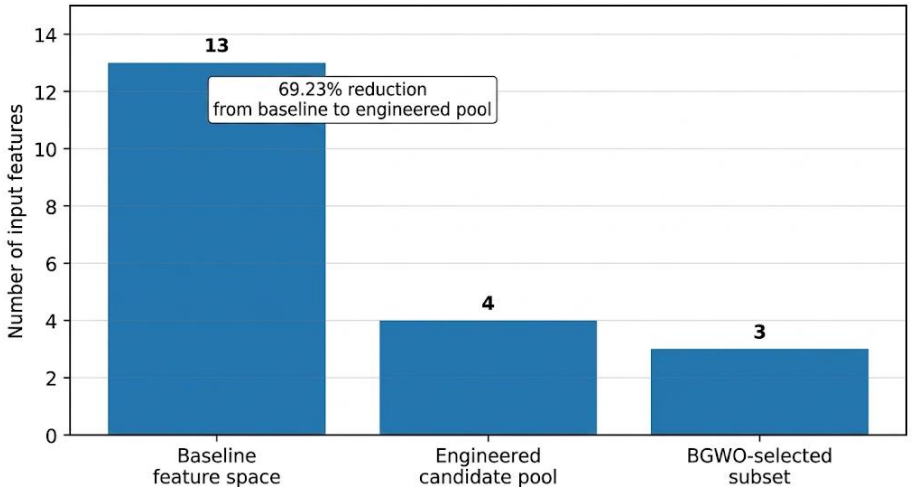

Fig. 3 Dimensionality reduction for edge-oriented inference

Table 3 BGWO feature selection

Feature token	Engineering interpretation	BGWO decision	Relative contribution
Signal strength	Normalized received radio-link quality	Rejected in the final mask	High contribution
Required bandwidth	Application-layer throughput demand	Selected	High contribution
App_encoded	Throughput/resource block mapping indicator	Selected	High contribution
Signal-latency interaction	Cross-layer non-linear degradation descriptor	Selected	Redundant under the tested feature subset

3.4. Binary grey wolf optimization for feature selection

As demonstrated in Table 4, while the BGWO framework successfully identified the most relevant feature set .However, it is not retained as the final framework because it only optimizes feature selection, leaving the machine learning models with their default hyperparameters. Experimental evaluation shows that feature selection alone is insufficient to achieve optimal predictive performance for all models. Since model accuracy depends on both relevant features and appropriate hyperparameter values, relying solely on BGWO leads to suboptimal learning performance. Therefore, BGWO is used only as an intermediate benchmark to assess the contribution of feature selection. Subsequently, the proposed framework incorporates a continuous GWO algorithm to optimize the model's hyperparameters, thus enabling the simultaneous optimization of both the feature set and the learning process. This results in better prediction accuracy, a lower mean squared error, and higher R^2 values

Table 4 Performance of the BGWO-Only framework

Model	Baseline MSE	BGWO only MSE	Baseline R^2	BGWO only R^2
Linear regression	0.00790	0.00558	0.0895	0.3176
Polynomial regression	0.00220	0.00103	0.7375	0.8744
DT	0.00470	0.00005	0.4632	0.9944
GB regressor	0.00230	0.00010	0.7280	0.9872
RF	0.00200	0.00001	0.7654	0.9988
Support vector machine (SVM)	0.00630	0.00011	0.2727	0.9866
KNN	0.00410	0.00026	0.5293	0.9677
Neural networks (NN) (100 epochs)	0.00240	0.00138	0.7293	0.8313
NN (200 epochs)	0.00150	0.00119	0.8235	0.8542

3.5. GWO-only ablation analysis

To isolate the effect of hyperparameter optimization, the GWO-only configuration is evaluated independently without applying BGWO-based feature selection. In this configuration, GWO is used solely to optimize the model-specific hyperparameters of the evaluated regression algorithms. The baseline and GWO-optimized performance results are presented in Table 5

GWO produces different levels of improvement across the evaluated models. DT, SVM, and GB achieve substantial reductions in MSE and corresponding increases in R^2 after GWO optimization. In particular, DT improves its R^2 from 0.4632 to 0.9944, while SVM improves from 0.2727 to 0.9941. GB also improves from 0.7280 to 0.9896. Other models, including Linear Regression and the neural-network configurations, show more moderate improvements. These results indicate that the effectiveness of GWO-based hyperparameter optimization is model-dependent and is more pronounced for models whose predictive performance is sensitive to parameter configuration.

The GWO-only results provide an independent assessment of the contribution of hyperparameter optimization and establish a baseline for comparison with the complete BGWO-GWO.

