Multicasting Strategies for Increasing Network Efficiency in 5G Using Deep Learning

Wael AlZoubi
Applied Science Department, Al-Balqa Applied University
Jordan
Wa2010@bau.edu.jo

Hazem Hatamleh
Applied Science Department, Al-Balqa Applied University
Jordan
Hazim_hh@bau.edu.jo

Abstract: *Multicasting in Internet of Things (IoT) includes transfer data from one source to multiple destinations instantaneously. One major issue is the lack of standardized procedures leading to interoperability problems and potential weaknesses in diverse IoT systems. To consider above mentioned limitation this investigate discovers the application of Artificial Intelligence (AI) methods to revolutionize numerous facets of multicast management. Initially, adaptive multicast group management influences real-time data on user mobility and scheme settings through sensors and monitoring tools. Using Deep Q-Networks (DQN) accomplished with Self Organizing Map and Particle Swarm Optimization (SOM-PSO). Secondly, AI-driven resource allocation hires Deep Reinforcement Learning (DRL) to examine traffic patterns and current network loads unceasingly. Third, predictive analytics for multicast traffic demand participates historical data and contextual information by means of the Dynamic Threshold Algorithm with Multi-Link Communication (DTA-MLC). Enhanced edge caching strategies apply Context-aware Long Short-Term Memory models with Graph Neural Networks (C-ALSTM-GNN) to forecast content demand at network edges. Finally, AI-based multicast routing procedures develop efficient Quality of Service (QoS) Multicast (EQM) trees to enhance routing paths founded on real-time network topology and traffic conditions. The recommended work is implemented by means of network simulator 3.26, and the efficiency of the proposed model is addressed utilizing several performance metrics such as latency, energy efficiency, throughput, packet delivery ratio, traffic prediction rate. The proposed method achieves latency with 32 ms, energy efficiency with 93%, Traffic Prediction rate 96%, throughput with 342 kbps and PDR with 96%.*

Keywords: *Multicasting, network performance, artificial intelligence, deep Q-networks, deep reinforcement learning, quality of service.*

Received July 20, 2024; accepted November 07, 2024
https://doi.org/10.34028/iajit/22/2/10

1. Introduction

The essential for bandwidth has grown up rapidly as a consequence of the extensive use of 5G requests, which are accompanying in a new era of wireless communication. Also, effective communication methods are dangerous in Internet of Things (IoT) and Wireless Sensor Networks (WSN) situations [1, 2]. The Third-Generation Partnership Project (3GPP), the most important group for mobile communication values. Multicast communication, a transmission mechanism that enables packets to be delivered from a single sender to several recipients simultaneously, is a competitive solution for bandwidth conservation and freshness maintenance if all receivers want the same information [3, 4]. A server multicasts environmental data, such as temperature or robot locations, to every robot in a smart factory. Based on this information and their observations, these robots take action and independently update the server [5]. Similar to this, positioning algorithms in wildfire detection modify the location of sensors in real-time based on variables like wind direction and dryness of the vegetation. Sensors are more widely distributed in low-risk locations than they are in high-risk ones. Location directions are disseminated by the central controller, which lowers overhead and conserves bandwidth. Outdated data might have serious effects in both cases, ranging from unexpected catastrophes in other applications to inefficiencies in robotics [6]. The possibility of multicast in WSN and the significance of data freshness in real-time applications are demonstrated by these instances [7]. The metric Age of Information (AoI), which is defined as the amount of time that has passed from the creation of a device's most recent received update packet, as a way to quantitatively assess how fresh a device. The device’s AoI increases with time and decreases upon receiving a more current update packet. Fresher data is indicated by a lower AoI at the device. Observe that performance measurements like latency are packet-centric, whereas AoI is destination-centric [8]. Prominent for its contributions to mobile communication standards, the 3GPP is essential for optimizing the use of wireless resources and reducing data redundancy, particularly in IoT and video streaming applications. The 3GPP paper deliberates a multicast usage scenario for mission-critical facilities, which proves the possible of multicast transmission in extremely real-time. In order to decrease transmission error and upsurge dependability, 3GPP uses rateless codes as a Forward Error Correction (FEC) device. A source can create an infinite number of packets from data using the rateless code, also known as the digital fountain code, and a receiver can decode the data provided it has enough packets. Numerous studies and applications of rateless codes in the 4G Long Term Evolution (LTE) Multimedia Broadcast Multicast Service (MBMS) have shown how successful they are in enhancing the dependability of multicast/broadcast transmissions [9]. For this reason, rateless codes have also been suggested by 3GPP for use in 5G NR MBS systems. Because they enable the transmitter to deliver freshly created fountain packets rather than retransmitting packets with mistakes, endless codes offer a substantial advantage in multicast circumstances [10]. Of course, there are additional difficulties in providing multimedia services in mobile networks due to their increasing demand [11]. Additionally, cloud servers may find it challenging to meet the computational demands of all connected devices. In this situation, mobile edge computing, or MEC, transfers computations from distant servers to edge base stations that are in closer proximity to users in order to reduce service latency [12]. While there are outstanding works that concentrate on service caching or resource allocation computation, these elements have typically only been optimized in one of these areas. Furthermore, multicast has not yet received as much attention in previous publications that examined computation and cache resource allocation in unicast circumstances. Multicast technology is an effective means of delivering similar content to meet many requests since it makes use of the inherent broadcast aspect of BS channels. MEC and multicast, which concentrate on same content transmission and individualized processing, respectively, initially appear to be two opposing study paths. Still, we find that multicast can be advantageous for a wide range of MEC services [13, 14]. Numerous obstacles stand in the way of multicast-aware resource allocation with simultaneous computation and cache optimization. First, since compute, caching, and multicast are closely related and interact with one another, they ought to be handled collectively. This problem has two variables: the computational allocation choice, which is a continuous variable, and the caching decision, which is an integer of zero–one type. Multicasting is anticipated to be a promising tool that makes it simple for mobile terminal users to access the ubiquitous multimedia experience. The scalability of broadcast and multicast transmissions in mobile networks is enhanced by using evolved Multimedia Broadcast Multicast Service (eMBMS), a point-to-multipoint service that permits data transmissions from a single source to numerous recipients. Additionally, as an improvement to eMBMS, Multicast/Broadcast over Single Frequency Network (MBSFN). This keeps performance stable as User Equipment (UE) moves away from the base station and prevents destructive interferences in areas where coverage overlaps. The Conventional Multicast Scheme (CMS) [15] for resource allocation in multicast transmissions takes a conservative stance, limiting data rate based on the user with the poorest channel circumstances. This method, of course, maximizes the fairness among multicast users; but, those with favorable channel conditions do not achieve the highest bit rates, and the multicast area's throughput performance is extremely inefficient [16]. Fixed block-length codes usually require data to be supplied at a consistent rate that all receivers can tolerate in terms of scalability. Because other receivers are being underutilized, inefficiencies result from having to set the transmission rate depending on the bottleneck device-the receiver with the poorest channel conditions [17]. On the other hand, rateless codes effectively resolve the bottleneck problem and improve network performance by enabling each receiver to collect packets at their own speed. It's interesting to note that reaching the minimum time-average AoI in a multicast network does not always mean that each status update that is given is trustworthy [18, 19]. The bottleneck device will continue to impact the AoI performance of other devices, provided that the source transmits a new update once every device has decoded the previous one. Consequently, the overall AoI performance might be improved by pre-empting the current update at the appropriate moment and creating a new status update. It is possible to identify a policy that reduces the total time-average AoI based on device feedback. However, the complex combinations of system information make it difficult to find such a strategy, especially when the number of devices in the network rises. It's still not clear how to use device feedback knowledge to create a more effective solution [20].

1.1. Motivations and Objectives

In existing methods, we meet the issues such as clustering issues, allocating the schedule problem, to predict the traffic demand problem, energy efficiency problem and finally data transfer problem. They are elaborately discussed below:

a) Issues in clustering: in existing methods the clustering techniques frequently result in an uneven distribution of users, overburdening cluster leaders, and an inability to adjust to changing network conditions.
b) Impacts in allocating the resource: in existing methods algorithms begin to decline as the number of users reaches a certain value. As a result, the suggested course of action for allocating resources might not be the optimum one. It does, however, depict an alternative method of allocating resources among BBUs in centralized radio access networks.
c) Inaccuracy to predict the traffic demand: since

multicast communication forms are difficult and system conditions are dynamic, it is trying to exactly forecast multicast traffic demand. This can consequence in ineffective resource allocation and smooth system congestion.

d) Falls in system energy efficiency: in existing methods in high mobility scenarios, the system's energy efficiency falls relative the extra time needed for the coalition and transmission tree creation phases.

e) Lower in throughput: existing algorithm performs compared to the other algorithms which displays how the cellular Existing algorithm target affects throughput.

The foremost goal and scope of this research is to develop and validate the AI-driven multicast approaches for improving the network performance in 5G systems. This includes the adaptive multicast clustering managing, AI created allotting the resources, for forecast the multicast congestion demand, upgraded edge caching systems, and lastly the multicast routing protocols. The key possibility is to influence in AI approaches to expand the efficiency, user satisfaction and reliability of multicast services in 5G environments. In this research objectives to provide a thorough understanding of how AI can be leveraged to enhance multicasting strategies and, consequently, optimize network performance in 5G. Some of the sub-objectives of this research are provided as follows:

a) To create a system that uses real-time data and to form optimal multicast groups based on dynamic user behavior and network conditions.
b) To design and evaluate an effective model for real-time resource allocation in 5G multicast networks, and to develop and refine predictive analytics models to forecast multicast traffic demand using historical and contextual data.
c) To combine Hybrid models to predict content demand and optimize edge caching strategies for improved performance and user satisfaction, and to create and test AI-driven routing protocols to establish efficient and reliable multicast routes while meeting Quality of Service (QoS) requirements.

1.2. Research Contributions

Below are some of this research's main contributions,

1. A new approach that leverages real-time data and reinforcement learning for dynamic and efficient multicast group management, enhancing user experience and network performance.
2. An innovative deep reinforcement learning model for real-time resource allocation, ensuring optimal utilization and responsiveness of 5G networks.
3. Development of a sophisticated predictive model using the Dynamic Threshold Algorithm with Multi-Link Communication (DTA-MLC) to forecast multicast traffic demand accurately.
4. Introduction of a context-aware LSTM-GNN model for predicting content demand and managing cache placement and replacement in real-time, significantly improving cache hit rates and reducing latency.
5. Creation of Efficient QoS Multicast (EQM) trees using AI, which ensure reliable and cost-effective multicast routing with adherence to QoS constraints.

