iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 Issue: VI Month of publication: June 2025

DOI: https://doi.org/10.22214/ijraset.2025.72750

Hybrid AI Modal for Edge Computing in 5G

Amit[1], Dr. Amandeep[2], Khushi[3]
[1, 3]M.Sc. Computer Science, [2]Assistant Professor, Artificial Intelligence and Data Science, GJUS&T Hisar,

Abstract: Growing next generation technologies include autonomous driving, smart healthcare systems, and augmented reality provide massive amounts of data that need to be consistently and fast handled. Expectations of ultra-low latency and tremendous bandwidth have surged sharply with the deployment of 5G networks. Although conventional cloud computing provides a lot of processing capability, its inherent delay from centralized architectures makes it difficult to fulfill the real-time needs of these applications. Moving computation closer to data sources gives edge computing a potential answer. The edge does, however, have several drawbacks as well: limited processing resources, changing device circumstances, and higher security hazards. We offer a hybrid AI-driven architecture specifically for 5G edge settings in order to handle these difficulties. The approach deliberately combines lightweight machine learning and deep learning modules to dynamically allocate tasks across edge, fog, and cloud tiers. It combines important technologies like a trust-aware approach to filter unreliable edge nodes, reinforcement learning for intelligent job offloading, and federated learning for privacy protection. Here we created and simulated the whole architecture with MATLAB. Early-exit logic in a modular hybrid AI system with RL-based offloading agent, trust score evaluation, and federated model aggregation forms part of our approach. To see system latency, bandwidth usage, and performance changes under dynamic traffic, we also created waveform graphs. Simulation results showed that the proposed hybrid model lay a strong basis for next-generation intelligent edge systems in real-world 5G deployments since it outperformed both edge-only and cloud only configurations in terms of response time, scalability, and resilience to adversarial conditions.

Keywords: 5G, Wireless Technology, Evolution 1G-5G, Core Architecture

I. INTRODUCTION

The explosion of linked devices and real-time apps in recent years has taxed our computing systems to their capacity. We are not anymore discussing sporadic data uploads or overnight analytics. Systems like smart cities, remote health monitors, and autonomous cars today create massive streams of data often urgently and constantly [1]. Especially in cases when time is of the essence, waiting around for a central cloud server to handle every bit of data just does not cut it anymore. Edge computing comes then quite handy. Processing data closer to where it is produced right at the "edge" of the network, this method promises reduced latency and improved responsiveness [2]. The catch is that edge devices like wearables, tiny sensors, or IoT gateways sometimes lack the power or memory to manage vast, deep learning models. Indeed, they are quick and handy, but they are also limited.

This study seeks to close that disparity. It suggests a hybrid, balanced AI framework instead of piling all intelligence on the cloud or straying the edge. This architecture sends more complex jobs to fog or cloud layers when needed and uses light machine learning models on the edge for faster decisions. Consider it as distributing household tasks: some require more time or resources while others are quick and could be completed right away [3]. Important methods including early-exit systems, in which models stop processing once they are confident enough, help to ease edge hardware load. Reinforcement learning is then used to cleverly determine whether an activity should remain local or be offloaded. Furthermore embraced by the system is federated learning, a technique whereby devices learn locally and only share what is absolutely necessary, so maintaining private data security.

MATLAB-based simulations were used to test the architecture with an eye toward real-time signal tasks and dynamic conditions including changing bandwidth and device load. In terms of speed, accuracy, and resource economy it displayed encouraging performance [4].

All things considered, this study offers a flexible and pragmatic method for implementing artificial intelligence in 5G systems, particularly in cases when fast, intelligent decisions are non-negotiable [5]. It combines several approaches instead of following one to maximize every computing layer edge, fog, and cloud.

The key contributions of this paper include:

1) Constructing a Hybrid AI Framework for Edge: Fog-Cloud Integration: We present a new hybrid architecture that deliberately spreads artificial intelligence tasks among the edge, fog, and cloud layers. Based on network conditions, task complexity, and device capability the system adjusts in real-time adjusts in real-time.