Table 5 Performance of the GWO-only framework

Model	Baseline MSE	GWO-only MSE	Baseline R^2	GWO-only R^2
Linear regression	0.00790	0.00558	0.0895	0.3176
Polynomial regression	0.00220	0.00103	0.7375	0.8744
DT	0.00470	0.00005	0.4632	0.9944
GB regressor	0.00230	0.00008	0.7280	0.9896
RF	0.00200	0.00007	0.7654	0.9915
SVM	0.00630	0.00005	0.2727	0.9941
KNN	0.00410	0.00026	0.5293	0.9677
NN (100 epochs)	0.00240	0.00138	0.7293	0.8313
NN (200 epochs)	0.00150	0.00119	0.8235	0.8542

3.6. Continuous GWO for model calibration

Given the selected regression models, feature selection results are used to tune the internal parameters of the models by a continuous GWO process based on the GWO's leadership-based search strategy, as presented in [15]. For instance, in a DT configuration, maximum depth, minimum samples per split, and pruning-related parameters can be added. The number of

estimators, tree depth, and split constraint are included in a RF configuration. The search vector for SVR may include a regularization parameter *C* and an epsilon-insensitive loss margin. The optimizer for GB allows tuning the learning rate, the number of estimators, estimator depth, and subsampling constraints, similar to light GBM [19], which implements GB efficiently.

The calibration objective is validation MSE, computed under a consistent cross-validation routine. The optimizer balances global exploration with local exploitation by allowing the population to follow the alpha, beta, and delta candidate vectors. This reduces the likelihood of premature convergence to poor parameter settings and gives each model a fair opportunity to reach its best performance on the selected feature subset.

As illustrated in Fig.4, the final tuned models are trained on the combined training and validation data using the optimized hyperparameters and then are evaluated on the test partition. This separation prevents the test set from influencing feature selection or hyperparameter tuning. It also mirrors a realistic deployment workflow, in which the stage is performed offline, and the final compiled model is exported for near-real-time inference.

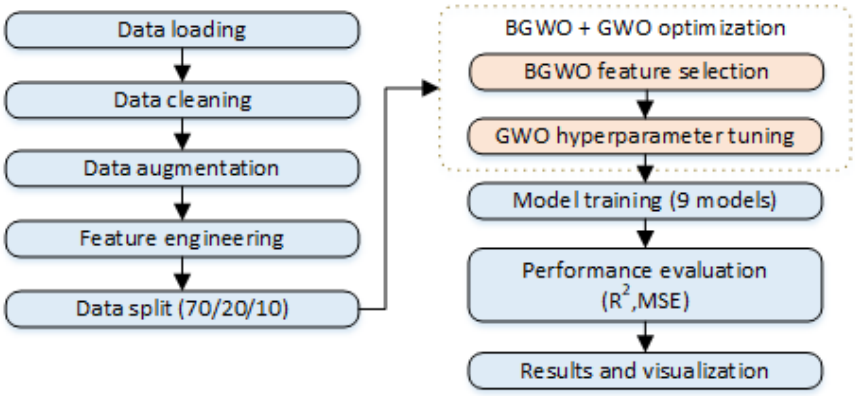

Fig. 4 Proposed BGWO-GWO framework

3.7. Algorithmic workflow

To clearly illustrate the procedural steps of the proposed optimization framework, Algorithm 1 outlines the iterative process of the resource allocation mechanism. This algorithm details how the candidate solutions are initialized, evaluated using the objective function, and updated iteratively to achieve optimal network performance while satisfying QoS constraints

Algorithm 1 Proposed BGWO-GWO resource allocation

Component	Procedure
Input	5G telemetry dataset D, candidate feature set F, number of wolves W, maximum iterations T, top feature count K
Step 1	Parse raw values, remove engineering units, encode application type, and normalize numerical variables
Step 2	Create engineered feature candidates, including normalized physical-layer variables and cross-layer interactions
Step 3	Initialize BGWO binary masks and evaluate each mask using cross-validated validation MSE
Step 4	Update alpha, beta, and delta masks; map continuous positions to binary decisions and enforce compact cardinality
Step 5	Freeze the best selected feature subset S*
Step 6	For each candidate regressor, run continuous GWO over the hyperparameter search space
Step 7	Train the optimized models on the selected features and evaluate on the unseen test partition
Output	Optimized resource-allocation predictor, final MSE, final R^2, and deployment recommendation

3.8. Evaluation metrics and deployment criteria

Two primary statistical metrics are used. The first is MSE, which measures the average squared difference between the predicted and actual resource-allocation values; lower MSE indicates lower prediction error. The second is the R^2, which measures how well the model explains the variance of the target variable; a higher R^2 indicates a stronger predictive fit.

Statistical accuracy alone is insufficient to claim 5G deployment. A model might achieve a high R^2 value but be unsuitable for edge inference due to its high memory consumption or the number of computations per request. Therefore, this paper also qualitatively evaluates deployment ability. DTs are attractive because the assembled model performs deterministic threshold checks. RFs improve durability, but they evaluate multiple trees. The SVR method may require support vector kernel calculations.

This analysis highlights a practical distinction: an optimizer can be computationally intensive because it's implemented outside the system, while an exported predictor must be lightweight. Therefore, the proposed framework favors the model that maximizes accuracy while maintaining low-cost deterministic inference.

4. Results

This section presents the experimental evaluation and performance analysis of the proposed resource allocation framework. The primary objective is to validate the effectiveness of the optimized model under various network scenarios, comparing its performance metrics against baseline approaches in terms of QoS and convergence stability.

4.1. Baseline versus optimized performance

The MSE and R^2 values are displayed before and after the application of the BGWO-GWO in Table 6. All the evaluated models improve. The change for linear regression is smaller, with R^2 increasing from 8.95% to 31.76%. The accuracy of polynomial regression increases from 73.75% to 87.44%. The R^2 and MSE values of the DT are significantly improved, from 46.32% to 99.44%, and from 0.00470 to 0.00005, respectively. SVM also reaches a high R^2 of 99.41% with an MSE of 0.00005, while RF reaches 99.15% with an MSE of 0.00007.