1.3. Research Organization

The following parts comprise the remaining portion of this document: Section 2 provides an illustration of the literature study of the earlier research that is more pertinent to our work. Section 3 presents the main problem statements addressed in the previous literature. Section 4 presents the research methodology for the proposed work, which consists of a pseudocode, a mathematical representation, and a protocol. In section 5, the experimental results are presented along with a comparison of the recommended and ongoing works. Section 6 offers a conclusion to the suggested study as well as future work plans for this research.

2. Literature Survey

This section deals with the survey of literature on Multicasting Strategies for Optimized Network Performance in 5G using Artificial Intelligence, Mach and Becvar [21], to improve data transmission rate, lessen the load of heavy data traffic, and increase system efficiency, the leitmotif of this research is to provide an architecture for the “Age of Information (AoI)” and “cache-assisted hybrid multicast/unicast/Device-to-Device (D2D)” communication using the promising cell-free “Massive Multiple-Input Multiple-Output (MIMO)” process. The user-centric transmission system is then designed, and the viability of the initial keys for D2D, AoI, multigroup multicast, wireless caching and signal processing typically for transmission mode selection process, cache replacement strategy, and signal processing. Here, some interesting new difficulties and potential future directions are discussed, such as full- band cooperative transmission, RIS, and Simultaneous Wireless Information and Power Transfer (SWIPT). Mahdi and Taşpınar [22], provides a “scalable video multicast” approach “based on user demand perception and D2D communication to enhance the D2D multicast network transmission performance” of scaled movies in “cellular D2D hybrid networks”. They first determine the preferences of users by using factors such as video popularity and watching history in order to improve the number of people who get multicast clusters. This enables them to ascertain the willingness of consumers to push videos. Second, they create a cluster head selection algorithm that takes into account the social factors, video quality needs, and channel quality of consumers. However, finalizing in a series of simulation tests, the suggested model skillfully integrates new users into the multicast group, expands the number of cluster members, satisfies a wide range of requirements improves video quality and elevates the bar for service quality that governs video transmission in the past for D2D connection. The Mishra and Tyagi, in [23] use knowledge from 3GPP standards to present the first thorough system-level evaluation of Ultra Reliable Low Latency Communications (mURLLC). It highlights two important points:

1. How mURLLC differs from traditional multicast broadband wireless communications.
2. Which mURLLC-providing technologies necessitate paradigm shifts in comparison to current solutions. In conclusion, the article offers suggestions for effectively fulfilling the stringent mURLLC requirements. This is due to the possibility that the network operator will require far less channel resources to deliver mURLLC service than what is advised for baseline solutions for broadband multicast or unicast URLLC traffic.

Mustafa *et al*. [24] demonstrate that MC significantly improves multicast streaming performance, which is especially advantageous for cell-edge users who frequently encounter unfavorable channel conditions. They paid particular attention to how many users multi-connected multicast systems could service at once. It has been discovered that more than 60% of the customers who are unsuccessfully served by single-connectivity multicast may be successfully served by employing multi-connectivity in multicast transmissions utilizing the same resources. They also prove that no other polynomial-time approach can provide a better estimate. However, Controller General of Accounts (CGA) process answer and the optimum key found by brute force analysis for a lower problem size match in terms of performance. Additionally, they showed “MC multicasting performs better than Multicast Broadcase Single Frequency Network (MBSFN)”, doing away with the requirement for prolonged cycle prefixes and tight synchronization. Ouyang *et al*. [25] provide a brand-new idea in this study called “BeamForming-as-a-Service (BFaaS)” for broadcast and multicast service delivery in 5G and 6G systems. They begin by providing a thorough review and analysis of the industrial operations and backdrop standards through broadcast initiatives that have been implemented across 5G stages. They clarified the conditions for offering end users multicast and broadcast services, as well as the significance of beamforming for multicast and broadcast service delivery in 5G and 6G networks. From here, they go on to describe the benefits of the suggested BFaaS plan as well as the philosophy and vision of BFaaS. Nonetheless, this plan solved issues with overlapping service regions' real-world deployment. By creating a PMI matrix for the overlapped portion and assigning the proper data layers, this can be resolved.

Pupo *et al*. [26] to tackle and contextualize the intricacy of “Multicast Radio Resource Management (RRM)” and the consequences of sudden changes in the MG members’ reception circumstances. In order to prevent the process from lasting too long, they provide a multicast-oriented trigger, “K-means clustering for group-oriented detection and splitting, a classifier” for determining which multicast access method is best, and a final resource allocation algorithm. We compare heuristic tactics with Machine Learning (ML) multiclass classification algorithms to determine which multicast access strategy best suits the unique reception conditions of the consumers. As access strategies, they take into consideration “Subgrouping Based On Orthogonal/Non-Orthogonal Multiplex Access (OMA/NOMA)” and the traditional multicast scheme (MCS). Also addressed the trade-off among multicasting gain and multiuser diversity, emphasizing the consequences of the connection that currently exists between the propagation frequency, the velocity of users, and the fluctuations in channel conditions. The Tan *et al*. [27] provide a group-based multicast service authentication and data delivery technique based on a model of a typical vehicle multicast service in 5G-V2X. The plan calls for grouping large cars that are covered by the same RAN together and connecting them to the content provider so that they may use the dispersed keys that the 5G home network securely distributes to access a multicast service to make it possible for automobiles to safely receive multicast service data in point-to-multipoint mode. The high rate of aggregation verification failure in the suggested approach's group-based multicast service access authentication process would result in the RAN transmitting all signatures again, which will incur computational and communication overheads. They will continue to look for ideal ways to lessen this restriction.

3. Problem Statement

This section focuses on the unique issues that current works frequently encounter. The suggested remedy is also provided. Several of the specific problem statements that already exist include, Background of existing problems: Tripathi *et al*. [28] to determine which users should receive eNB service, the authors of this study employ machine learning methods, specifically “support vector machine, random forest, and deep neural network.” To enhance the functionality of current clustering techniques, they suggest a mixed-mode content distribution scheme in which the two divided user groups are served by the cluster leaders and evolved Node B (eNB). A scenario for D2D-enabled multicasting has been established in order to conduct a thorough simulation research that shows how the mixed-mode scheme may greatly increase the throughput, energy consumption, and fairness of both individual users' performance and the network as a whole. The problems employed from these approaches are, nevertheless, clustering techniques frequently result in an uneven distribution of users, overburdening cluster leaders, and an inability to adjust to changing network conditions.

In this research by Trung and Anh [29], a learning-based “Resource Segmentation (RS)” technique is proposed to efficiently handle the Resource Allocation (RA) problem using a modified scheme. To get the position coordinates of end users, it modifies the “Random Forest Algorithm (RFA)” and adds the “Signal Interference and Noise Ratio (SINR) and position coordinates”. Additionally, it forecasts the Modulation and Coding Schemes (MCS) needed to connect the “Remote Radio Head (RRH)” to the end-user device. The issue employed in this work are:

1. The performance of both algorithms begins to decline as the number of users reaches a certain value. As a result, the suggested course of action for allocating resources might not be the optimum one. It does, however, depict an alternative method of allocating resources among BBUs in centralized radio access networks.

This research suggests “Multi-rate Multicast Reinforcement Learning based Opportunistic Routing (2MRLOR)” as a resolution to these problems. This technique adapts the nodes communication rates to alteration their communication series. As a consequence, here may be additional applicants nearby for the packet furtherance and a difference in the neighboring nodes. To recognize the optimum forwarders for a packet in a multi-rate scenario, 2MRLOR presents a routing limitation named Expected Multicast Delay (EMD). This procedure regulates respectively node suitable transmission frequency based on the state of the system. Concentrated data distribution among system nodes is additional request of reinforcement learning [30]. Some of the major problems employed in this work are.

2. Since multicast communication forms are difficult and system conditions are dynamic, it is trying to exactly forecast multicast traffic demand. This can consequence in ineffective resource allocation and smooth system congestion.

Wang *et al*. [31] the suggested concept, users are divided into coalitions. While the other coalition members receive service via D2D lines, coalition heads receive Non-Orthogonal Multiple Access (NOMA) unswervingly from the base stations. Using Monte Carlo simulations, they explore the system SE and EE for various mobility patterns and talk about the best possible system configurations. The acquired data demonstrate that, particularly in low mobility scenarios, the suggested scheme performs better than traditional OMA and NOMA. The problems existed in this approach are.

3. In high mobility scenarios, the system's energy efficiency falls relative to NOMA because of the extra time needed for the coalition and transmission tree creation phases.

Zhou *et al*. [32] proposed Shuffled Frog Leaping Algorithm (SFLA) was to RA in D2D multicast communications. They contrasted the outcomes of the SFLA algorithm with those of the “Firefly Algorithm (FA), Ant Colony Optimization (ACO), and Particle Swarm Optimization (PSO)”; with respect to the target Signal Interference Noise Ratio (SINR).

Another example of how well the SFLA algorithm performs compared to the other algorithms which displays how the cellular SINR target affects throughput.

Research solutions: They use one of the more established unsupervised learning algorithms, Self-Organizing Map (SOM), for the multicast group creation in order to rapidly arrive at a close to ideal solution. SOM complexity is quadratical when it comes to the number of map units and linear when it comes to the number of users. The new SOM including Particle Swarm Optimization (PSO) is shown here: After the initial SOM training, integrate PSO to adjust the weights and map topology, making sure the multicast groups are appropriately generated and the clusters are more refined. To overcome the resource allocation problem using the Deep Reinforcement Learning (DRL) it is a best for resource allocation strategy. To adaptively alter resource edges founded on existing system situations, usage a dynamic threshold procedure. To recover network performance and flexibility, usage a multi-link communication approach for balancing the load and severance. This will enhance traffic dispersal transversely numerous relations. Have a judgement on effects and regulate the strategies in retort to information in real time to assurance actual multicast congestion control. Ultimately, the network traffic demand was predicted using the innovative DTA-MLC. Long- Short-Term Model (LSTM) models with Graph Neural Networks (GNNs)- (C-ALSTM-GNN) to manage the energy efficiency in the 5G network. EQM (Efficient QoS Multicast) trees should be established. When queuing size ratio and link stability are combined, an EQM tree is a shortest-path multicast tree with the lowest End-to-End (E2E) cost. It must adhere to QoS restrictions, which include avoiding primary users’ licensed channel and queuing size ratio and link stability.

4. Proposed Method

In this research objectives to provide a thorough understanding of how AI can be leveraged to enhance multicasting strategies and, consequently, optimize network performance in 5G. Figure 1 indicates the overall architecture of this research.

1. Adaptive multicast group management.

2. AI-Driven resource allocation.
3. Predictive analytics for multicast traffic demand.
4. Enhanced edge caching strategies.
5. AI-based multicast routing protocol.

