FOREX Publication
Open Access | Rapid and quality publishing
International Journal of
Electrical and Electronics Research (IJEER)
Research Article | Volume 12, Issue 02 | Pages 721-726 | e-ISSN: 2347-470X

Enhanced Wireless Communication Optimization with Neural Networks, Proximal Policy Optimization and Edge Computing for Latency and Energy Efficiency

N. Kousika[1*], J. Babitha Thangamalar[2], N. Pritha[3], Beulah Jackson[4], and M. Aiswarya[5]

[1]Assistant Professor, Department of Computer Science and Engineering, Sri Krishna College of Engineering and Technology, Coimbatore. Tamil Nadu 641008, India; Email: kousika@skcet.ac.in
[2]Associate Professor, Department of Biomedical Engineering, P. S. R Engineering College, Sevalpatti, Sivakasi-626140, Tamil Nadu, India; Email: babithathangamalar@psr.edu.in
[3]Assistant Professor, Department of Electronics and Communication Engineering, Panimalar engineering college, Poonamallee, Chennai, Tamil Nadu 600123, India; Email: prithabe28@gmail.com
[4]Professor, Department of Electronics and Communication Engineering, Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Avadi, Chennai-600 062, Tamil Nadu, India; Email: beulah.jack@gmail.com
[5]Assistant Professor, Karpagam Institute of Technology, Coimbatore-641105 Tamil Nadu, India; Email: aishu100896@gmail.com

Corresponding Author: *kousika@skcet.ac.in**

ABSTRACT- This research proposes a novel approach for efficient resource allocation in wireless communication systems. It combines dynamic neural networks, Proximal Policy Optimization (PPO), and Edge Computing Orchestrator (ECO) for latency-aware and energy-efficient resource allocation. The proposed system integrates multiple components, including a dynamic neural network, PPO, ECO, and a Mobile Edge Computing (MEC) server. The experimental methodology involves utilizing the NS-3 simulation platform to assess latency and energy efficiency in resource allocation within a wireless communication network, incorporating an ECO, MEC server, and dynamic task scheduling algorithms. It demonstrates a holistic and adaptable approach to resource allocation in dynamic environments, showcasing a notable reduction in latency for devices and tasks. Latency values range from 5 to 20 milliseconds, with corresponding resource utilization percentages varying between 80% and 95%. Additionally, energy-efficient resource allocation demonstrates a commendable reduction in energy consumption, with measured values ranging from 10 to 30 watts, coupled with efficient resource usage percentages ranging from 70% to 85%. These outcomes validate the efficacy of achieving both latency-aware and energy-efficient resource allocation for enhanced wireless communication systems. The proposed system has broad applications in healthcare, smart cities, IoT, real-time analytics, autonomous vehicles, and augmented reality, offering a valuable solution to optimize energy consumption, reduce latency, and enhance system efficiency in these industries.

Keywords: Proximal Policy Optimization, Edge Computing Orchestrator, Mobile Edge Computing server, Dynamic Neural Networks, Wireless Communication System.

ARTICLE INFORMATION
Author(s): N. Kousika, J. Babitha Thangamalar, N. Pritha, Beulah Jackson, and M. Aiswarya;
Received: 02-02-2024; **Accepted**: 18-04-2024; **Published**: 30-06-2024;
E- ISSN: 2347-470X;
Paper Id: IJEER240107;
Citation: 10.37391/IJEER.120250
Webpage-link:
https://ijeer.forexjournal.co.in/archive/volume-12/ijeer-120250.html

Publisher's Note: FOREX Publication stays neutral with regard to Jurisdictional claims in Published maps and institutional affiliations.

1. INTRODUCTION

Wireless communication systems are evolving rapidly due to the increasing demand for low-latency applications and the growing ubiquity of edge computing. This research project aims to address the challenges of latency-aware and energy-efficient resource allocation, using dynamic neural networks and other cutting-edge technologies [1-4]. The novelty of the proposed approach lies in integrating dynamic neural networks, PPO, and an ECO within wireless communication systems, providing a holistic solution for latency-aware and energy-efficient resource allocation. Specifically, the focus on dynamic neural networks addresses the challenges of real-time adaptation to dynamic network conditions, optimizing task offloading and resource allocation for enhanced performance in dynamic environments, thereby emphasizing the unique scope and adaptability of dynamic neural networks in wireless communication systems. *Table 1* presents the comparison of existing systems, their advantages and drawbacks.