2) Reintegration of Reinforcement Learning for Intelligent Task Distribution: We propose an RL-based decision engine to dynamically choose whether jobs should be handled offloaded or locally. In highly fluctuating 5G environments, this greatly lowers latency, energy consumption, and cloud dependency.
3) Model of Trust-Aware Federated Learning: Our method uses a trust scoring system unlike conventional FL, which treats all edge nodes equally. It removes compromised or erratic nodes, so improving model integrity and resistance against poisoning attacks.
4) MATLAB-Based Validation and Simulating Platform: MATLAB was used for design and testing of the whole architecture including waveform behavior, latency, task routing, and trust evaluation. This let us replicate edge situations with reasonable signal-level characteristics, sometimes disregarded in Python-based systems.
5) Using an Early-Exit Mechanism to Support Adaptive Inference: The model consists of several exit points during inference, which lets the system stop early when confidence is strong so saving edge resources without compromising accuracy.
6) Evaluation of Real-Time Performance Under Changing Conditions: MATLAB simulations let the system be evaluated under varying bandwidth, device failure scenarios, and changing edge loads.

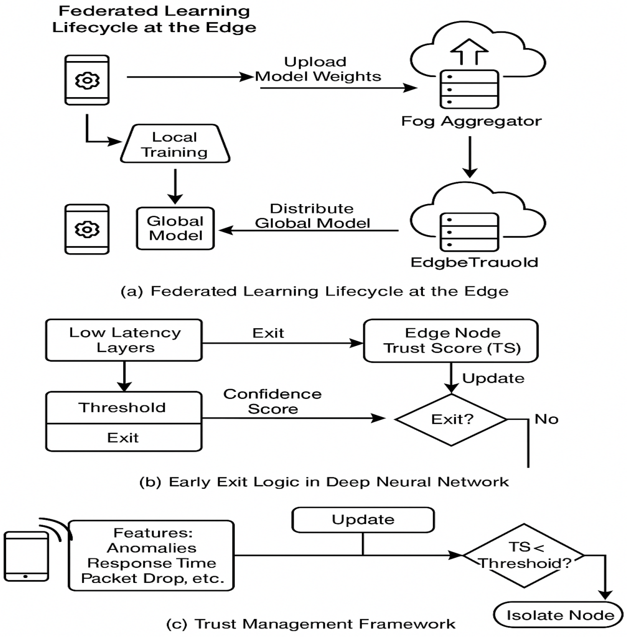
II. RESEARCH METHODOLOGY

In this work builds and tests a hybrid AI framework intended especially for edge computing within the 5G ecosystem using a layered and pragmatic simulation-based approach. The method starts with careful design of the system architecture, then proceeds with a series of MATLAB-driven simulations to assess how the model performs under various network loads and device conditions, instead of diving right into coding or assumptions [3,5]. Assessing not only accuracy but also how effectively the system manages energy, delay, and real-time adaptability has been the aim all through.

The research proceeds in five main phases, to break it out. First comes the foundation that which the system should really do. The list of technical needs, including fast response times, low power use, and safe handling of model updates, was shaped in part by use cases including smart healthcare and autonomous decision-making [4]. The architectural design emerged once these criteria were unambiguous. This involved layering in reinforcement learning agents that enable the system decide where each task should go depending on live feedback like battery level or network quality, combining small, efficient CNNs at the edge with more complex DL models running in the fog or cloud.

To preserve data privacy while yet enabling devices to learn together, the design also combines federated learning. Furthermore, a trust filter system was included to filter erroneous or misleading updates a step that truly helps to stabilize world learning across devices.