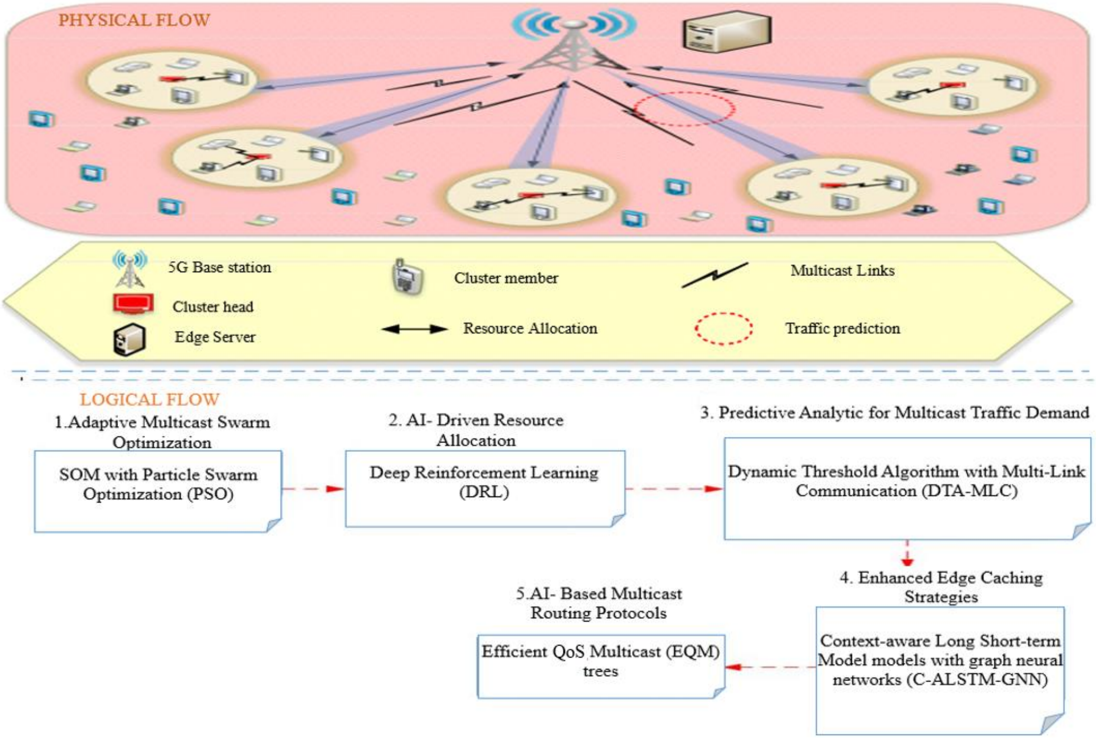
PHYSICAL FLOW
5G Base station
Cluster head
Edge Server
Cluster member
Resource Allocation
Multicast Links
Traffic prediction
LOGICAL FLOW
1.Adaptive Multicast Swarm Optimization
SOM with Particle Swarm Optimization (PSO)
2. AI- Driven Resource Allocation
Deep Reinforcement Learning (DRL)
3. Predictive Analytic for Multicast Traffic Demand
Dynamic Threshold Algorithm with Multi-Link Communication (DTA-MLC)
4. Enhanced Edge Caching Strategies
Context-aware Long Short-term Model models with graph neural networks (C-ALSTM-GNN)
5.AI- Based Multicast Routing Protocols
Efficient QoS Multicast (EQM) trees

Figure 1. Overall architecture of this research.

4.1. Adaptive Multicast Group Management

The method of adaptive multicast group managing includes uninterruptedly gathering real-time data on user mobility, device usage patterns, and system settings by means of sensors and monitoring tools. This information is kept in a central repository, allowing for ongoing study and model training. Consistent updates to the train the dataset, reflecting present system settings and user actions, confirm that Deep Q-Networks (DQN), remains accurate and adaptive. For handling huge state-space Markov Decision Process (MDPs) with complex dynamics $D(St+1|St,\ Dt)$, DQN is a well-liked “Deep Reinforcement Learning (DRL)” approach. With the DQN, an action-value functional is roughly represented by a neural network using a value- iteration technique. Despite the fact that this approach can be improved upon by a number of subsequent publications, we choose to adopt it because of its simplicity. We demonstrate that DQN itself can give us the best tracking and solution. These enhancements could lead to even better results in terms of estimator variance, sample efficacy, etc. Prior attempts at integrating nonlinear function approximators, like neural networks and RL, failed due to instabilities caused by:

a) Correlated samples for training.
b) A radical change in policy with a small improvement in operate estimation.
c) Correlation between the function being trained and approximated function.

Experience replay memory (M) and the target network ($Q\theta*$) are the two essential components of the algorithm that handle these problems and are responsible for the success of DQN. The changeovers of an MDP, namely the tuple (S_t, D_t, r_t, S_{t+1}) are stored in the replay memory. Next, the program uniformly samples a minibatch of randomly selected transitions from memory. In doing so, the data correlation is eliminated, and the data distribution shift with each iteration is smoothed out. An additional neural network in the algorithm approximates the value function, Q_θ. A randomly selected mini batch from memory M and the target network make up the training set used to train the Q_θ at each epoch.

The following *i.i.d* samples are provided by this random sampling for stochastic gradient descent with loss.

$$k_Q^{\pi} = \sum_{i=0}^{n}(x_i - Q_\theta(S_i, A_i))^{\,2} \tag{1}$$

Where $x_i = r_j + \tau\, max_{a'} Q_\theta * (S_i, a')$. The iteration $\{\theta_t\}$ are given by the following:

$$\theta_{t+1} = \theta_t - \eta_1(t)\nabla_t k_Q^{\pi\theta} \tag{2}$$

Where $\eta 1(t)$, the step size, satisfies the following:

$$\sum_{t=0}^{\infty}\eta_1(t) = \infty\,, \qquad \sum_{t=0}^{\infty}\eta_1^2(t) < \infty, \eta_1(t) \geq 0 \tag{3}$$

For epochs T_{target}, the weights of the target network $Q*$ are maintained constant, limiting any significant policy changes and decreasing the correlation between Q and $Q*$. In non-parametric regression, this can be viewed as a risk minimization problem with regression function $Q\theta *$ and risk $\kappa_Q^{\pi\theta}$. The trained model is used to forecast optimal user groupings for multicast by means of the Self Organizing Map with Particle Swarm Optimization (SOM-PSO), with constraints accustomed based on feedback and new data to enhance prediction accuracy.

4.1.1. Self-Organizing Map (SOM)

The clustering algorithm used was Self Organizing Map (SOM), an unsupervised machine learning approach. SOM can identify a data set's cluster in a competitive manner. First, the neuron with the weight closest to the data sample is identified as the winner neuron via SOM. The updating of neighboring neurons' weights then occurs, ensuring the formation of clusters of neurons with comparable weights.

The weight vectors are updated using two functions: the neighborhood function $[h_{cj}(t)]$ and the learning rate $[\beta(t)]$ $[\alpha(t)]$. The learning rate has a value between 0 and 1. The neighborhood function's Gaussian type is expressed as follows:

$$h_{ui}(t) = exp\left(\frac{d_{ui}^2}{2\sigma^2(t)}\right) \tag{4}$$

where d_{ui}^2 is the separation between the stimulated neuron j and the winner neuron u. The parameter σ represents the neighborhood' s radius at iteration t. $w_i(t+1)$ is the weight vector. The following is a basic algorithm that represents SOM's algorithm.

4.1.2. Particle Swarm Optimization (PSO)

The algorithm is divided into two stages: cluster creation and CH selection. PSO is used to determine which CH to use. The CH selection procedure using distance and residual energy factors. During the CH selection stage, each sensor node first transmits its position and remaining energy to the base station, which uses this information to determine if the node meets the threshold energy required to be eligible for a CH. The base station next executes the PSO-based CH selection process, which is followed by the cluster building phase. We calculate the weight function for the cluster formation using a number of variables, including the node degree, energy, and distance of the CHs. Prior to presenting the linear programming formulation for the cluster head selection problem, we first outline our suggested PSO based strategy for CH selection and give a detailed presentation of the cluster formation phase.

Prior to presenting the Linear Programming (LP) formulation and suggested algorithm, we define a few terms that are pertinent.

Terminologies

In order to facilitate comprehension of the suggested algorithm, we first define a few terms as follows:

1. δ: set of sensor nodes $\delta = \{\delta_1, \delta_2, \dots, \delta_n\}$
2. ς: set of cluster head $\varsigma = \{\varsigma h_1, \varsigma h_2, \dots, \varsigma h_m\}$
3. Where: $m < n$
4. l_j: number of δ in the cluster j.
5. d_{max}: the sensor node's maximum communication range.
6. R_{max} the maximum ςh can communicate.
7. T_H: the energy barrier for becoming a ςh
8. d_0: the threshold distance
9. $\varepsilon_{\varsigma h_j}$: the cluster head ςh_j current energy
10. ε_{δ_i}: The sensor node's initial energy was δ_i
11. $Dis(\delta_i, \delta_j)$: The distance of two sensor nodes, δ_i and δ_j
12. $Comm(\delta_i)$: The group of nodes that are inside the (δ_i) communication range, that is,

$$Comm(\delta_i) = |\left\{\delta_j \middle| \forall_{\delta_j} \epsilon \delta \Lambda Dis(\delta_i, \delta_j)\right) \leq d_{max} \tag{5}$$

The period until the first node dies, the amount of time that passes between a node's first deployment and the point at which a predetermined percentage of nodes run out of energy, and other factors are some examples of how the network lifetime is specified. Nevertheless, we take into account it as the number of rounds till the last node death in the suggested approach, which is commonly referred to as Last Node Death (LND). One way to express a single node's lifetime is as follows:

$$L = \frac{e_{initial}}{e_{total}} \tag{6}$$

where e_{total} is the e_{total} used by the sensor to transmit and receive data, and einitial is the initial energy of a sensor node. This expression

$$e_{total} = E_{TX}(l, d) + E_{RX}(l) \tag{7}$$

E_{TX} is the total energy consumption for transmitting, E_{RX} is the total energy consumed by the receiver to receive.

4.1.2.1. LP Formulation for ς̧h Selection Problem

The main objective of the proposed algorithm is to select ς̧h from the standard sensor nodes while considering energy efficiency to increase the network lifetime. We take into account the sensor nodes' residual energy as well as a number of distance characteristics, such as the average intra-cluster distance and the sensor nodes' distance from the sink, in order to select the ς̧h with the best energy efficiency.