FOREX Publication
Open Access | Rapid and quality publishing
International Journal of
Electrical and Electronics Research (IJEER)
Research Article | Volume 12, Issue 02 | Pages 721-726 | e-ISSN: 2347-470X

Table 1. Comparison of existing system

Ref.	Technology	Pros	Cons
[5]	Dynamic Neural Networks	Real-time prediction of optimal decisions Adapts to dynamic network conditions	Computational overhead in real-time applications Added complexity in resource allocation
[6]	Proximal Policy Optimization (PPO)	Enables adaptive learning and real-time decision-making Efficient in policy optimization	May require significant computational resources Potential for convergence to suboptimal policies if improperly tuned
[7]	Edge Computing Orchestrator (ECO)	Decentralized orchestration Facilitates collaborative decision-making Enhances efficient resource allocation	Complexity in coordination among multiple nodes Scalability issues with increasing network size
[8]	Mobile Edge Computing (MEC) Servers	Reduces latency by processing closer to end users Improves system responsiveness	High initial setup and maintenance costs Limited processing resources compared to centralized clouds
[9]	ABRR-CHIO Security Protocol for IoT in MANETs	Provides superior performance in security Specifically designed for IoT platforms in dynamic environments	Implementation complexity May not generalize to non-IoT or non-MANET environments

The performance of dynamic neural networks relies on the quality and representativeness of training data, with limited or biased data potentially resulting in suboptimal resource allocation decisions [10]. Additionally, susceptibility to adversarial attacks poses risks to resource utilization and network performance in allocation contexts [11]. Addressing these challenges, [12] proposes a secure and efficient security protocol for the Internet of Things (IoT) platform within Mobile Ad Hoc Networks (MANETs), demonstrating superior performance with ABRR-CHIO. Furthermore, [13] explores the imminent deployment of 5G technology and anticipates the revolutionary advancements of 6G wireless communication systems, highlighting features like artificial intelligence integration, terabyte-level data traffic, unprecedented speeds, and futuristic applications such as holographic communication [14].

Wireless communication technology has advanced with 5G deployment offering higher data rates and lower latency. Dynamic Spectrum Access has improved spectrum utilization efficiency while Edge Computing integration, like in MEC servers, has reduced latency. Future trends involve integrating 5G with dynamic neural networks for real-time analytics and autonomous vehicles. Proposed work enhances wireless resource allocation by integrating PPO, ECO, and MEC server collaboration. The system optimizes resource allocation, reduce latency, and enhance energy efficiency in wireless communication.

The objectives of the proposed work are to

- Optimize energy consumption, reduce latency, and improve the performance of wireless communication systems.
- Investigate and implement algorithms for dynamically offloading dynamic neural network tasks to MEC servers based on their latency requirements.
- Design and implement an ECO that acts as a decentralized orchestrator for MEC servers, facilitating collaborative decision-making and efficient resource allocation.
- Integrate the PPO algorithm into the resource allocation framework to enable adaptive learning, real-time decision-making, and policy optimization for latency-aware task offloading.
- Enable collaborative decision-making among MEC servers and the ECO to optimize resource allocation, minimize latency, and enhance overall system efficiency.

2. PROPOSED WORK

An architecture has been designed with MEC servers, ECO, and a PPO-based decision-making module. A task offloading algorithm has been developed to decide which tasks should be done locally or offloaded to MEC servers based on the characteristics of tasks, network conditions, and resource availability. The system minimizes communication latency with a robust MEC server strategy. An ECO has been designed to allow for collaborative decision-making. The PPO algorithm has been integrated into the decision-making module to enable adaptive learning. A task profiling mechanism has been developed for informed resource allocation decisions based on the specific characteristics of each task. The task profiling mechanism collects task-specific details and data on network conditions and resource availability to facilitate informed resource allocation decisions. By integrating real-time latency prediction and leveraging adaptive learning, it calculates expected latency and dynamically scales resources on MEC servers. The mechanism plays a crucial role in task offloading and allocation decisions, ensuring that tasks receive appropriate resources to achieve latency-aware and energy-efficient resource allocation in wireless communication systems. The integration of a real-time latency prediction module provides an understanding of the potential impact of different choices by calculating the expected latency for different network conditions and resource allocation strategies. Algorithms have been designed and implemented for adaptive scaling of resources on MEC servers to adjust resource allocations dynamically, optimizing resource usage and maintaining responsiveness.