Based on the prediction results and the computational comparison, the DT is selected as the preferred model for the proposed deployment scenario. It achieves a high R^2 and low MSE while maintaining a relatively simple tree-based structure. The selected model is shown in Fig. 5.

Table 6 Performance comparison of the baseline and BGWO-GWO models

Model	Base MSE	BGWO-GWO MSE	Base R^2	BGWO-GWO R^2	Optimization outcome
Linear regression	0.00790	0.00558	8.95%	31.76%	Bounded convergence; non-linear structure remains unresolved
Polynomial regression	0.00220	0.00103	73.75%	87.44%	Improved but still below high-accuracy models
DT	0.00470	0.00005	46.32%	99.44%	Best edge-deployable trade-off
GB	0.00230	0.00008	72.80%	98.96%	High accuracy but more sequential computation
RF	0.00200	0.00007	76.54%	99.15%	High accuracy but larger memory footprint

Table 6 Performance comparison of the baseline and BGWO-GWO models (continued)

Model	Base MSE	BGWO-GWO MSE	Base R^2	BGWO-GWO R^2	Optimization outcome
SVM	0.00630	0.00005	27.27%	99.41%	Very accurate but kernel inference is heavier
KNN	0.00410	0.00026	52.93%	96.77%	Accurate but not ideal for online edge inference
NN (100 epochs)	0.00240	0.00138	72.93%	83.13%	Improved but plateaued below optimized tree models
NN (200 epochs)	0.00150	0.00119	82.35%	85.42%	Training complexity increases without proportional gain

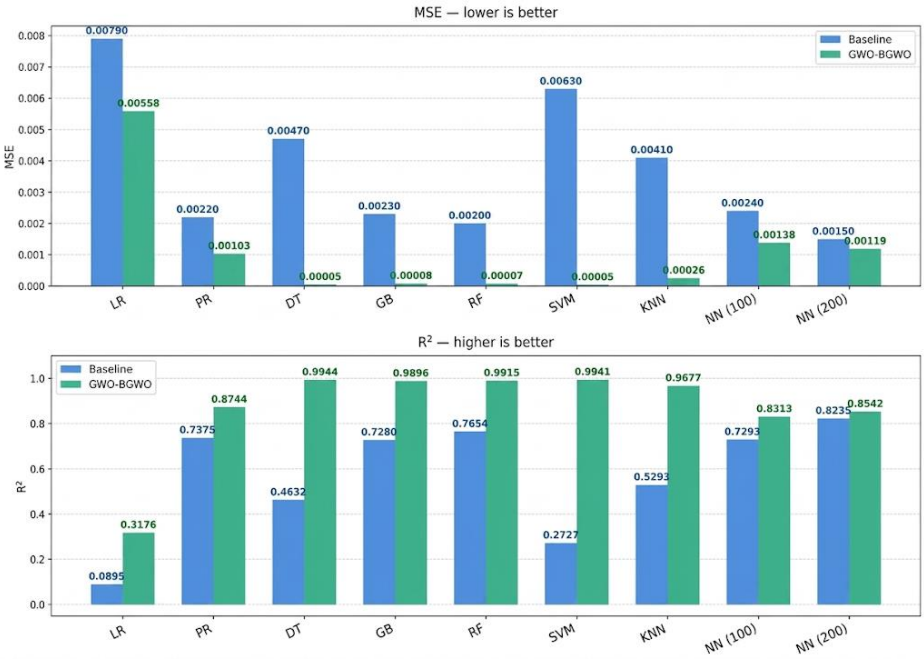

Fig. 5 Baseline vs GWO-BGWO MSE and R^2

4.2. Accuracy gain analysis

As illustrated in Fig.5, the proposed BGWO-GWO significantly improves the performance of all evaluated regression models. The SVR algorithm achieves the greatest improvement in the R^2, followed by the DT, demonstrating greater efficiency in feature selection and parameter optimization. All the improved models produce lower MSE, indicating that the proposed framework enhances prediction accuracy by reducing redundant features and improving model calibration for different regression algorithms, while maintaining robust and reliable prediction performance.

4.3. Data augmentation and distribution stability

The distributions of selected variables from the telemetry measurements (Signal_Strength, Resource_Allocation, and Latency) are shown in Fig. 6 before and after the introduction of Gaussian noise. The augmented data retain a similar shape to the original data, but becomes more spread out. This suggests that the added noise makes the training samples more diverse without substantially altering the overall distribution. The augmentation is only performed exclusively on the training data, while the validation and test data are held back for evaluation. This approach enables strength assessment through simulation but does not replace verification using network measurement devices. Consequently, the presented results are independent of external simulation artifacts.