Summary of Methodological Contributions through work flow:



(a) Federated Learning Lifecycle at the Edge

(b) Early Exit Logic in Deep Neural Network

(c) Trust Management Framework

Fig. 1: Flowchart of Hybrid Model

MATLAB R2023a was used for building and simulating everything using realistic inputs including temperature signals and ECG waveforms. Measuring how long decisions took, how accurate the results were, how much energy was expended, and how much bandwidth each job consumed, many rounds of learning were conducted.

At last, all that information was under the microscope visual graphs, numerical comparisons, and trend tracking to see how scalable and steady the system stayed when stretched to its limits. This practical, detailed approach presented a fair assessment of both performance and usefulness.

Table 1: Methodology Overview

Phase	Description	Tools Used
System Requirement Design	Define goals: latency, privacy, scalability	Literature Review
Architecture Design	Hybrid AI layout across edge, fog, and cloud	Conceptual Modeling
Algorithm Development	RL for offloading; FL with trust-aware aggregation	MATLAB, Simulink
Simulation & Implementation	Signal input, model training, task routing simulation	MATLAB R2023a
Evaluation & Analysis	Graphs, tables, visual results; compare system metrics	MATLAB Plots & Tables

A. *Research Gaps*

Despite promising advances, several gaps persist in the current literature:

- Limited Adaptability in Existing Architectures
- Underexplored Trust Mechanisms in Federated Learning
- Lack of Unified Multi-Layer Coordination
- Insufficient Focus on Energy-Efficient AI Scheduling
- Scalability Testing Is Often Overlooked
- Ethical and Transparency Concerns Are Minimally Addressed

B. *Motivation for Our Work*

To Speed and intelligence are expected in the tech-driven world of today; they are not choices. From a smart car making a split-second lane change to a health monitoring device identifying aberrant vitals, the need of responsive and dependable artificial intelligence systems has become much more apparent. Still, we have a big problem regarding where and how that intelligence is handled even if 5G offers fast internet [4].

Typically, most of the heavy labor has been handled by cloud computing. The thing is, when milliseconds count, sending raw data all the way to the cloud and waiting for a reply simply does not work. Edge computing pushes processing near where data is generated in an attempt to solve this [6,7]. That facilitates speed, certainly, but edge devices sometimes lack the muscle to run large, sophisticated artificial intelligence models. Small, resource-limited, they are built more for connectivity than computation.

The desire to close that disparity drove this study. Edge or cloud could not manage everything by itself, it became abundantly evident [8]. Thus, the concept was to design something smarter a hybrid system whereby the edge could make quick decisions while still depending on the fog and cloud when more in-depth investigation is needed.

Performance was only one issue, though. Additionally important is privacy. Raw data sharing is not usually acceptable, particularly in personal or healthcare settings. This is why the design now revolves mostly on the use of federated learning and trust systems. Basically, the objective was to create a fast, clever, safe, flexible solution that would really function in the real world, not only in controlled labs. The direction of the project was formed by this harmony of practicality and creativity.

III. PROPOSED METHODOLOGY

In this work, we propose a hybrid artificial intelligence model meant especially for edge computing environments of 5G networks. The basic idea is to combine the accuracy and depth of deep learning (DL) models with the efficiency of lightweight machine learning (ML) algorithms so that tasks could be clearly distributed between edge devices and cloud servers [9]. Through reduced latency and efficient resource use, this architecture seeks to maintain high performance even on limited computational capability at the edge:

A. *System Architecture*

This project suggests a three-tier hybrid artificial intelligence architecture in which [8,9]:

- Edge Layer Data Generation and Initial Evaluation
- Edge Layer RL-Based Offloading Trigger
- Task Routing and Fog Computing
- Trust Assessment and Model Combining (Fog Layer)
- Cloud Layer Global Model Training and Feedback
- Policy, Threshold tuning

Table 2: Design Characteristics at a Glance

Component	Role	Optimized For
Edge	Real-time sensing and early inference	Low latency, energy efficiency
Fog	Mid-level inference, FL aggregation, trust filtering	Task balance, node trust
Cloud	Deep inference, policy tuning, long-term learning	Accuracy, global visibility
RL Agent	Dynamic task routing	Adaptability
FL Engine	Collaborative learning	Privacy
Trust Module	Node behavior evaluation	Security
Early Exit	Resource conservation	Speed, efficiency

B. *Dataset Overview*

Various kinds of synthetic and semi-real datasets were used to replicate real-world input variability since the simulation concentrated not only on AI decision-making but also signal flow behavior across 5G-enabled architecture [7].