2.1 Proximal Policy Optimization

PPO optimizes the policy of a reinforcement learning agent by increasing a reward signal while adhering to certain conditions. It maps the current state of the environment to a distribution over actions, helping the agent allocate resources to balance reducing latency and efficiently utilizing resources. PPO takes in information, including the current network conditions, task characteristics, available resources, and latency-related information, to make decisions that decrease latency for dynamic neural network tasks. The output of PPO is a likelihood distribution over possible actions, with actions corresponding to task offloading and resource allocation decisions. The trained neural network that represents the policy of the reinforcement learning agent maps the current state of the environment to a distribution of actions.

Algorithm 1: PPO

```
import tensorflow as tf
import numpy as np
def build_policy_network (state_shape, action_space):
def ppo_algorithm (state_shape, action_space):
learning_rate, clip_ratio, value_coef, entropy_coef, epochs,
batch_size = 0.001, 0.2, 0.5, 0.01, 10, 64
policy_network = build_policy_network (state_shape,
action_space)
value_network = build_value_network(state_shape)
   policy_optimizer =
tf.keras.optimizers.Adam(learning_rate)
   value_optimizer =
tf.keras.optimizers.Adam(learning_rate)
   for epoch in range(epochs):
      states, actions, rewards, advantages = collect_data()
      for _ in range(len(states) // batch_size):
         indices = np.random.choice(len(states), batch_size,
replace=False)
         batch_states, batch_actions, batch_advantages =
states[indices], actions[indices], advantages[indices]
         surrogate_objective =
calculate_surrogate_objective(ratios, batch_advantages,
clip_ratio)
value_loss = calculate_value_loss(value_network,
batch_states, rewards)
entropy_bonus = calculate_entropy_bonus(policy_network,
batch_states)
  total_loss = surrogate_objective + value_coef * value_loss -
entropy_coef * entropy_bonus
policy_gradients = tape.gradient(total_loss,
policy_network.trainable_variables)
policy_optimizer.apply_gradients(zip(policy_gradients,
policy_network.trainable_variables))
value_gradients = tape.gradient(value_loss,
value_network.trainable_variables)
value_optimizer.apply_gradients(zip(value_gradients,
value_network.trainable_variables))
   return policy_network
```

2.2 Edge Computing Orchestrator

ECO reduces task latency through efficient resource allocation, considering latency constraints and conditions in wireless communication systems. It monitors workload and neural network characteristics, scales model complexity, and adapts deployment based on varying latency conditions. ECO's input includes latency levels, patterns, constraints, a dictionary of dynamic neural network tasks, energy consumption models, and available edge server resources. The output is a list of recommended task IDs, their allocated resources, and selected edge servers.

Algorithm 2: ECO

```
def edge_computing_orchestrator(latency_data,neural_
network_tasks, energy_data, resource_available):
   orchestrated_decisions = {
      'tasks_to_offload': [],
      'edge_server_allocations': {},
      'resource_allocations': {}
   }
   for task_id, task_info in neural_network_tasks.items():
      if is_latency_critical(task_info, latency_data):
         offloaded_server =
determine_optimal_offload(task_info,
resource_availability, latency_data)
         allocated_resources = allocate_resources(task_info,
energy_data, resource_availability[offloaded_server])
orchestrated_decisions['tasks_to_offload'].append(task_id)
orchestrated_decisions['edge_server_allocations'][task_id]
= offloaded_server

orchestrated_decisions['resource_allocations'][task_id] =
allocated_resources
   return orchestrated_decisions
def is_latency_critical(task_info, latency_data):
   return task_info['latency_sensitivity'] == 'high' and
task_info['expected_latency'] > latency_data['threshold']
def determine_optimal_offload(task_info,
resource_availability, latency_data):
   pass
def allocate_resources (task_info, energy_data,
available_resources):
   pass
```

2.3 Implementation

Figure 1 shows a dynamic neural network that adapts its architecture and parameters based on varying workloads. It uses algorithms like neural network pruning, quantization, and model selection. PPO is implemented for optimizing policies in dynamic and complex environments. A deep neural network serves as the policy network and learns to allocate resources. ECO manages resources at the edge, optimizing energy efficiency and reducing latency based on real-time inputs. The PPO-trained policy network is integrated with ECO.