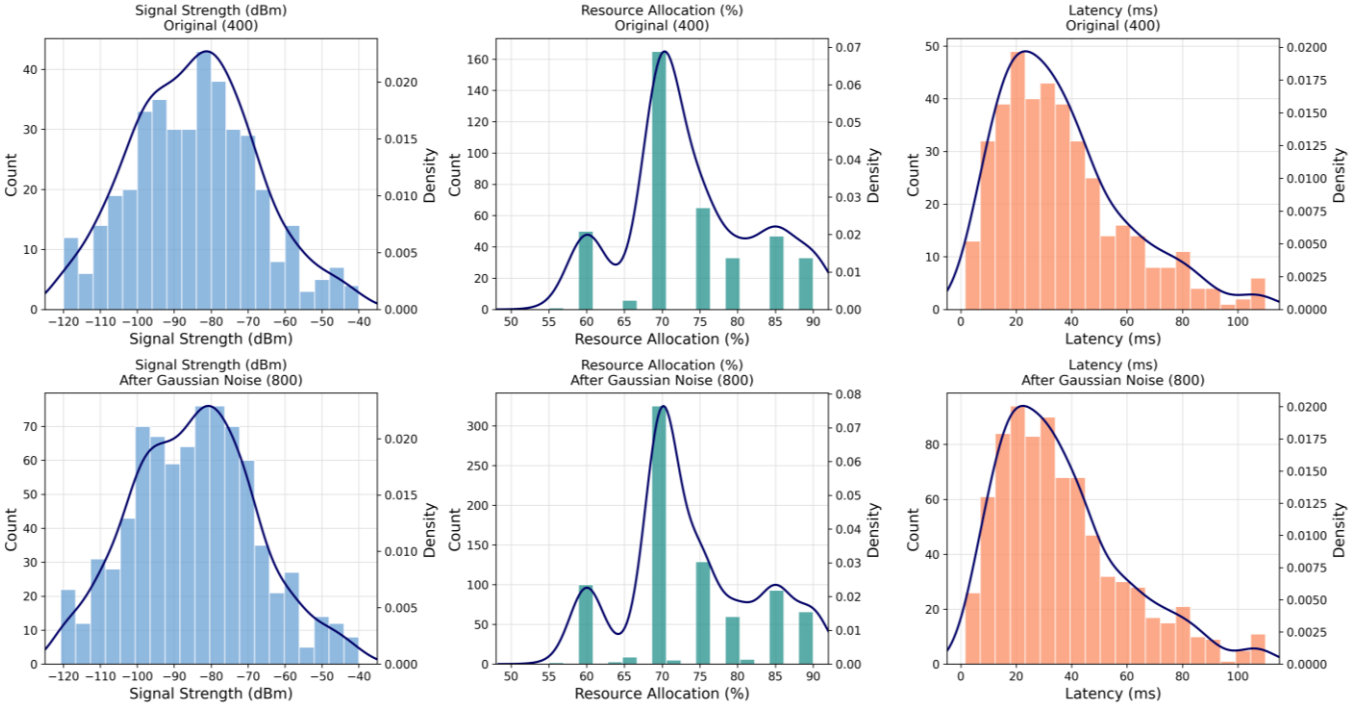

Fig. 6 Original versus noise-augmented variables

4.4. Statistical significance, robustness, and computational analysis

To assess the statistical robustness and reliability of the proposed BGWO-GWO, all nine regression models are evaluated over 42 independent runs using different random seeds in Table 7. The mean R^2, standard deviation (STD), 95% confidence interval (CI), and computational time are recorded. In addition, the Wilcoxon signed-rank test is employed to examine whether the improvement obtained by the proposed BGWO-GWO framework over the corresponding baseline configuration is statistically significant

Table 7 Statistical robustness, computational cost, and pairwise significance analysis of the evaluated models

Model	R^2 Mean ± STD	95% CI	Computational time (s) Mean ± STD	Wilcoxon W	p-value	Median error reduction	Significant
Linear regression	0.2402 ± 0.0790	[0.2156, 0.2648]	0.0011 ± 0.0002	1075.0	0.00447	0.002524	YES
Polynomial regression	0.8320 ± 0.0571	[0.8142, 0.8498]	0.0022 ± 0.0004	917.0	0.000373	0.028349	YES
DT	0.9188 ± 0.0446	[0.9049, 0.9327]	0.0029 ± 0.0006	333.5	3.35×10^{-10}	0.042685	YES
GB	0.9350 ± 0.0356	[0.9239, 0.9461]	0.1600 ± 0.0139	333.0	3.35×10^{-10}	0.032447	YES
RF	0.9372 ± 0.0368	[0.9257, 0.9486]	0.3271 ± 0.0320	271.0	4.89×10^{-11}	0.041815	YES
SVM	0.9332 ± 0.0372	[0.9216, 0.9448]	0.8697 ± 0.1071	635.0	1.15×10^{-6}	0.047571	YES
KNN	0.9364 ± 0.0302	[0.9270, 0.9458]	0.0014 ± 0.0003	935.0	5.09×10^{-4}	0.000419	YES
NN (100 epochs)	0.8379 ± 0.0651	[0.8176, 0.8581]	0.0699 ± 0.0137	388.0	1.72×10^{-9}	0.061377	YES
NN (200 epochs)	0.8086 ± 0.0826	[0.7829, 0.8344]	0.0539 ± 0.0118	319.0	2.19×10^{-10}	0.056327	YES

The results show that the proposed framework consistently improves the performance of all nine evaluated models, with statistically significant improvements ($p<0.05$) observed for every model. RF achieves the highest mean R2 of 0.9372 ± 0.0368, followed closely by KNN (0.9364 ± 0.0302) and GB (0.9350 ± 0.0356). The relatively narrow confidence interval of RF [0.9257,0.9486] further indicates stable performance across repeated runs.