Let f_1 be a function of the ςh' sink distance and average intra-cluster. For the best ςh selection, f_1 must be minimized. Let f2 represent a function that is the reciprocal of the total energy of the chosen ςh current. Keep in mind that the ideal ςh selection should involve maximizing this ratio. This indicates that f_2, or its reciprocal, needs to be decreased. We normalize the two goal functions between 0 and 1 to effectively reduce the resulting linear combinations of both of these functions.

$$Min\, F = \beta \times f_{1\_}(1-\beta) f_2 \tag{8}$$

$$dis(\delta_i, \varsigma h_j) \leq d_{max}, \forall_{\delta_j} \epsilon \delta \ and \ \varsigma h_j \epsilon \varsigma \tag{9}$$

$$dis(\varsigma h_j, BS) \leq R_{max}, \qquad \forall_{\varsigma h_j} \varsigma \tag{10}$$

$$\varepsilon_{\varsigma h_j} > T_H \ 1 \leq j \leq m \tag{11}$$

$$0 < \beta < 1 \tag{12}$$

$$0 < f_1, f_2 < 1 \tag{13}$$

Equation (8) state that the sensor nodes are inside the ςh nodes' intra-cluster communication range. Moreover, the limitation in Equation (9) indicates that the base station will be inside the maximum communication range of ςh. Constraint in Equation (10) states that the energy of each ςh node must exceed the threshold value, and which is the average value of all sensor nodes. Constraint Equation in (11) ensures that the values of the two objective functions are normalized between 0 and 1. The relationship among the energy and distance variables is governed by Equation (12) which ensures that neither of them has a weight of 100% or 0.

4.1.2.2. Particle Representation and Initialization

In PSO, a particle is the same as a whole solution. It shows the optimal CH positions for the ςh selection of the suggested method. The ith particle of the population is represented by:

$$p_i = [\mathrm{X}_{i,1}(t), \mathrm{X}_{i,2}(t), \mathrm{X}_{i,3}(t), \ldots, \mathrm{X}_{i,D}(t) \tag{14}$$

Each component, $\mathrm{X}_{i,d}(t) = (\mathrm{x}_{id}(t), \mathrm{y}_{id}(t)), 1 \leq i \leq N_p, 1 \leq d < -D$, shows the sensor nodes' coordinates that should be selected as ςh. Next, the *ith* particle might be displayed as follows:

$$P_i = [(\mathrm{x}_{i,1}(t), \mathrm{y}_{i,1}(t)), (\mathrm{x}_{i,2}(t), \mathrm{y}_{i,2}(t)), (\mathrm{x}_{i,3}(t), \mathrm{y}_{i,3}(t)), \ldots, \mathrm{x}_{id}(t), \mathrm{y}_{id}(t))] \tag{15}$$

Every particle has the same number of dimensions (D), which is equal to the number of $\varsigma h(m)$. We use a metaphorical example to demonstrate it, where o indicates the generated at random coordinates of the sensor nodes, s indicates the position of the sensor nodes, and ςh indicates the index of the cluster heads.

4.1.2.3. Derivation of Fitness Function

The following variables affect how the fitness function is derived:

1. Average intra- cluster distance

It is defined as $\frac{1}{l_j}\sum_{i=1}^{l_j} dis(\delta_i, \varsigma h_j)$which is the average of the total of the distances of all the sensor nodes from their chosen ςh. Every sensor node in an intra-cluster communication system uses energy to transmit data to its ςh. We need to shorten this average intra-cluster communication distance in order to use less energy. This indicates that a sensor that is close to each sensor node must be chosen to serve as a ςh.

2. Average sink distance

It can be expressed as $\frac{1}{l_j} dis(\delta_i, BS)$, This is the ratio of the quantity of sensor nodes l_j in the cluster head ςh_j to the distance from the cluster head and Base Station (BS). In the data route phase, they must transmit their collected information to the BS every hour. Therefore, in order to use less energy, we must shorten the distance among every CH and the BS. Reducing the standard intra-cluster and sinks length for each ςh is the aim of optimal selection.

$$Minf_1 = \sum_{j=1}^{m} \frac{1}{i_j} \left(\sum_{i=1}^{l_j} dis(\delta_i, \varsigma h_j) + dis(\delta_i, BS) \right) \tag{16}$$

3. Energy parameter

The current energy of every cluster head $\varsigma h_j 1 \leq j \leq m$, selected in a sequence from the normal sensor nodes, is denoted by $\varepsilon_{\varsigma h_j}$. All selected CHs will have an overall current energy of $\sum_{j=1}^{m} \frac{1}{i_j} \varepsilon_{\varsigma h_j}$. Therefore, it makes sense to minimize its reciprocal and maximize the overall current energy of all the selected ςh when selecting the optimal cluster heads. And so, this is our second objective.

$$Minf_2 = \frac{1}{\sum_{j=1}^{m} \varepsilon_{\varsigma h_j}} \tag{17}$$

Since the two aforementioned objective functions do not substantially conflict with one another, it is prudent in our Instead of minimizing each objective function separately, the PSO approach is used to reduce the linear relationship between the two. Then, a single, ideal answer exists. As a result, we employ the fitness function listed below.

$$Fitness = \beta \times f_1 + (1 - \beta) \times f_1, 0 < \beta < 1 \tag{18}$$

Reducing the fitness value is our goal. The particle position-that is, the ςh selection-is better the lower the fitness value. A scheme for real-time monitoring and alteration of multicast groups is applied, creation immediate changes based on the models' predictions. Continuous performance assessment, directing on system of measurement like as latency, bandwidth usage, and user satisfaction, allows for fine-tuning of the model and modification approaches to continue optimal multicast performance.

4.2. AI-Driven Resource Allocation

After clustering we perform the resource allocation. To enhance network resource allocation by means of the DRL for multicast traffic, the procedure initiates with the continuous investigation of real-time and historical traffic patterns and current network load. DRL includes the outside world and an agent. By taking various activities, the agent modifies the external environment, and the environment returns the favour by rewarding the agent. DRL seeks to identify the best course of action that maximizes reward. Three types of Deep

Reinforcement learning algorithms exist: policy-based, actor-critical, and value-based. DQN, a quality-based technique in the family of reinforcement learning methods, has not a policy network and only one value function model.

4.2.1. State Space

If we know the location distribution of users, we assume that a specific r serves user traffic demand. B_r is a representation of the standard bandwidth of the unicast users that r serves. For the multicast users, we treat the bandwidth within the k-th multicast service group as B_r, k. Consequently, a vector B_r, where B_r=[$B_{r,0}$, B_r, 1, B_r, 2, …, B_r, k], can be used to represent the bandwidth allocation status of r. K is the quantity of multicast group services. Note that any two distinct RRHs assigned to the same spectrum band for each $k\in$ {1, …, k}. [N_1, N_2, …, N_M] is a representation of the DRL state space, where M is the total number.

4.2.2. Action Space

Every decision epoch, the DRL agent will choose one RRH at random to guarantee a clear-cut choice. Here, the DRL agent determines what increases or decreases a specific service's bandwidth and satisfies the requirement for constraint. Ultimately, the rules that determine the reward system are as follows: The greater the incentive, the lower the transmission energy throughput consumption ratio overall after the action. Consequently, we set the reward to equal E_{max}-E, where E_{max} is the maximum energy throughput used that RRH may provide, and E is the energy throughput consumption ratio after the action.

DRL consists of an online deep Q-learning phase and an offline Deep Learning (DL) construction phase to minimize computing complexity. During the offline stage, each state-action pair (s, a) and associated value function Q(s, a) are correlated using a DL structure. Here is how Q(s, a) is defined:

$$Q(s,a) = r(s,a,s') + \lambda Q(s',a') \tag{19}$$

Where r(s, a, s') denotes the reward for carrying out action a, λ stands for the discounted parameter, and p(a'| s') denotes the likelihood of carrying out action a' in the subsequent state s'.

During the online phase, the actual and predicted action-state value functions are stored in DL named Q*(s, a, θ*) and Q(s, a; θ), respectively. The current state and action are represented by s and a, respectively, while the neural strengths of the estimated and real networks are denoted by θ* and θ.

The DRL agent use the greedy approach of probability ε to select the next action in every training epoch.

Second, the convex optimization problem P1 can be solved to determine the ideal beamforming weight following the execution of state a. Subsequently, the agent will retrieve the subsequent states and the instant reward r(s, a, s') from the cloud RAN. It will then store the state transition (s, a, r(s, a, s') s') in the experience memory D, which has a capacity of M_D. Third, using the historical data in D, the small batch gradient descent algorithm will update the parameters in Q*(s, a, θ*) DL. Q*(s, a, θ*)-Q(s, a, θ) is the loss function. Ultimately, after every C step, we transfer the parameters from the Q*(s, a, θ*) DL to the Q(s, a, θ) DL. Algorithm (2) below illustrates the DRF resource allocation framework's flow. The process's "average time complexity" is $0(|S|^3)$, where |S| represents the state space's size. This information is employed to update and improve traffic models frequently. A DRL model is established and unceasingly advanced by means of updated traffic data, confirming it familiarizes to developing network environments and pattern of traffic. Numerous network situations are frequently replicated to train the model on dynamic resource allocation plans, with the simulation environment reorganized to reflect existing network states. The AI model is then arranged in the live network to achieve resource allocation in real-time, with continuous monitoring to confirm ideal choices. Real-time feedback and performance system of measurement are used to regulate the model parameters and approaches, confirming ongoing optimization of resource usage.

4.3. Predictive Analytics for Multicast Traffic Demand

To allocate the resource after analyze the multicast traffic demand prediction. Predictive analytics for multicast traffic demand includes the incessant gathering of historical traffic data and contextual data, such as time of day, location, and user behavior. The dataset is frequently updated to reflect present situations, are advanced and uninterruptedly advanced to forecast future multicast traffic demand using the Dynamic Threshold Algorithm with Multi-Link Communication (DTA-MLC). Forecasts are used to pre-allocate network resources to areas and times with anticipated high demand, with approaches incessantly updated based on the recent forecasts and real traffic data. Real-time modifications to resource allocation are finished based on efficient forecasts and actual situations, with feedback loops in place to improve forecast models and allocation approaches. Systematic assessments of the predictive model's efficiency in enlightening Quality of Service (QoS) guide additional modifications and refinements.

4.3.1. Dynamic Threshold Algorithm (DTA)

The chosen network resource in this instance is the network's maximum session count per hour, which is subject to a maximum value. To lower the number of lost sessions in the network, the maximum number of permitted sessions must be divided among all slices.