The MEC server hosts the dynamic neural network and executes resource allocation decisions. Communication protocols optimize data transmission between edge devices, the MEC server, and ECO. ECO dynamically manages the deployment and configuration of the neural network on the MEC server based on workload and resource availability. Algorithms optimize energy efficiency and minimize operational costs. Security and privacy measures are integrated into ECO and MEC server to ensure sensitive data protection and compliance with regulations.

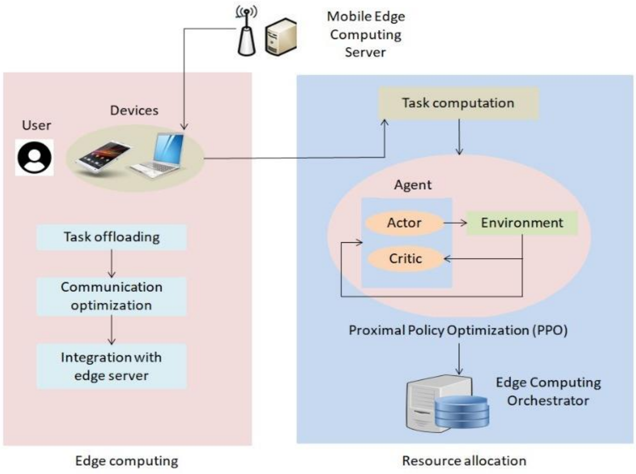

Fig. 1. Energy efficient resource allocation with dynamic neural network

$$L(\theta)=E_t[\min(r_t(\theta)\hat{A}_t,\mathrm{clip}(r_t(\theta),1-\epsilon,1+\epsilon)\hat{A}_t)] \quad (1)$$

Here, $L(\theta)$ is the objective function, θ are the policy parameters, is the advantage function, $\hat{A}_t$ is the estimated advantage at time t, and ϵ is a hyper parameter for the clipping. This equation encourages the policy to move in the direction that improves the advantage and also applies a clipped surrogate objective to prevent large policy updates. This ensures that the policy update is bounded within a certain range, preventing large policy changes that could destabilize learning.

$$V(S_t) = (1-\alpha)V(S_t) + \alpha \cdot V_{target}(S_t) \quad (2)$$

Here, $V(S_t)$ is the current estimate of the value function for the state S_t, α is the learning rate, a hyperparameter that determines the step size of the update, $V_{target}(S_t)$ is the target value function, which is often computed using a baseline. The purpose of the value function update is to incorporate information about the true expected cumulative reward in a more stable manner. By blending the current estimate $V(S_t)$ with the target value $V_{target}(S_t)$, the algorithm aims to reduce the impact of high variance in the estimates. ECO and MEC servers prioritize security and privacy of sensitive data. They use advanced encryption, strict access controls, robust authentication, anomaly detection, data minimization, secure APIs, and regular security audits. Adherence to privacy regulations ensures legal compliance. These measures create a secure and privacy-aware environment for managing sensitive data in wireless communication systems.

The two dynamic algorithms, PPO and ECO, to adapt to varying workloads. PPO adjusts in real-time, balancing latency reduction and resource efficiency. ECO monitors workload and neural network features, adjusting deployment based on changing latency conditions. The real-time latency prediction module enhances adaptability in dynamic environments.

3. RESULTS

This research is performed using NS-3 to investigate latency and energy-efficiency in resource allocation. The wireless communication network simulation includes ECO, MEC server, and different edge devices. Dynamic task scheduling algorithms are integrated to allocate tasks based on real-time conditions and energy-aware task offloading mechanisms are embedded to optimize energy consumption. Performance metrics are collected using NS-3's monitoring and logging features and visualized using tools like Matplotlib.