Linear regression obtains a comparatively lower mean R2 of 0.2402, indicating that the underlying resource-allocation relationship is not adequately represented by a purely linear model. This result further supports the use of nonlinear machine-learning models for the investigated 5G resource-allocation problem.

The neural-network models exhibit higher variability than the tree-based models, particularly the 200-epoch configuration. This behavior indicates that increasing the training epochs does not necessarily improve generalization for the given dataset and configuration. Nevertheless, both neural-network configurations show statistically significant improvements after BGWO–GWO optimization.

4.5. Additional experimental validation

To more precisely assess the robustness of the presented BGWO-GWO framework an additional verification experiment is conducted using an independent evaluation procedure. Accordingly, nine different regression model configurations are evaluated, and the additional validation results are presented in Table 8. The base models are compared to their optimized versions after applying a selection of BGWO-based features and GWO-based hyperparameter optimization.

The results show consistent improvements in the MSE and R^2 for all evaluated models, demonstrating the effectiveness of the proposed optimization strategy. Among the tested models, the SVM achieves the best results, while the RF and neural network (200 epochs) also shows significant gains. These results confirm the hypothesis that the proposed scheme improves the prediction accuracy and generalization performance of various regression algorithms.

Table 8 Results of independent performance verification

Model	Baseline R^2	Proposed R^2	Baseline MSE	Proposed MSE
Linear regression	0.0895	0.4970	0.00790	0.00531
Polynomial regression	0.7375	0.8888	0.00220	0.00117
DT	0.4632	0.8801	0.00470	0.00126
GB	0.7280	0.8887	0.00230	0.00117
RF	0.7654	0.9012	0.00200	0.00104
SVM	0.2727	0.9528	0.00630	0.00050
KNN	0.5293	0.9166	0.00410	0.00088
MLP (100 epochs)	0.7293	0.8793	0.00240	0.00127
MLP (200 epochs)	0.8235	0.9391	0.00150	0.00064

4.6. Prediction trace and residual behavior

The prediction trajectories and residual distributions of the optimized models are shown in Fig 7. The change in the Resource_Allocation target are well predicted by the prediction curves of the tree-based models and optimized SVM, with relatively small deviations from the actual values. This comparison offers more insights than the MSE and R^2 values, as it illustrates tracking performance across individual observation.

Furthermore, the better-performing models exhibit minimal systematic prediction errors, as indicated by residual plots concentrated around zero. This behavior is critical for resource allocation because chronic underestimation or overestimation directly impacts user bandwidth provisioning. Specifically, underestimation leads to resource scarcity, whereas overestimation results in network bandwidth waste.