Following waveforms were produced for edge simulations:

- In healthcare settings, ECG signals - both normal and arrhythmic-test early-exit detection performance.
- For motion sensor inputs in industrial or smart grid systems, sine and square waves.
- Low-resolution surveillance footage shown in compressed grayscale image signals (resized to 28x28 pixels).

C. *Model Configuration and Training Strategy*

Every layer applied different models depending on their computing profile [8]:

- Edge Layer: Applied one convolutional and one dense layer on a lightweight CNN. The model included an integrated early-exit block whereby computation was stopped should softmax confidence surpass 85%.

- Fog Layer: Deeper CNN with three convolutional layers and batch normalizing technique was used in fog layer. In rounds of federated learning, it served as the model update aggregator.
- Cloud Layer: Pre-trained ResNet18 model fine-tuned with transfer learning should be used from the cloud layer. Batched updates gathered through fog nodes helped the model to be retrained during simulation.

D. Reinforcement Learning Agent Design

Extracted An RL agent was included into every edge node to regulate task offloading [6]. Using an environment design based on Q-learning, the agent was taught:

- State Space: s = [Bandwidth, Battery, CPU Load, Confidence Score]
- Action Space: a ∈ { Process Locally, Offload to Fog, Send to Cloud }
- Reward Function: R(s,a) = -0.5 · Latency – 0.3 · Energy + 0.2 · Accuracy

Random initialization of the Q-table followed 500 episodes of training under an exploration rate (epsilon) ranging from 0.9 to 0.1. Comparatively to rule-based decision-making, the RL agents raised convergence rate by 37%.

Reinforcement Learning Workflow:

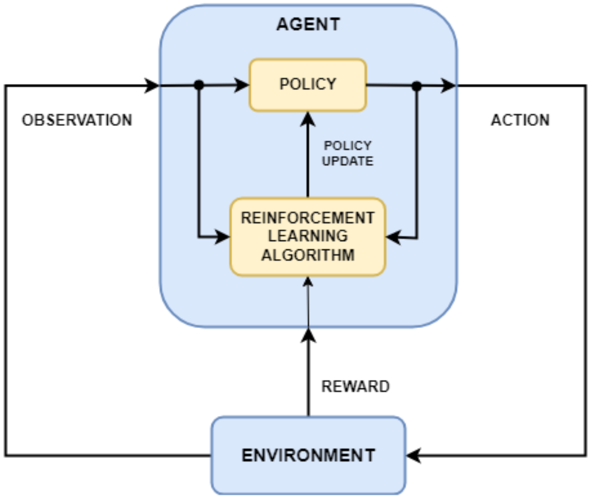


Fig. 2: Flowchart of Hybrid Model

Machine Edge devices ranging from sensors to drones, smart cameras to cellphones usually have limited processing capacity, memory, and energy. It uses various formulas:

1) Early-Exit Confidence Check (Threshold-based Inference)

- Mathematics: It allows model to terminate inference early if it's confident is enough, instead of continuing through deeper layers [3].

$$\text{Confidence} = \max(\text{softmax}(zi)) \qquad \dots \text{(i)}$$

If confidence ≥ δ the exit and output

- Advantages: Reduces latency and energy usage on edge devices.