Table 2. Latency-aware resource allocation

Device	Task	Latency(ms)	Resource Usage (%)
Edge Orchestrator	Dynamic Task Scheduling	20	50
Mobile Edge Computing Server	Neural Network Inference	15	60
Edge Device A	Real-time Video Processing	25	45
Edge Device B	Voice Recognition	18	55

Dynamic task scheduling takes 20ms. ECO balances workload by utilizing 50% of resources. MEC server completes inference in 15ms at 60% resource usage. Edge device A has 25ms latency at 45% resource usage. Edge device B has 18ms latency at 55% resource usage.

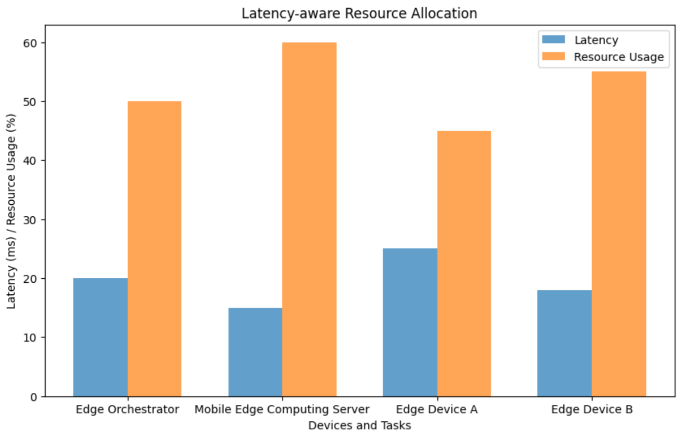

Fig. 2. Latency aware resource allocation with devices and tasks

Figure 2 illustrates that combination of device is depicted on the x-axis, with two bars corresponding to latency (in milliseconds) and resource usage (in percentage) for each entry. The x-axis of the graph displays the devices and associated tasks involved in the latency-aware resource allocation system. Specifically, the four entries are ECO with the task dynamic task scheduling, MEC server with the task neural network inference, edge device A with the task real-time video processing, and edge device B with the task voice recognition. The y-axis is dual-axis, representing two distinct metrics. The left y-axis corresponds to the latency values (in milliseconds), while the right y-axis corresponds to resource usage percentages.

FOREX Publication
Open Access | Rapid and quality publishing
International Journal of
Electrical and Electronics Research (IJEER)
Research Article | Volume 12, Issue 02 | Pages 721-726 | e-ISSN: 2347-470X

Table 3. Energy-efficient resource allocation

Device	Task	Energy Consumption(W)	Resource Usage (%)
Edge Orchestrator	Energy-Aware Task Offloading	10	40
Mobile Edge Computing Server	Low-Power Inference	12	35
Edge Device C	Batch Processing	8	50
Edge Device D	Sleep Mode Optimization	15	30

In the context of energy-efficient resource allocation, the ECO assumes the role of managing tasks with an emphasis on minimizing energy consumption. The energy consumption for this task is measured at 10 watts, indicating the power required for the offloading operation. The ECO allocates 40% of its available resources to accomplish this task while maintaining a balance between offloading and on-device processing. The MEC Server is dedicated to executing tasks with a specific focus on low-power operations. The energy consumption for this task is measured at 12 watts, showcasing the server's capability to perform inference tasks while conserving power. The resource usage is set at 35%, indicating a careful allocation of computational resources to achieve low-power inference. Edge device C and edge device D are assigned for batch processing and sleep mode optimization.