This split can be performed dynamically, where the threshold for each slice is changed in real-time based on demand, or statically, where a fixed threshold is specified for each slice. We will examine three splitting methods: a best-possible dynamic thresholds method which relies on its choices on predictions made for the slices' next hour; an optimal dynamic threshold method that supports its splitting decisions on the potential information of all slices' metrics (as if the projections were always correct). By applying a fixed threshold for each slice, which is determined by an optimization searching on the historical values of the asset in question for that slice, a fixed-threshold method divides the greatest number of sessions among the slices. Furthermore, the highest number of sessions the network can have in the tests, assuming no sessions are lost, varies from 0 to the greatest range of sessions needed for the network to support every user. The performance parameter that is reduced for this problem is the number of sessions within the threshold for that slice, as stated in Equation (19) using Iverson bracket notation, wherein $n_s(\partial)$ is the amount of session in a slice. $\mathfrak{H}$ defines the thershold; ð defines the length,

$$\mathcal{L}(n_s(\partial), \mathfrak{H}(\partial), t) = \sum_{\substack{t=0 \\ > \mathfrak{H}(\partial), t)}}^{\eth(n_s(\partial)} n_s(\partial)[t] - \mathfrak{H}(\partial)[t] n_s(\partial)[t] \quad (20)$$

Algorithm 1: Dynamic Threshold Algorithm.

```
Procedure $Dynamic_{\mathfrak{H}}(\psi_s, min_s NoS, R_{max})$
  For $\partial$ in $NoS$ do
      If $\psi_s(\partial) \leq min_s(\partial)$ then
          $n_s(\partial) = min_s(\partial)$
      Else
          $n_s(\partial) = min_s(\partial) + \frac{\psi_s(\partial) - min_s(\partial)}{\sum_\partial \psi_\partial(\partial)} * + free_{res}$
      End if
  End for
  Return $n_s$
End Procedure
```

The dynamic threshold algorithm adjusts the threshold value based on forecasts for the next hour in order to better adjust to variations in network traffic and maximize the number of sessions that are accessible in the network. It is possible to set a minimum number of sessions required for a slice, so future sessions cannot be added to a slice that is predicted to have few sessions. The dynamic threshold algorithm is shown below Algorithm (1) to control the resources of each slice. The following are the components of this method: mins is the minimum number of resources and number of sessions provided in each slice ∂; $n_s(\partial)$; represents the amount of resources allocated to slice x; NoS is the amount of active slices; Rmax is the greatest number of actual resources that can be split between each of the slices; and $\psi_s(\partial)$ is the estimated amount of the necessary resources and discussions for slice ∂. The resources allocated to each slice are determined by the dynamic threshold approach using the following two criteria: If every forecast for the approaching the period are less than the number of assets assigned for that slice, it is assigned to that slice's minimum resources; Algorithm (2) if otherwise, it is allocated to that slice's minimum resources plus several additional resources, depending on the expected needed assets and the network's available resources. In order to establish a baseline for the optimal result, we also investigate a dynamic threshold method where actual values are employed as the forecasts.

4.3.2. Multi-link Communication

This architecture makes use of a multi-link approach to estimate traffic, analyse user demands, and recommend content. It also makes decisions on data transport. This solution enables the traffic shaping function in multi-link methods of the content retrieval service to provide further traffic optimization. The highest production rate is achieved by users who can serve the greatest number of receivers with their required content. Because of the features of the Zeta dispersion, devices can transfer content to several users at once, and a cluster can include multiple IoT pairings. This issue develops when numerous users may seek the same information at the same time because to Zeta dispersion and the widespread use of the material that individuals have stored on their computers. Put otherwise, this kind of occurrence might occur within the network. Set ϑ has an entirely random number of users, determined by the beta coefficient and a number of system attributes. The fact that they all require consumers to submit the same kind of video content is what matters most. Rather than activating just one transmitter per cluster, when user interference increases and more transmitters become active inside every cluster, we engaged more than two transmitters in this study. In order to evaluate the potential productivity gains inside the system, we have accounted for user interference as well as intra- and intercluster interference.

4.3.2.1. Protocols for Users to Submit Requests

The Zeta distribution is a way to describe a video file's popularity, as was previously indicated. We take a particularly extreme user request scenario as an illustration of the effectiveness of the system. Put another way, we believe there is a possibility that all network users will make a simultaneous request for a video file. If the user is on a device in the cluster that is close to them, they will get the files they have requested over the IoT communications channel. Users will get the files they have requested through the central station, unless otherwise noted. Using the Zeta distribution, each participant selects a random integer ranging from 1 and M. The produced number and the requested central station file ID can be compared. In this instance, the parameter's objective will be ascertained. This feature shows the popularity of each video file as well as the number of requests made for that specific clip. The users should indicate their requests before receiving the data. Cluster-to-cluster variations in IoT communication link quality are caused by the placement detachment of the applicants. Consequently, different users perceive the communication channel at different speeds. Here, the following is how the common path loss model is used:

$$\rho_r = \rho_t d^{-a} \tag{21}$$

As a result, the transmission power (p_t) metric has significance in this situation. The route loss coefficient is represented by the parameter a (alpha), while the distance between the transmitter and receiver is indicated by the parameter d. The degree of control that each user has is represented by this correlation. It makes sense that each user would get less power the further they were from their source and receiver, even though the opposite is also true. We have taken into account the reality that there will always be noise on wireless networks, nevertheless. The signal-to-noise ratio is determined as follows, assuming that the system's noise level and every communication have a ϱ Gaussian distribution with a certain variance:

$$SNR = \frac{\rho_r}{\varrho} \tag{22}$$

Once we are aware of each wireless connection's signal-to-noise ratio, we can apply the well-known Shannon relation to determine the link speed:

$$\varpi = \tau log_2(1 + SNR) \tag{23}$$

The W value in Hertz above indicates the communication's bandwidth. The speed at which data is carried via a wireless connection is measured in bits per second. Furthermore, the connection in Equation (23) between the communication link's introduction and its capacity is given. The terms "capacity" and "speed" for a communication link will henceforth be used interchangeably. The bandwidth of a single user's wireless link can be used to determine the total capacity of the network for communicating between a tiny station and multiple users.

$$Tp = \sum_{x=1}^{N} \varpi_x \tag{24}$$

We discuss the *pth* user's access to a entire of *N* wireless communiqué lines for this reason. The ϖ_x parameter represents the communication link capability of the *xth* user. When a cell environment has the same number of clusters as nodes, then Q is the entire system capacity (including IoT communication).

$$T_{femto} = \sum_{q=1}^{Q} T_q = \sum_{x=1}^{N_q} \varpi_x \tag{25}$$

The number of connections between the cluster's *ith* and *qth* is represented by the N_q parameter in the calculation above. Complex random-process equations must be solved in order to calculate N_q The precise value of this parameter will be computed using mathematical analysis, with simulations being used to ascertain the parameter's value.

4.3.2.2. Selecting the best Users

Considering the capacity of the wireless communication network that was described in the previous section, we now need to turn on the devices that enable the network to function at its fastest possible rate (data speed). Stated otherwise, we choose and activate users who can create the greatest transmission rate in the network, if each cluster contains five possible transmitters, all of which can serve multiple users.

4.4. Enhanced Edge Caching Strategies

Enhanced edge caching approaches initiate with the continuous investigation of real-time data on content requests and user preferences at the system edge. This investigation is frequently modernized with the modern usage patterns. DL models are trained to forecast content demand at the network edge by means of this modernized information and are retrained occasionally to adapt to altering user preferences and content trends. Combines Context-Aware Long- Short- Term Model with Graph Neural Networks (C-ALSTM-GNN) to forecast content demand. LSTM detentions temporal patterns, while GNNs model the associations among users and content across dissimilar contexts that algorithms for dynamic cache placement and replacement approaches are advanced and incessantly polished, then applied in edge servers to optimize caching in real-time. Constant monitoring of cache performance, with cache hit rates, latency, and system load, permits for unvarying assessment and modification of the caching approach to recover its success.

4.4.1. LSTM

The current study indicates that LTSM is a useful technique for temporal prediction. A particular kind of neural network called an RNN is designed to mimic sequence- or time-dependent behaviour. One type of RNN network in particular is an LSTM. The structure of the hidden unit is replaced with memory blocks in the LSTM network, which sets it apart from the traditional RNN.

1. Input Gates: this gate, which is the network's prior output, regulates the activation of input into the cells using the input X_t and the preceding input value $h_{t\text{-}1}$. The following is a representation of its output.

$$\kappa_t = \tanh(W_k([h_{t-1}, \mathrm{X}_t]) + b_k \tag{26}$$

$$i_t = \sigma(W_i[h_{t-1}, \chi_t] + b_i \tag{27}$$

Where b_k and b_i represent the input bias, and W_k and W_i are the input and previous cell outputs weights, accordingly.

2. Forget gates: the output of this gate, which adapts resets the cell's memory, is stated as follows:

$$f_t = \sigma(W_f[h_{t-1}, \mathrm{X}_t] + b_f) \tag{28}$$

Where b_f is the input bias and W_f is the "weight for the input cell output".

3. Memory gates: The network's temporal state is stored by this gate, and its output is as follows:

$$C_t = C_{t-1} \odot f_t + i_t \odot \kappa_t \tag{29}$$

Where C_t=C_t-1 is, respectively, the previous cell output and the gates' output current. Figure 2 represents the structure of LSTM.

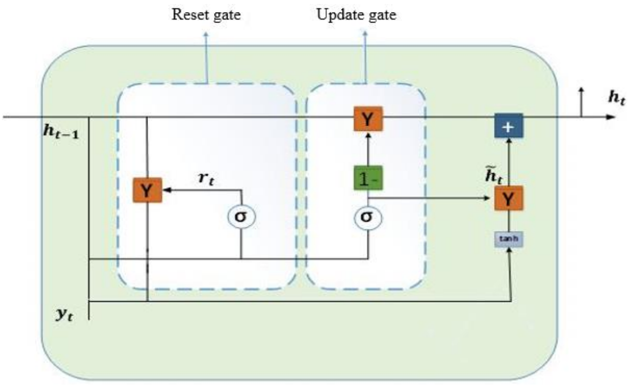

Figure 2. Structure of LSTM.

4. Output Gates: the following is the output of this gate, which modifies the cells' output activation:

$$o_t = \sigma(W_o h_{t-1}, \mathrm{X}_t + b_o) \tag{30}$$

$$h_t = o_t \odot \tanh(C_t) \tag{31}$$

Where W_o represents the relative "weights for the input" value and prior cell output, and b_o stands for the "input bias". For different multicast services, we might use LSTM predictions to estimate the total number of consumers and the separation between users and RHs.

The Unicast-Multicast Allocation of Resources System can then be used to proactively address user demand by using this data as input. We can predict the number of consumers and the greatest distance between them for every multicast service in the months to come by using the data from the last three to nine days, as shown in Figure 3. This prediction classical is derived from the sequential forecasting technique. First, we transform the base station traffic data into data frames in order to offer "an array of data for the LSTM" forecasting model. "The dimensional array (*N*, *W*, *F*)" that the LSTM prediction model utilizes is made up of the quantity of input sequencing (*N*), "the length of the sequences (*W*)," and the number of features (*F*) in each sequence. In this research, we have chosen to use a slide window that has 96 time points (sequence length). It is a succession of sliding windows. The "LSTM prediction model adopts" the "supervised learning paradigm," which makes use of data for training with "input (*X*) and output (*Y*) components." "The observations collected during a slide frame that ended in time step *t*-1" served as the input for the forecast of the data in the subsequent time step *t*. Data collected at "time step *t* is used to calculate the prediction errors." By optimizing parameters, the training of an "LSTM prediction model" aims to reduce mistakes.