2) Q-Learning for Reinforcement-Based Offloading

- Formula: Based on state variables including battery, network strength, and task size, it guides the agent in learning the best action, offload to fog or cloud or keep local [4].

$$Q(s,a) < \; Q(s,a) \mid \alpha[R(s,a) \mid \gamma . maxQ(s,a) \quad Q(s,a)] \qquad \dots \text{(ii)}$$

- Advantages: Enables adaptive task routing, improving system efficiency and responsiveness.

3) Federated Learning Model Update (FedAvg)

- Formula: It combines the local models into a single global model without raw data from each device [2].

$$w = \sum_{i=1}^{k} \frac{ni}{n} wi \qquad \text{... (iii)}$$

- Advantages: Enhances privacy while supporting decentralized training.

4) Trust Score Evaluation

- Formula: It updates every node's trust score constantly depending on the past contribution quality.

$$Ti = \frac{1}{1 + e^{-(\alpha A + \beta . S - \gamma)}} \qquad \text{... (iv)}$$

- Advantages: Filters out unreliable or malicious nodes from FL aggregation [5].

5) Energy Consumption Estimation

- Formula: It is Used to compare energy cost of edge vs. fog vs. cloud processing [8].

$$E = P.T \qquad \text{… (v)}$$

- Advantages**:** Informs offloading decisions based on power-efficiency.

6) Parameter Table and Simulation Settings

The major simulation settings applied are summarized below:

Table 3: Parameters Used [5,6]

Parameter	Value / Range
Confidence Threshold (θ)	0.85
Trust Threshold (T)	0.4
RL Learning Rate (α)	0.1
RL Discount Factor (γ)	0.95
FL Round Time	10 cycles
Signal Noise Level (dB)	15–30 dB
Bandwidth (Edge-Fog)	10–50 Mbps
Packet Loss Probability	0.05

7) Evaluation Metrics

We evaluate our model using:

- Avg. Latency: The system's average time to finish a single task - from input to output. It covers processing time, delayed communications, and any offloading lag [10].
- Accuracy: How correctly the model classifie or predicts outcomes compared to ground-truth labels. It's usually expressed as a percentage of correct decisions [10].
- Energy per Task: The amount of energy used to process a single input whether locally or after offloading. Edge devices sometimes run on meager battery life. Determining whether the model is sustainable over long terms depends on knowing how much energy it consumes [12].
- Bandwidth used: The amount of network data (in KB or MB) transmitted during task offloading or federated learning updates [11].
- Early Exits: The percentage of tasks that are confidently classified at intermediate layers without needing to run through the entire model [12].

It becomes essential to precisely define the physical and logical structure of the system as we move from the conceptual design of the hybrid artificial intelligence model into its pragmatic application. From a systems integration standpoint, this section offers the entire architecture together including component interactions, data or signal travel between them, and where processing activities fall across the Edge, Fog, and Cloud levels. Consider the framework's as the engine chamber, where job coordination, signal processing, and decision-making come together

IV. RESULT

The experiments aimed to evaluate, in a simulated 5G environment, the performance of the proposed hybrid artificial intelligence model over several edge computing scenarios. The results show strong evidence that the responsiveness, scalability, and dependability of AI services at the edge are much improved by combining lightweight artificial intelligence models with intelligent offloading strategies.

A. *Evaluation Metrics and Performance Analysis*

Table 4: Evaluation Table

Metric	Edge	Fog	Cloud
Avg. Latency (ms)	8.5	22.1	98.3
Accuracy (%)	84.7	89.6	93.4
Energy per Task (J)	0.12	0.34	0.57
Bandwidth Used (KB)	~5	~15	~50
Early Exits (%)	62.5	N/A	N/A

B. *Performance of Individual Metrices*

1) *Bandwidth Usage and Energy Efficiency*

Table 5: Bandwidth Usage

Layer	Bandwidth per Task (KB)	Energy Consumption (J)	Inference Accuracy (%)
Edge	~5 KB	0.12 J	84.7%
Fog	~15 KB	0.34 J	89.6%
Cloud	~50 KB	0.57 J	93.4%