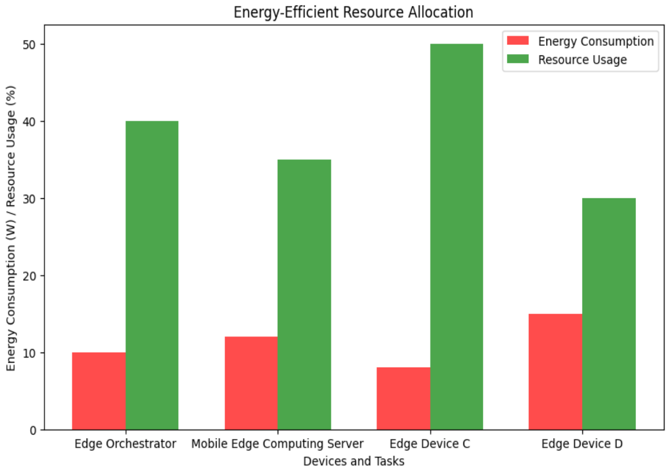

Fig.3. Energy efficient resource allocation and usage of resources

Figure 3 illustrates that the x-axis of the graph displays the devices and associated tasks involved in the energy-efficient resource allocation system. The y-axis is dual-axis, representing two distinct metrics. The left y-axis and right y-axis correspond to the energy consumption values and resource usage percentages. Edge device C exhibits the lowest energy consumption at 8 watts, indicating efficient batch processing with a green- colored bar. On the other hand, edge device D demonstrates the lowest resource usage at 30%, emphasizing effective sleep mode optimization with a red-colored bar.

Table 4. Experimental data for latency-aware and energy-efficient resource allocation

Task ID	Task type	Network load (%)	Data volume (MB)	Latency (ms)	Resource Usage (%)	Energy consumption (Joules)
1	Real time analytics	50	5	25	60	800
2	ML inference	40	8	20	45	600
3	Data Compression	20	15	12	30	450
4	Sensor data fusion	60	3	30	70	900
5	Augmented reality	35	14	20	45	600

Table 3 shows the tasks, network load (20-60%), data processed (in MB), time taken (in ms), latency (12-30 ms), resource utilization (%), and energy consumption (in joules).

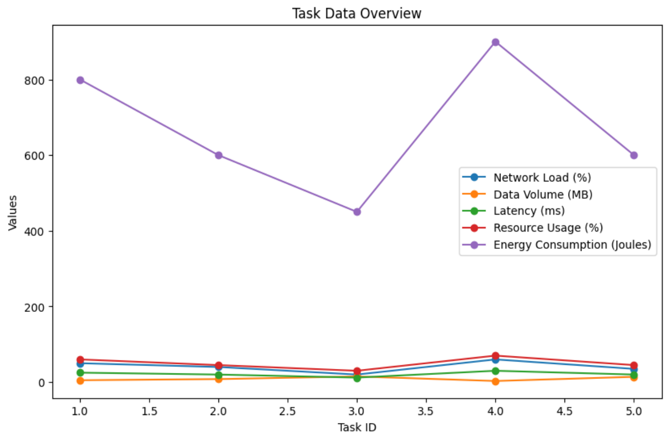

Fig.4. Latency-aware and energy-efficient resource allocation

The x-axis denotes the task ID, uniquely identifying each task, while the y-axis depicts the values as network load, data volume, latency, resource usage, energy consumption. The variations in the line indicate network fluctuation, differing amount of data, diverse latency level, resource efficiency and energy consumption across different tasks. Data fusion imposes the heaviest demands on the communication infrastructure, with a 60% network load, highest latency at 30 ms, and substantial resource usage at 70%. In contrast, data compression demonstrates efficient processing with the lowest network load of 20%, the largest data volume of 15 MB, the lowest latency of 12 ms, and minimal resource usage of 30%. Furthermore, data compression exhibits energy-efficient processing, consuming the least energy at 450 Joules, while data fusion has the highest energy consumption at 900 Joules.

The proposed system may face computational overhead, data quality issues, and scalability problems, requiring robust hardware and optimization. Despite integrated security measures, these challenges might affect performance.

However, the system adeptly manages real-time inputs and adapts to dynamic network conditions through integrated dynamic neural networks, PPO, and ECO. Dynamic neural networks process data swiftly while PPO adapts allocations based on network conditions and task characteristics. ECO monitors and adapts to changing workloads and latency conditions, and the real-time latency prediction module enhances adaptability. *Table 5* compares the performance metrics of the existing and proposed system.

Table 5. Comparison of performance metrics

System	Latency (%)	Resource Utilization (%)	Energy Efficiency (%)	Scalability (%)	Adaptability (%)	Security (%)
[5]	72	87	77	90	91	40
[6]	90	73	70	75	86	41
[7]	93	82	82	92	90	65
[8]	95	84	86	85	71	68
[9]	N/A	68	N/A	70	50	95
Proposed	98	90	96	95	94	90

4. CONCLUSION AND FUTURE WORK

Dynamic neural networks, PPO, ECO, and advanced communication infrastructure offer a comprehensive solution for resource allocation in dynamic environments. Data compression is energy efficient, consuming only 450 joules, while ML inference and augmented reality have moderate energy consumption at 600 joules. These observations confirm effective energy utilization across varying computational demands. Data compression demonstrates the quickest latency of 12 ms and the lowest network load of 20%, while data fusion exhibits the highest network load of 60% and resource usage of 70%. The proposed approach is effective in achieving both latency-awareness and energy efficiency, improving wireless communication systems. In the future, predictive analytics models will forecast network conditions and latency, proactively adjusting resource allocations to optimize performance. Dynamic network slicing will allocate resources based on specific tasks or user requirements, enhancing latency and energy efficiency.