4.4.2. GNN

Taking the graph-structured data "*g* (*V*, *V*˘,*E*, *E*˘)" as input, the GCN conducts two sorts of processes on the data: information reduction and message transmission. We define the following two functions for the two sorts of operations. The message function determines the theme that should be transmitted from node *v* to node *u*.

$$Msg(v,u) = \bar{v}, \bar{e}, v, u\epsilon V, e = (v,u)\epsilon E, \tag{32}$$

where nodes v and u are linked in *g*(*V*, *V*˘,*E*, *E*˘) by a link *e*=(*v*, *u*).The communications that every node in *V* receives from its neighbours are reduced by the reduction function.

$$rdu(v) = \sum_{\{u:(u,v)\in E\}} msg\ (u,v), v \in V \tag{33}$$

Next, the new feature vector of *v* is obtained by passing {*rdu*(*v*),∀*v*∈*V*} we send through a linear network in the GCN, and the transmission purpose from layer-l to layer-(l+1) is specified as;

$$\breve{v}^{l+1} = \sigma(W.rdu^{(l)}(v) + b) \tag{34}$$

Where $rdu^{(l)}(v)$ and is the decreased data for node *v* obtained at layer-l of the linear network, Wand *b* indicates the weight matrices and bias of the linear network, and *σ*(.) is the on linear transmission function. To be able to collect processed graph-structured information and generate a vector for its representation, we design a pooling layer after numerous layers of GCNs. More specifically, we select the layer for pooling that takes the average of the feature vectors for every node.

$$\tilde{g} = \frac{1}{|V|}\sum_{v\epsilon V} \breve{v}^{(k)} \tag{35}$$

Where |*V*| is the quantity of nodes in $\tilde{g}$ is the obtained vector, and k is the amount of GCN layers. At last, we send $\tilde{g}$ forgets the final output and proceeds through several linear levels.

4.5. AI-Based Multicast Routing Protocols

The advance of EQM trees-based multicast routing protocols starts with the incessant gathering of real-time data on network topology, traffic patterns, and link states, sustaining a modernized dataset for continuing examination and model training. AI processes for ideal multicast routing results are established and unceasingly advanced by means of this efficient network data. Dissimilar network scenarios are frequently replicated to test and refine the routing procedures, with the simulation environment efficient to replicate current network situations. To create EQM trees. An EQM tree is a shortest-path multicast tree with minimum E2E cost when it comes to QoS restrictions such queue size ratio, link stability, number of hops, number of time slots, and avoiding the primary users' permitted channel. (a combinations of these factors). The routing protocols are then arranged in the live network, with incessant monitoring to ensure they adapt to real-time variations. Continuing performance monitoring and the usage of real-time data enable continuous optimization of routing results, confirming the procedures continue effective and efficient.

This section introduces the DQN Design for QoS Multicast Routing (DQMR) protocol, which constructs EQM trees-a shortest-path multicast tree with minimum E2E cost-subject to QoS restrictions, avoiding primary user areas and reducing interference links. In addition, the DQMR protocol offers high PDR, low control overhead, minimal routing delay, and good stability. In IoT environment can move according to several mobility models. Specifically, nodes 1 through 11 can move in accordance with the Random WayPoint mobility (RWP) model, with the Reference Point Group Mobility model (RPGM).

Figure 3 illustrates the steps of the Dynamic Queue-based MAC-layer Routing (DQMR) Protocol.

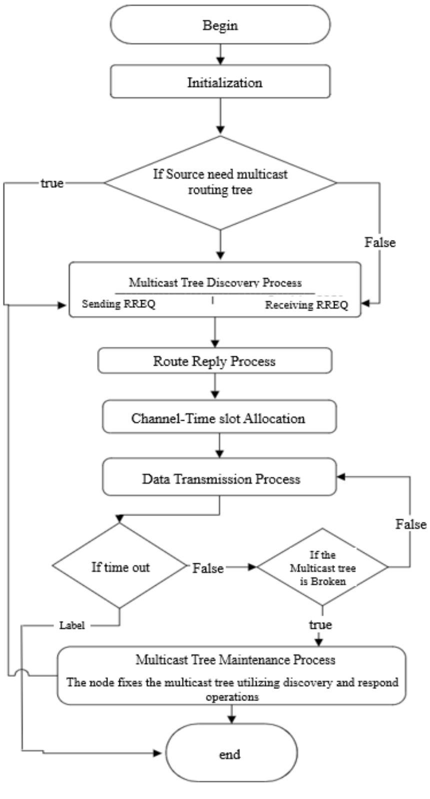
Begin
Initialization
If Source need multicast routing tree
true
False
Multicast Tree Discovery Process
Sending RREQ
Receiving RREQ
Route Reply Process
Channel-Time slot Allocation
Data Transmission Process
False
If time out
False
If the Multicast tree is Broken
Label
true
Multicast Tree Maintenance Process
The node fixes the multicast tree utilizing discovery and respond operations
end

Figure 3. DQMR model.

The DQMR protocol in Figure 3 is therefore designed to function effectively in "both mobility models." The DQMR protocol can be shown as follows in the supplied IoT, which has a source node S_{node} and a multicast group $\mathcal{D}$.

4.5.1. Initialization

Routing table's variables are initialized as follows by each node in the specified IoT environment: Set of last visit nodes $\mathcal{L}_{rt} = \emptyset$, Route cost $C_{rt} = +\infty$

- *Step* 1: Step 2 is to be followed if a node has to establish the tree to the multicast group D and turns into a S_{node}. If not, proceed to step 3.

$$\mathcal{L}_{rreq} = \mathcal{L}_{(S_{node})}, = \eta_{rreq}\eta_{S_{node}}, \complement_{rreq} = \complement_{S_{node}} \tag{36}$$

Proceed to step 4. The fields listed below are included in the RREQ packet:

$$\left\langle \begin{matrix} pac_{type}, hop_{count}, rreq_{id} \\ Multicast_{ip_{address}}, \\ Muticast_{seq_{number}}, \\ Source_{Ip_{address}}, Source_{seq_{number}}, \\ Last_{visit}, next_{visit}, link_{cost}, route_{cost} \end{matrix} \right\rangle \tag{37}$$

4.5.2. Multicast Tree Discovery Process Sending RREQ Process

- *Step* 2: S_{node} needs the position, velocity, way, queue size, and frequencies of the Primary Users p_U information of its neighbors. To predict values $Q_i^*(S_{node}, w)$ for each terminus $d_i \in D$, the S_{node} uses the DQN-MEC perfect to identify the best neighbour w_i^* that has the highest value $Q_i^*(S_{node}, w_i^*)$; for instance, in Figure 4, the S_{node} best neighbors are nodes 3, d_2, 15, and 16, which correspond to destinations d_1, (d_2, d_3) d_4, and d_5. ($\mathscr{L}_{(}S_{node)}$= ($\mathscr{L}_{rt}$=∪{ S_{node}}, the usual of next visit nodes $\eta_{S_{node}} = \{w_{j,}^* \forall dst_i \epsilon D\} \backslash S_{node}$, the list of costs since the S_{node} to every next visit node $\complement_{S_{node}} = \{cost(S_{node}, w_i^*), \forall w_i^* \epsilon\ \eta$, and the route cost ($C_{rt}$=0) are all updated by the S_{node}. Subsequently, the S_{node} creates a "route request (RREQ)" packet, broadcasting it to neighbours, with the contents.

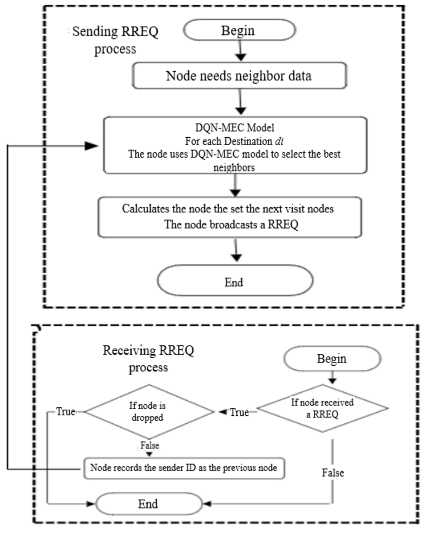
Sending RREQ process
Begin
Node needs neighbor data
DQN-MEC Model
For each Destination di
The node uses DQN-MEC model to select the best neighbors
Calculates the node the set the next visit nodes
The node broadcasts a RREQ
End
Receiving RREQ process
Begin
If node received a RREQ
True
If node is dropped
True
False
Node records the sender ID as the previous node
False
End

Figure 4. Multicast tree discovery process.

4.5.3. Receiving RREQ Process

- *Step* 3: Proceed to Step 3.1 if the node receives an

RREQ. If not, the procedure is terminated.

- *Step* 3.1: If at least one of the following scenarios is true, the RREQ is dropped:

The node is not included in the RREQ's η_{rreq} list.
Route cost C_{rt} in the route table, where cost (w, *node*) is located in the C_{rreq}, is less than or equal to the new cost $C_{rreq}+C(w, node)$.

- *Step* 3.2: Sender's ID is stored by the node as the preceding node. Return to Step 2 now.

4.5.4. Route Reply Process

- *Step* 4: Proceed to step 5 if the node is the *dst*. If not, proceed to step 6.
- *Step* 5: Proceed to step 9 if the dst generates and replies an RREP packet via unicast transmission to the preceding node after receiving an RREQ packet. If not, the procedure is terminated. The fields listed below are included in the RREP packet:

$$\begin{pmatrix} pac_{type}, hop_{count} \\ Multicast_{ip_{address}} \\ Muticast_{seq_{number}} \\ Source_{Ip_{address}} \end{pmatrix} \tag{38}$$

- *Step* 6: In the event that the node gets an RREP packet, it proceeds to Step 7 after appending the contributor to the list of following hops (η) in the route table. If not, the procedure is terminated.
- *Step* 7: Proceed to Step 8 if the node is the S_{node}. In the event that "node v unicasts the RREP packet to the node" before it, go to Step 8.
- *Step* 8: Each node in the resulting EQM tree uses the GT-CTA classical to create an ideal "channel-time slot" schedule that avoids affecting multiple PUs' affected regions and gives the EQM tree the fewest time slots possible for an assumed quantity of channels. For instance, the EQM tree avoids the impacted PU zones and prevents interference linkages by using channels c_1, c_2, c_3 and three time slots t_1, t_2, t_3 proceed to step 9. Figure 5 represents the route reply process.