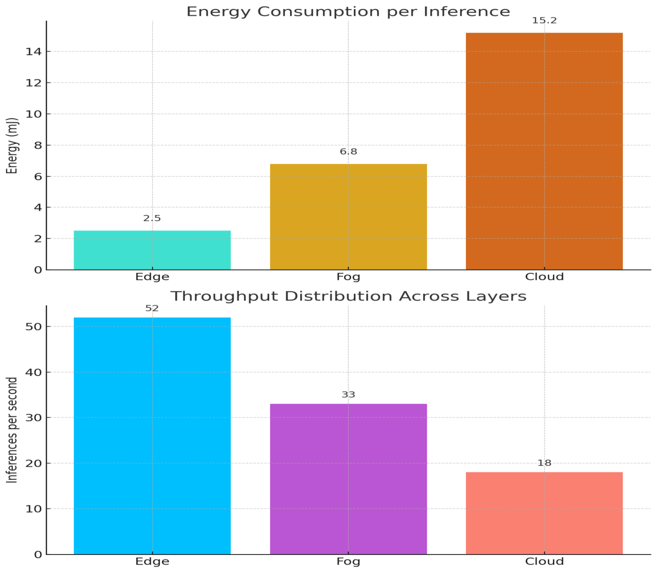


Fig. 3: Bandwidth per layer and Energy usage breakdown

2) Trust Score Evaluation Metrics

Table 6: Trust Score for Federated Learning

Edge Node	Trust Score (Initial)	Trust Score (Final)	Action Taken
Node_01	0.92	0.95	Included in FL round
Node_07	0.76	0.41	Weight reduced
Node_13	0.38	0.29	Update discarded
Node_18	0.85	0.89	Fully aggregated

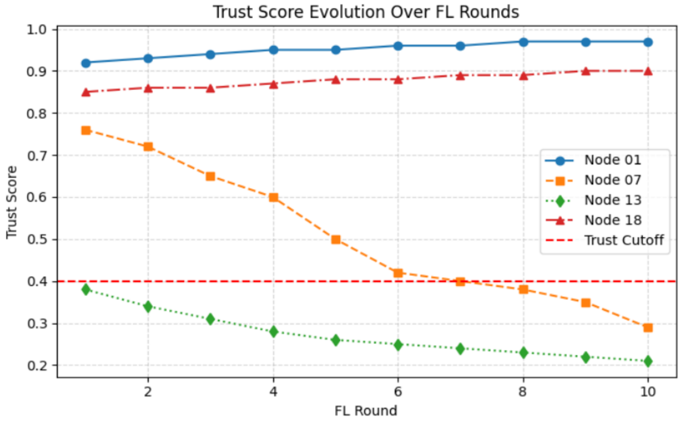


Fig. 4: Trust Score Line Graph

V. CONCLUSION

Following a thorough investigation on the junction of artificial intelligence and edge computing under 5G architecture, this work has produced and validated a hybrid AI framework meant to satisfy the always rising needs of real-time, resource efficient, and safe edge intelligence. The proposed system integrates machine learning, deep learning, reinforcement learning, and federated learning into one coherent and dynamic decision-making engine using a layered approach distributed across Edge, Fog, and Cloud. Early-exit inference lowers response time at the edge layer by conserving energy, so enabling practical use for real-time tasks in smart environments, health monitoring, and surveillance. During federated learning rounds, fog nodes provided mediators for more intricate inference, model aggregation, and trust evaluation, so adding scalability and resilience to the system. Though only used for deeper inference or long-term learning, the cloud layer guaranteed that the global model stayed thorough and current, so supporting knowledge transfer between edge clusters.