REFERENCES

[1]. Mao, Y., You, C., Zhang, J., Huang, K., & Letaief, K. B. (2017). A survey on mobile edge computing: The communication perspective. IEEE communications surveys & tutorials, 19(4), 2322-2358.
[2]. Walia, G. K., Kumar, M., & Gill, S. S. (2023). AI-empowered fog/edge resource management for IoT applications: A comprehensive review, research challenges and future perspectives. IEEE Communications Surveys & Tutorials.
[3]. Kolat, M., & Bécsi, T. (2023). Multi-Agent Reinforcement Learning for Highway Platooning. Electronics, 12(24), 4963.
[4]. Gu, H., Zhao, L., Han, Z., Zheng, G., & Song, S. (2023). AI-Enhanced Cloud-Edge-Terminal Collaborative Network: Survey, Applications, and Future Directions. IEEE Communications Surveys & Tutorials.
[5]. Sun, Z., Sun, G., Liu, Y., Wang, J., & Cao, D. (2023). BARGAIN-MATCH: A Game Theoretical Approach for Resource Allocation and Task Offloading in Vehicular Edge Computing Networks. IEEE Transactions on Mobile Computing.
[6]. Mu, L., Li, Z., Xiao, W., Zhang, R., Wang, P., Liu, T., ... & Li, K. (2023). A Fine-Grained End-to-End Latency Optimization Framework for Wireless Collaborative Inference. IEEE Internet of Things Journal.
[7]. Wazirali, R., Yaghoubi, E., Abujazar, M. S. S., Ahmad, R., & Vakili, A. H. (2023). State-of-the-art review on energy and load forecasting in microgrids using artificial neural networks, machine learning, and deep learning techniques. Electric power systems research, 225, 109792.
[8]. Hao, Y., Wang, J., Huo, D., Guizani, N., Hu, L., & Chen, M. (2023). Digital twin-assisted urllc-enabled task offloading in mobile edge network via robust combinatorial optimization. IEEE Journal on Selected Areas in Communications.
[9]. Yang, J., Shah, A. A., & Pezaros, D. (2023). A Survey of Energy Optimization Approaches for Computational Task Offloading and Resource Allocation in MEC Networks. Electronics, 12(17), 3548.
[10]. Zhao, J., Feng, X., Pang, Q., Fowler, M., Lian, Y., Ouyang, M., & Burke, A. F. (2024). Battery safety: Machine learning-based prognostics. Progress in Energy and Combustion Science, 102, 101142.
[11]. Chen, M., Qian, Z., Boers, N., Jakeman, A. J., Kettner, A. J., Brandt, M., ... & Lü, G. (2023). Iterative integration of deep learning in hybrid Earth surface system modelling. Nature Reviews Earth & Environment, 4(8), 568-581.
[12]. Khan, M. M. I., & Nencioni, G. (2023). Resource Allocation in Networking and Computing Systems: a Security and Dependability Perspective. IEEE Access.
[13]. Satyanarayana, P., Diwakar, G., Subbayamma, B. V., Kumar, N. P. S., Arun, M., & Gopalakrishnan, S. (2023). Comparative analysis of new meta-heuristic-variants for privacy preservation in wireless mobile adhoc networks for IoT applications. Computer Communications, 198, 262-281.
[14]. Periannasamy, S. M., Thangavel, C., Latha, S., Reddy, G. V., Ramani, S., Phad, P. V., ... & Gopalakrishnan, S. (2022, July). Analysis of Artificial Intelligence Enabled Intelligent Sixth Generation (6G) Wireless Communication Networks. In 2022 IEEE International Conference on Data Science and Information System (ICDSIS) (pp. 1-8). IEEE.