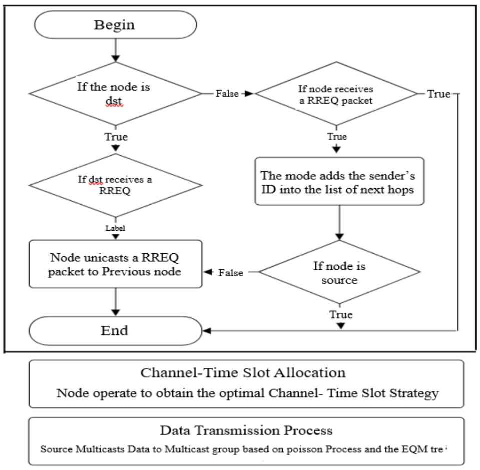
Begin
If the node is dst
False
If node receives a RREQ packet
True
True
True
If dst receives a RREQ
The mode adds the sender's ID into the list of next hops
Label
Node unicasts a RREQ packet to Previous node
False
If node is source
True
End
Channel-Time Slot Allocation
Node operate to obtain the optimal Channel- Time Slot Strategy
Data Transmission Process
Source Multicasts Data to Multicast group based on poisson Process and the EQM tre

Figure 5. Route reply process.

4.5.5. Data Transmission Process

- *Step* 9: Using the best channel-time slot technique and the members of the multicast group's NH, "the source and mobile nodes of the obtained EQM tree" multicast data to them. In particular, the source uses the Poisson process to generate data packets. The data packets are then broadcast by the source to the subsequent hops via "the channel-time slot" technique. When a "node in the EQM tree gets a data packet", it forwards it to the multicast group using the same method that the source used.

4.5.6. Multicast Tree Maintenance Process

A node will create backup routes locally by employing the same technique as the multicast route procedure' source if its existing link to the ensuing hops is broken during the flow and data transfer operations. In particular, the node will need neighbour information in order to compute L_rreq, η_rreq, C_rreq. if it is unable to establish a connection with at least one of the subsequent hops. "The node then generates and sends an RREQ" message to each of its close neighbours. "If a node receives an RREQ" from a different "node and knows the routes to the multicast group", it will reply with an RREP to that node with the goal to build further routes. Node w will keep using the same method as the node to identify alternate "routes to the multicast group" in the event that it receives an RREQ from the node and is unable to determine a route to the multicast group. As a result, this maintenance procedure is limited in nature and only creates a few workarounds for the broken EQM tree.

5. Experimental Results

This section presents the experimentation analysis and performance evaluation of the suggested study plan. This part is divided into three subsections: research overview, comparative analysis, and simulation study.

5.1. Simulation Study

To simulate the proposed research method, Network Simulator version 3.26 (NS3) is utilized. Figure 11 that will be displayed at the end of this paper due to formatting purposes-represents the NS-3 simulation environment.

Table 1. System specifications.

Software specifications	OS	Ubuntu 14.04LTS
	Network simulator	NS-3.26
Hardware specifications	RAM	4 GB
	Hard Disk	500 GB

This tool has an efficient network topology and provides all specifications for the proposed technique. Table 1 indicates the system specifications and Table 2 represents the simulation parameters.

Table 2. Simulation parameters.

Parameters		Descriptions
Network Parameters	No. of IoT devices	50
	Base Station	2
	Edge server	1
Transmission slot parameters	Length of slot	1040 bits
	Duration of slot	8µs
	packet length	830 bits
Parameters of packet	Packet Size	1024
	No. of. Packets	100 bytes
	Packet interval	0.99s
	Data rate	280kbps
	No. of. Retransmission	Max 5
Parameters of energy	Initial energy	0.5J
	Receiving power	47J
	Transmission power	47J
	Data aggregation power	5J
	Battery power	3.3V
Number of. run		1100
Number of rounds		600
Probability of node		0.1
Duration of a single round		18s
Simulation time		150s

Table 3 and Figure 6 the latency in milliseconds (ms) against the number of devices for three different strategies: 2MRLOR, SFLA, and the proposed method. At 10 devices, the latencies are 45 ms, 35 ms, and 20 ms respectively. As the number of devices increases to 20, latencies are 55 ms for 2MRLOR, 45 ms for SFLA, and 22 ms for the proposed method. At 30 devices, the latency for 2MRLOR is 65 ms, SFLA is 55 ms, and the proposed method shows 25 ms. For 40 devices, the latencies recorded are 75 ms, 65 ms, and 30 ms respectively. Finally, with 50 devices, 2MRLOR has a latency of 80 ms, SFLA at 70 ms, and the proposed method at 32 ms, illustrating the superior performance of the proposed method across all scenarios.

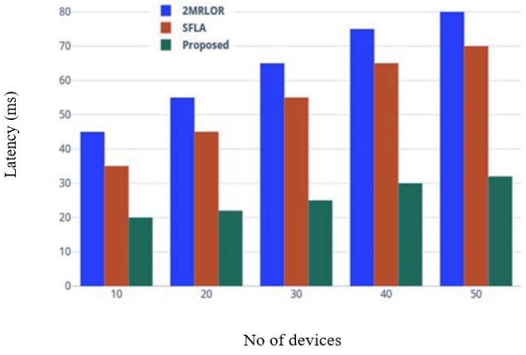

Figure 6. Latency.

5.2. Comparative Analysis

This subsection compares the suggested technique to different existing approaches, including 2MRLOR [30], SFLA [32], Radio Resource Management Policy [33] is performed to evaluate its performance in order.

5.2.1. Comparison of Latency

Latency (€) is well-defined as the amount of delay taken to process or complete a particular request/task. Generally, low latency system will possess high QoS. The mathematical formulation of € is represented as,

$$€ = Tot^t - Com^t \tag{39}$$

Table 3. Numerical results of latency.

X-axis (number of devices)	y-axis latency(ms)		
	2MRLOR	SFLA	Proposed
10	45	35	20
20	55	45	22
30	65	55	25
40	75	65	30
50	80	70	32

As shown in Table 3 above.

5.2.2. Comparison of Energy Efficiency

The goal of energy efficiency is to use the least amount of energy necessary to finish a task or achieve a desired result. This is how the energy efficiency e is determined,

$$\vartheta^e = \vartheta^t - \vartheta^o \tag{40}$$

Where ϑ^t reflects the energy used to carry out packet transmission, which is deducted from the total energy available.

To perform the evaluation, or validation, performance metrics like Latency, Energy efficiency, Traffic Prediction Rate, Throughput, Packet Delivery Ratio (PDR).

Table 4. Numerical results of energy efficiency

X-axis (number of devices)	y-axis energy efficiency (%)		
	RRMP	SFLA	Proposed
10	68	73	75
20	70	75	83
30	73	80	85
40	75	83	90
50	80	85	93

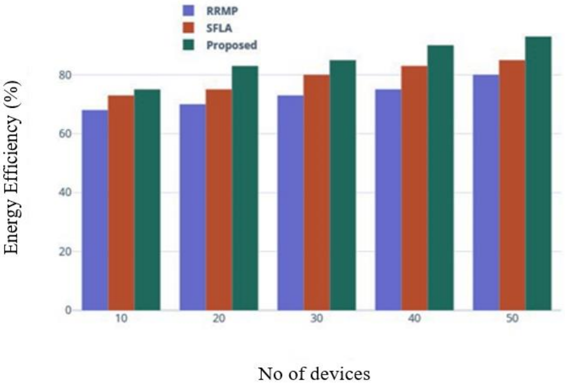

Figure 7. Energy efficiency.

Table 4 and Figure 7 illustrate energy efficiency in percentage (%) against the number of devices for three different strategies: RRMP, SFLA, and the proposed method. With 10 devices, energy efficiencies are 68% for RRMP, 73% for SFLA, and 75% for the proposed method. As the number of devices increases to 20, efficiencies are 70% for RRMP, 75% for SFLA, and 83% for the proposed method. At 30 devices, energy efficiency for RRMP is 73%, SFLA is 80%, and the proposed method achieves 85%. For 40 devices, the efficiencies recorded are 75% for RRMP, 83% for SFLA, and 90% for the proposed method. Lastly, with

50 devices, RRMP demonstrates an energy efficiency of 80%, SFLA at 85%, and the future technique at 93%, importance the superior energy efficiency of the future method transversely all device situations.

5.2.3. Comparison of Traffic Prediction Rate

By utilizing historical network traffic data, network traffic prediction seeks to forecast future network traffic. This can be used as a proactive strategy for planning and network management duties.

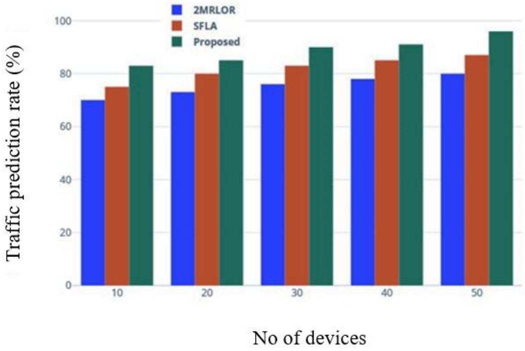

Figure 8. Traffic prediction rate.

At 10 devices, as displayed in Figure 8, the traffic prediction rates are 70% for 2MRLOR, 75% for SFLA, and 83% for the future technique. With 20 strategies, the rates upsurge to 73% for 2MRLOR, 80% for SFLA, and 85% for the suggested technique. At 30 devices, 2MRLOR attains 76%, SFLA influences 83%, and the suggested technique attains 90%. For 40 devices, the prediction rates are 78% for 2MRLOR, 85% for SFLA, and 91% for the proposed method. Finally, with 50 devices, the rates are 80% for 2MRLOR, 87% for SFLA, and an impressive 96% for the proposed method, demonstrating the superior performance of the proposed method in traffic prediction across varying device counts in Table 5 and Figure 9.

Table 5. Numerical results of traffic prediction rate.

X-axis (number of devices)	y-axis traffic prediction rate (%)		
	2MRLOR	SFLA	Proposed
10	70	75	83
20	73	80	85
30	76	83	90
40	78	85	91
50	80	87	96

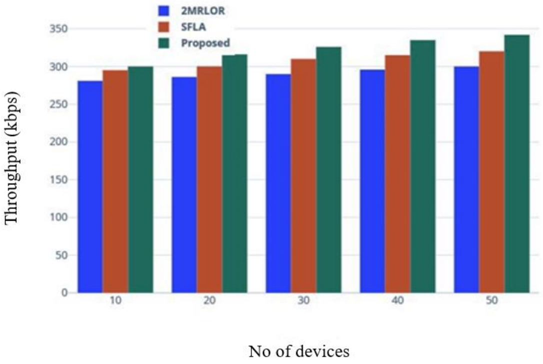

Figure 9. Throughput.

5.2.4. Comparison of Throughput

Throughput (Ť) is defined as the amount of transmission proportion of information from source to destination. A well-being system must have high throughput. Mathematically, Ť can be formulated as,

$$\check{T} = \frac{Tr^d}{Tr^d + 2 \times Pr^d} \times BW \tag{41}$$

Where, Tr^d and Pr^d denotes the transmission delay and propagation delay respectively, and BW denotes the bandwidth.