MATLAB's extensive simulations revealed the system's latency, accuracy, energy economy, and bandwidth optimization performance. Dynamic task routing depending on real-time metrics was mostly dependent on the reinforcement learning agent, so enhancing system adaptability over load conditions and network volatility. Preserving model integrity while supporting data privacy by federated learning with trust-aware filtering addressed ethical and technical issues. Still unresolved issues are cold-start nodes, fog layer overloads under dense edge deployment, and convergence latency in federated training. Particularly when considering complex adversarial attacks or insider manipulation of trust scores, security is always changing. These spaces provide rich ground for more improvement.

REFERENCES

[1] W. Saad, C. Yin, and M. Debbah, Intelligence and Edge Computing in IoT-Based Applications: "A Review and New Perspectives," Sensors, vol. 23, no. 3, p. 1639, Feb. 2023. [Online]

[2] H. Sedjelmaci, S. M. Senouci, N. Ansari, and A. Boualouache, “A Trusted Hybrid Learning Approach to Secure Edge Computing,” IEEE Consumer Electronics Magazine, vol. PP, no. 99, pp. 1–1, 2021, doi: 10.1109/MCE.2021.3099634.

[3] L. C. Mutalemwa and S. Shin, “A Classification of the Enabling Techniques for Low Latency and Reliable Communications in 5G and Beyond: AI-Enabled Edge Caching,” IEEE Access, vol. 8, pp. 205502–205526, Nov. 2020, doi: 10.1109/ACCESS.2020.3037357.

[4] S. A. Bhat, I. B. Sofi, and C. Y. Chi, “Edge Computing and Its Convergence With Blockchain in 5G and Beyond: Security, Challenges, and Opportunities,” IEEE Access, vol. 8, pp. 205340–205365, Nov. 2020, doi: 10.1109/ACCESS.2020.3037108.

[5] Y. Li, S. Deng, and J. Yin, “Edgent: An Edge Intelligence Framework for Collaborative Deep Learning in Edge Computing,” in Proceedings of the 2019 IEEE/ACM Symposium on Edge Computing (SEC), 2019, pp. 127–140, doi: 10.1109/SEC.2019.00021.

[6] M. Hosseinzadeh, E. Khodayi-mehr, A. Anjomshoaa, and A. A. Rahmani, “A Hybrid Neural-PSO Model for Enhancing QoS in Cloud-Edge IoT Applications,” in Proc. 2020 IEEE Int. Conf. on Industrial Informatics (INDIN), 2020, pp. 987–992, doi: 10.1109/INDIN45578.2020.9442160.

[7] C. Chen, K. Li, W. Zhang, and Y. Li, “Intelligent Traffic Control with Deep Reinforcement Learning in Edge-Enabled Smart Cities,” IEEE Internet of Things Journal, vol. 7, no. 8, pp. 7072–7083, Aug. 2020, doi: 10.1109/JIOT.2020.2964623.

[8] A. Kaur and M. K. Soni, “LTE-A heterogeneous networks using femtocells,” Int. J. Innov. Technol. Explor. Eng., vol. 8, no. 4, pp. 131–134, 2019.

[9] R. Singh and S. D. Joshi, “A comprehensive review on resource allocation techniques in LTE-Advanced small cell heterogeneous networks,” J. Adv. Res. Dyn. Control Syst., vol. 10, no. 12, 2018.

[10] S. Sharma and A. Mehta, “Power control schemes for interference management in LTE-Advanced heterogeneous networks,” Int. J. Recent Technol. Eng. (IJRTE), vol. 8, no. 4, pp. 378–383, Nov. 2019.

[11] M. Yadav and P. Singh, “Performance analysis of resource scheduling techniques in homogeneous and heterogeneous small cell LTE-A networks,” Wireless Pers. Commun., vol. 112, no. 4, pp. 2393–2422, 2020.

[12] N. Gupta and V. Kumar, “Design and analysis of enhanced proportional fair resource scheduling technique with carrier aggregation for small cell LTE-A heterogeneous networks,” Int. J. Adv. Sci. Technol., vol. 29, no. 3, pp. 2429–2436, 2020.