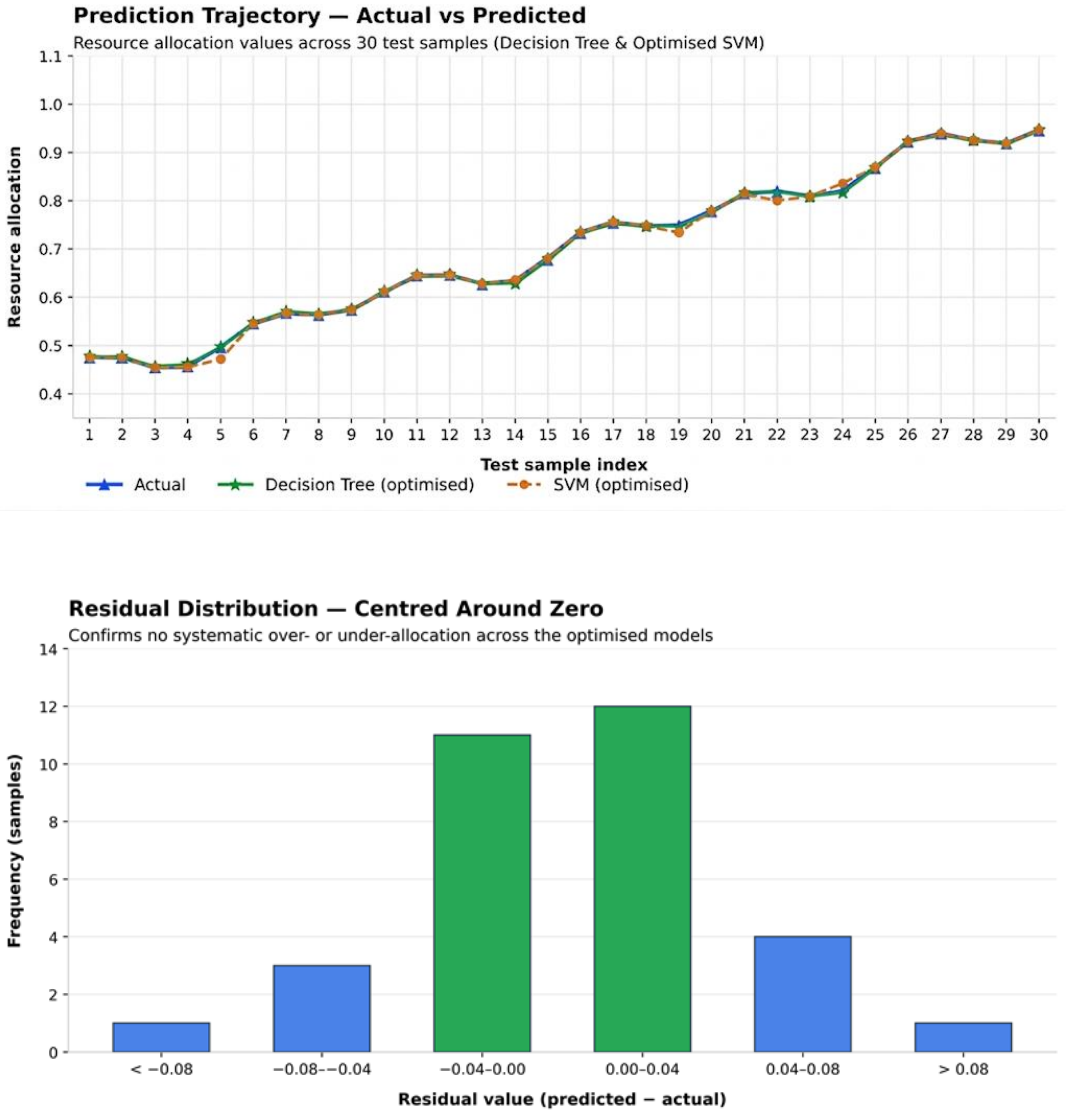

Fig. 7 Actual versus predicted resource allocation

5. Discussion

The simultaneous optimization of the feature space and hyperparameters in the proposed BGWO-GWO framework is an important advancement that enhances prediction. Apart from the initial evaluation, additional validation in Section 4 shows that these improvements are observed across nine different regression models, which means that the proposed framework is not restricted to a single learning algorithm. Among the evaluated models, the optimized DT strikes a perfect balance between prediction accuracy and computational efficiency and is well suited to allocating resources in real time in 5G networks. While the SVR and RF models achieve competitive prediction performance, their computational and memory requirements are generally higher, making them more appropriate for offline or cloud-assisted optimization.

The proposed optimization also improves the performance of MLP models, but they perform slightly worse than the top-performing lightweight models. This implies that, for the dataset used, feature selection and hyperparameter optimization are more effective in improving the prediction quality than increasing model complexity. Furthermore, the proposed framework intrinsically enables an offline and online deployment approach from an open radio access network (O-RAN) perspective. The RAN intelligent controller (RIC) (non-real-time) layer is used for optimal model preparation with the BGWO and GWO algorithms, whereas only the trained predictor is deployed in the semi-real-time layer for rapid inference. This division decreases the extra cost.

6. Computational Complexity and Practical Deployment

The computational cost of the proposed framework depends mainly on the number of search agents, iterations, candidate features, and the model-evaluation cost. Since optimization is performed offline on a compact feature space, the computational overhead is acceptable. As presented in Table 9, continuous GWO further optimizes model hyperparameters before deployment. Among the evaluated models, the optimized DT offers the best balance between prediction accuracy and computational efficiency because its inference cost depends only on tree depth. In contrast, models such as SVR and RF require higher computational resources despite achieving competitive accuracy.

For practical O-RAN deployment, BGWO and GWO are executed in the non-real-time (Non-RT) RIC to generate an optimized prediction model, while the trained model is deployed as an xApp or rApp for lightweight near-real-time resource allocation. Periodic retraining can be performed to adapt to changing network conditions.

Table 9 Complexity and deployment framework

Layer	Operation	Main computational cost	Effect on live scheduling
Data engineering	Parsing, unit conversion, encoding, normalization, and augmentation	Linear in dataset size	No direct online cost after the preprocessing pipeline is established
BGWO feature selection	Search binary masks and evaluate validation MSE	W × T × D × cross-validation cost	Offline cost; feature mask is reused in deployment
Continuous GWO	Tune model hyperparameters over selected features	Population × iterations × model training cost	Offline cost; tuned parameters are exported
DT predictor	Threshold-based resource-allocation prediction	Proportional to tree depth	Low deterministic online cost
Monitoring and retraining	Detect distribution drift and repeat optimization	Periodic batch cost	Improves robustness without affecting each scheduling

6.1. Architectural placement in O-RAN

As shown in Table 10, in practical applications, the computational complexity varies depending on hardware efficiency, including central processing unit (CPU) usage, memory usage, and inference latency. The O-RAN architecture performs computationally intensive functions, such as feature selection using the BGWO algorithm and hyperparameter optimization using the GWO algorithm in a Non-RT manner at the RIC layer or service management and orchestration layer (SMO). The trained model performs only simple inference during real-world use, enabling deployment in latency-sensitive network components with minimal computational overhead. This deployment strategy, whether online or offline, aligns the computational characteristics of the proposed framework with the practical requirements of O-RAN networks, facilitating efficient and scalable implementation in 5G networks.

Table 10 Practical deployment of AI in O-RAN

AI scheme	O-RAN layer	Latency	Typical application
Base ML model	DU / gNodeB	< 10 ms	Scheduling, HARQ, beamforming
BGWO-based model	Near-RT RIC	10–1000 ms	QoS optimization, traffic prediction, handover
BGWO + GWO	Non-RT RIC / SMO	> 1000 ms	Network slicing, capacity planning, energy optimization

6.2. Computational overhead

The training time, memory usage, CPU usage, and inference latency of the evaluated models are measured experimentally. The baseline and BGWO–GWO optimized versions of all nine models are compared in Table 11. The results indicate that the impact of BGWO–GWO on the computational cost is model-dependent. Some of the models, such as DT and GB models, show gains in both training time and inference latency when optimized. Other models, however, suffer from higher computational costs with the optimized configuration. For the model that shows the best predictive performance (RF), the optimization process increases the training time from 1.164 s to 1.658 s, the inference latency from 0.380 ms to 0.597 ms, alongside an increase in memory consumption. This is a computational compromise when we want to improve the prediction accuracy.

Moreover, the feature selection using BGWO and hyperparameter optimization using GWO take place offline. This means that they have no direct impact on the computational cost of each online scheduling request. Once trained, the chosen model is deployed for the prediction phase. These measurements serve as an experimental complement to the theoretical complexity analysis, confirming that the proposed approach should be assessed based on both predictive performance and computational expense. In the practical O-RAN, the offline optimization can be performed at a fixed frequency at the Non-RT RIC or the SMO layer, while the trained model is utilized for online inference. This provides computation-light scheduling in the real-time domain and facilitates viable deployments under the latency-critical conditions of 5G network.

Table 11 Computational cost of baseline and BGWO–GWO optimized models

Model	Baseline training time (s)	Improved training time (s)	Baseline RAM (MB)	Improved RAM (MB)	Baseline CPU (%)	Improved CPU (%)	Baseline latency (ms)	Improved latency (ms)
Linear regression	0.00325	0.00295	0.179	0.186	31.1	278.3	0.00118	0.00170
Polynomial regression	0.00440	0.00592	0.103	0.127	279.6	277.0	0.00665	0.00414
DT	0.01257	0.00670	0.025	0.028	307.3	403.6	0.00300	0.00145
GB	0.39932	0.35093	0.294	0.327	157.2	176.1	0.00834	0.00578
RF	1.16397	1.65805	0.567	0.657	111.0	105.9	0.38025	0.59671
SVM	0.00611	0.83638	0.155	0.185	0.0	97.8	0.00740	0.04762
KNN	0.00237	0.00238	0.013	0.167	31.1	0.0	0.01109	0.00840
Neural networks (100 epochs)	0.23131	0.16930	1.050	0.666	389.0	383.9	0.00369	0.00335
Neural networks (200 epochs)	0.24482	0.16128	1.050	0.394	393.5	375.7	0.00432	0.00392

6.3. Green AI considerations

The proposed framework aims to enhance the efficiency of network resource allocation, but the energy usage of the AI models themselves needs to be taken into account as well. The computing power needed for training and inference is evaluated in future applications. Furthermore, lightweight models, model compression, and optimization strategies can help to minimize computational costs without compromising predictive performance.

7. Threats to Validity and Reviewer-Ready Improvements

Although the proposed BGWO-GWO framework achieves high accuracy, several factors need to be considered for its practical deployment. First, the initial evaluation is performed on a public telemetry dataset that is made available by Nokia for 5G networks; therefore, it requires further validated in a wider network environment, such as multiple vendors, to ensure its scalability. Second, the framework must be tested in the real-time O-RAN test environment to evaluate the conditions in which data is not leaked, and to measure CPU overload and real-time processing latency under realistic conditions.

8. Conclusion and Future Work

This paper proposes a hybrid optimization algorithm (BGWO-GWO) to allocate radio resources intelligently in 5G networks. This framework combines two algorithms: the BGWO feature selection algorithm and GWO hyperparameter tuning algorithm, which balance the prediction accuracy and computational complexity to meet the requirements for edge deployment. A pre-processed public 5G network dataset was used to evaluate nine regression models, while strict train/validation/test separation, feature selection and data augmentation were applied to avoid data leakage. The main conclusions of this study are summarized as follows:

(1) The optimized DT model achieved the best overall balance between prediction accuracy and computational efficiency for near real-time edge deployment (R^2 = 99.44% and MSE = 0.00005).
(2) BGWO reduced the four candidate features to three selected features: Required Bandwidth, App encoded, and sig_lat thereby streamlining the feature space while retaining the most informative predictors.
(3) Continuous GWO optimization improved the MSE and R^2 for all nine evaluated models, confirming that improving the quality of model characteristics and parameters contributes more to predictive performance than increasing model complexity.
(4) The proposed framework integrates with the O-RAN architecture, where BGWO and GWO optimizations are performed offline at the non-real-time RIC layer, while only the trained lightweight predictor is used for real-time inference.
(5) Although the framework achieved excellent results on the evaluated Nokia dataset, its generalizability to other network environments still needs to be verified.

Future work will focus on validating the proposed framework on larger, heterogeneous, multi-vendor 5G datasets, incorporating time-varying traffic characteristics to evaluate real-time scalability and robustness under dynamic network conditions.

Conflicts of Interest

The authors declare no conflict of interest.

Reference

[1] Jyoti, Amandeep, and D. Kumar, "Investigating Resource Allocation Techniques and Key Performance Indicators (Kpis) For 5g New Radio Networks: A Review," International Journal of Computer Networks and Applications, vol. 10, no. 3, 2023.
[2] N. R. Pindi and F. J. Velez, "Traffic Scheduling and Resource Allocation for Heterogeneous Services in 5g New Radio Networks: A Scoping Review," Smart Cities, vol. 8, no. 5, article no. 168, 2025.
[3] W. K. Abdulwahab and S. M. Hadi, "Low Complexity High Throughput Low Density Parity Check Code Based on Compromised Iteration Over 5g Out Door Channel," Proceedings of Engineering and Technology Innovation, vol. 30, pp. 53-65, 2025.
[4] A. Yousuf and N. Kumar, "5G Resource Allocation for Efficient Usage of Bandwidth Using Machine Learning," International Journal for Research in Applied Science & Engineering Technology, vol. 12, no. 6, pp. 2408-2421, 2024.
[5] P. Ramesh and P. T. V. Bhuvaneswari, "Machine Learning Driven Clustering for Silhouetting 5G Network Throughput," Scientific Reports, vol. 16, article no. 10583, 2026.
[6] M. Kamruzzaman, N. I. Sarkar, and J. Gutierrez, "Machine Learning-Based Resource Allocation Algorithm to Mitigate Interference in D2D-Enabled Cellular Networks," Future Internet, vol. 16, no. 11, article no. 408, 2024.
[7] K. J. Rustamov, "5G-Enabled Internet of Things: Latency Optimization through AI-Assisted Network Slicing," Qubahan Techno Journal, vol. 2, no. 1, pp. 1-10, 2023.
[8] A.-I. Manolopoulos, V.-M. Alevizaki, M. Anastassopoulos, and A. Tzanakaki, "An AI-Assisted Framework for Lifecycle Management of Beyond 5G Services," IEEE Access, vol. 12, pp. 179463-179477, 2024.

[9] I. Konstantoulas, I. Loi, D. Tsimas, K. Sgarbas, A. Gkamas, and C. Bouras, "A Framework for User Traffic Prediction and Resource Allocation in 5G Networks," Applied Sciences, vol. 15, no 13, article no. 7603, 2025.

[10] Z. Z. Saleh, M. F. Abbod, and R. Nilavalan, "Intelligent Resource Allocation via Hybrid Reinforcement Learning in 5G Network Slicing," IEEE Access, vol. 13, pp. 47440-47458, 2025.

[11] H. E. Benmadani, M. Azni, T. E. Alharbi, M. S. Alzaidi, and M. Tounsi, "Deep Reinforcement Learning-Based Dynamic Scheduling for Real-Time Applications in LTE and RAN Slicing for eMBB in 5G," IEEE Access, vol. 13, pp. 33555-33570, 2025.

[12] P. Ramesh, P. T. V. Bhuvaneswari, V. S. Dhanushree, G. Gokul, and S. Sahana, "User Association-Based Load Balancing Using Reinforcement Learning in 5G Heterogeneous Networks," The Journal of Supercomputing, vol. 81, article no. 328, 2025.

[13] A. H. Abbas, "A Reinforcement Learning-Driven Scheduler for Minimizing Uplink Delay in 5G Networks," Journal of Al-Qadisiyah for Computer Science and Mathematics, vol. 17, no. 4, pp. 274-284, 2025.

[14] G. Kougioumtzidis, V. K. Poulkov, P. I. Lazaridis, and Z. D. Zaharis, "Deep Reinforcement Learning-Based Resource Allocation for QoE Enhancement in Wireless VR Communications," IEEE Access, vol. 13, pp. 25045-25058, 2025.

[15] Z. Yu, F. Gu, H. Liu, and Y. Lai, "5G Multi-Slices Bi-Level Resource Allocation by Reinforcement Learning," Mathematics, vol. 11, no. 3, article no. 760, 2023.

[16] M. A. Hady, S. Hu, M. Pratama, Z. Cao, and R. Kowalczyk, "Multi-Agent Reinforcement Learning for Resources Allocation Optimization: A Survey," Artificial Intelligence Review, vol. 58, article no. 354, 2025.

[17] B. G. Padmageetha, P. K. Naik, and M. Patil, "Resource Allocation in 5G Networks—Machine Learning Approach," Journal of Electrical Systems, vol. 20, no. 3, pp. 4165-4179, 2024.

[18] A. A. Alhussan and S. K. Towfek, "5G Resource Allocation Using Feature Selection and Greylag Goose Optimization Algorithm," Computers, Materials & Continua, vol. 80, no. 1, pp. 1179–1201, 2024.

[19] X. Yao and A. P. Yuste, "An ML-Based Resource Allocation Scheme for Energy Optimization in 5G NR," Sensors, vol. 25, no. 16, article. no. 4978, 2025.

[20] C. Wongoutong, "The Impact of Neglecting Feature Scaling in k-Means Clustering," PLoS ONE, vol. 19, no. 12, article no. e0310839, 2024.

Copyright© by the authors. Licensee TAETI, Taiwan. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY-NC) license (http://creativecommons.org/licenses/by/4.0/).