Table 6 and Figure 9 represents throughput in kilobits per second (kbps) against the number of devices for three different strategies: 2MRLOR, SFLA, and the proposed method. At 10 devices, throughputs are 281 kbps for 2MRLOR, 295 kbps for SFLA, and 300 kbps for the proposed method. With 20 devices, throughputs increase to 286 kbps for 2MRLOR, 300 kbps for SFLA, and 316 kbps for the proposed method. At 30 devices, the throughput for 2MRLOR is 290 kbps, SFLA achieves 310 kbps, and the proposed method reaches 326 kbps. For 40 devices, the throughputs recorded are 296 kbps for 2MRLOR, 315 kbps for SFLA, and 335 kbps for the proposed method. Lastly, with 50 devices, 2MRLOR has a throughput of 300 kbps, SFLA at 320 kbps, and the suggested methods peaks at 342 kbps, demonstrating the suggested methods superior presentation in terms of throughput transversely all device counts.

Table 6: Numerical results of throughput.

X-axis (number of devices)	y-axis throughput (kbps)		
	2MRLOR	SFLA	Proposed
10	281	295	300
20	286	300	316
30	290	310	326
40	296	315	335
50	300	320	342

5.2.5. Comparison of Packet Delivery Ratio (PDR)

The number of packets distributed separated by the total number of packets formed by all nodes is recognized as the PDR. The PDR can be specified in the next way:

$$\check{R} = \frac{\check{D}}{\hat{G}} * 100 \tag{42}$$

In this case, Ř attitudes for the number of delivered packets, Ď for the number of formed packets, and Ĝ for the PDR.

Table 7 and Figure 10 exemplifies the PDR in percentage (%) in contradiction of the number of devices for 3 various policies: 2MRLOR, SFLA, and the proposed method. At 10 devices, PDR are 75% for 2MRLOR, 80% for SFLA, and 83% for the suggested methods. With 20 devices, the ratios are 70% for 2MRLOR, 83% for SFLA, and 85% for the proposed method. At 30 devices, the packet delivery ratio for 2MRLOR is 75%, SFLA achieves 86%, and the proposed method reaches 90%. For 40 devices, the ratios are 78% for 2MRLOR, 85% for SFLA, and 93% for the proposed method. Finally, with 50 devices, the packet delivery ratios are 80% for 2MRLOR, 89% for SFLA, and 96% for the proposed method, showcasing the superior packet delivery performance of the proposed method across all device counts.

Table 7. Numerical results of traffic prediction rate.

X-axis (number of devices)	y-axis packet delivery ratio (%)		
	2MRLOR	**SFLA**	**Proposed**
10	75	80	83
20	70	83	85
30	75	86	90
40	78	85	93
50	80	89	96

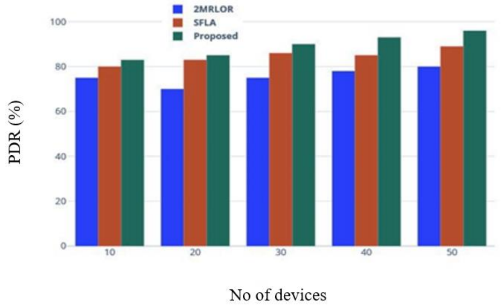

Figure 10. PDR.

6. Research Summary

Initially, we construct a network consisting of 50-Devices, 2-Base Station, 1-Edge Server. Then, we perform Multicast group management using the SOM-PSO. Next, we perform Resource Allocation using Deep Reinforcement Learning. Then, we perform Predictive Analytics for Multicast Traffic Demand using the DTA-MLC Next, we implement the Enhanced Edge Catching strategies using the Context-aware Long-Short-Term Model with graph neural networks (C-ALSTM-GNN) to forecast content demand. Then, we perform Multicast routing protocol using the Efficient QoS Multicast (EQM). Finally, we plot graph for the following metrics are No of Devices vs. Latency (ms), No of Devices vs. Energy efficiency (%), No of Devices vs. Traffic prediction rate (%), No of Devices vs. Throughput (kbps) and No of Devices vs. packet delivery ratio (%) Figures 7 to 11 and Tables 3 to 7.

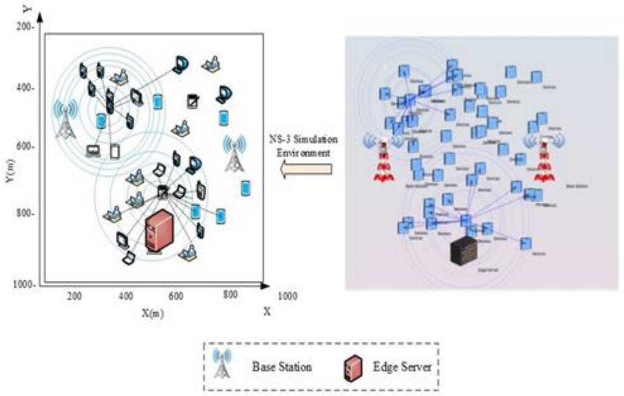

Figure 11. NS-3 simulation environment.

The network simulator version 3, or NS-3, has an efficient network topology and provides all specifications for the proposed technique.

7. Conclusions

In conclusion, foremost goal and scope of this research is to develop and validate the AI-driven multicast approaches for improving the network performance in 5G systems we use the SOM-PSO to handle Multicast groups. Next, we use Deep Reinforcement Learning for Resource Allocation. Next, we use Multi-Link Communication and the DTA-MLC Next, we use the Context-aware Long-Short-Term Model with Graph Neural Networks (C-ALSTM-GNN) to estimate content demand and execute the enhanced edge catching techniques. Next, we use the EQM to carry out the multicast routing protocol. The proposed achieves latency with 32 ms, energy efficiency with 93%, traffic prediction rate 96%, throughput with 342 kbps and PDR with 96%.

References

[1] Asghar M., Memon S., and Hämäläinen J., "Evolution of Wireless Communication to 6g: Potential Applications and Research Directions," *Sustainability*, vol. 14, no. 10, pp. 6356, 2022. https://doi.org/10.3390/su14106356

[2] Aslam S., Alam F., Hasan S., and Rashid M., "A Machine Learning Approach to Enhance the Performance of D2D-Enabled Clustered Networks," *IEEE Access*, vol. 9, pp. 16114-16132, 2021. DOI:10.1109/ACCESS.2021.3053045

[3] Bhattacharjee S., Acharya T., and Bhattacharya U., "Cognitive Radio-Based Spectrum Sharing Models for Multicasting in 5G Cellular Networks: A Survey," *Computer Networks*, vol. 208, pp. 108870, 2022. https://doi.org/10.1016/j.comnet.2022.108870

[4] Cao J., Zhu X., Sun S., Kurniawan E., and Boonkajay A., "Risk-Aware and Energy-Efficient AoI Optimization for Multi-Connectivity WNCS with Short Packet Transmissions," *IEEE Internet of Things Journal*, vol. 11, no. 12, pp. 21474-21485, 2024. DOI:10.1109/JIOT.2024.3375805

[5] Chen Z., Xu M., She C., Jia Y., Wang M., and Li Y., "Improving Timeliness-Fidelity Tradeoff in Wireless Sensor Networks: Waiting for All and Waiting for Partial Sensor Nodes," *IEEE Transactions on Communications*, vol. 71, no. 7, pp. 4151-4164, 2023. DOI:10.1109/TCOMM.2023.3277001

[6] Chukhno N., Chukhno O., Pizzi S., Molinaro A., Iera A., and Araniti G., "Efficient Management of Multicast Traffic in Directional mmWave Networks," *IEEE Transactions on Broadcasting*, vol. 67, no. 3, pp. 593-605, 2021.

DOI:10.1109/TBC.2021.3061979

[7] Dao N., Tu N., Hoang T., Nguyen T., Nguyen L., Lee K., Park L., Na W., and Cho S., "A Review on New Technologies in 3GPP Standards for 5G Access and Beyond," *Computer Networks*, vol. 245, pp. 110370, 2024. https://doi.org/10.1016/j.comnet.2024.110370

[8] De la Fuente A., Leal R., and Armada A., "Resource Allocation Management for Broadcast/Multicast Services," *Resource*, vol. 10, no. 6, pp. 1-4, 2015. https://core.ac.uk/download/30276883.pdf

[9] Elouafadi R., El Fenni M., and Benjillali M., "NOMA Clustering for Improved Multicast IoT Schemes," *Journal of Sensor and Actuator Networks*, vol. 11, no. 2, pp. 1-14, 2022. https://doi.org/10.3390/jsan11020026

[10] Hao H., Xu C., Yang S., Zhong L., and Muntean G., "Multicast-Aware Optimization for Resource Allocation with Edge Computing and Caching," *Journal of Network and Computer Applications*, vol. 193, pp. 103195, 2021. https://doi.org/10.1016/j.jnca.2021.103195

[11] Hashemi M. and Moghim N., "An Efficient Multicast Multi-Rate Reinforcement Learning Based Opportunistic Routing Algorithm," *Multimedia Tools and Applications*, vol. 82, pp. 26613-26630, 2023. https://doi.org/10.1007/s11042-023-14645-1

[12] Hewage C., Ahmad A., Mallikarachchi T., Barman N., and Martini M., "Measuring, Modeling and Integrating Time-Varying Video Quality in End-to-End Multimedia Service Delivery: A Review and Open Challenges," *IEEE Access*, vol. 10, pp. 60267-60293, 2022. DOI:10.1109/ACCESS.2022.3180491

[13] Imam-Fulani Y., Faruk N., Sowande O., Abdulkarim A., Alozie E., Usman A., Adewole K., Oloyede A., Chiroma H., and Garba S., "5G Frequency Standardization, Technologies, Channel Models, and Network Deployment: Advances, Challenges, and Future Directions," *Sustainability*, vol. 15, no. 6, pp. 5173, 2023. https://doi.org/10.3390/su15065173

[14] Jayaraman R., Manickam B., Annamalai S., Kumar M., Mishra A., and Shrestha R. "Effective Resource Allocation Technique to Improve QoS in 5G Wireless Network," *Electronics*, vol. 12, no. 2, pp. 451, 2023. https://doi.org/10.3390/electronics12020451

[15] Kabir H., Tham M., and Chang Y., "Internet of Robotic Things for Mobile Robots: Concepts, Technologies, Challenges, Applications, and Future Directions," *Digital Communications and Networks*, vol. 9, no. 6, pp. 1265-1290, 2023. https://doi.org/10.1016/j.dcan.2023.05.006

[16] Krasilov A., Lebedeva I., Yusupov R., and Khorov E., "Resource-Efficient Multicast URLLC Service in 5G Systems," *Sensors*, vol. 24, no. 8, pp. 2536, 2024. https://doi.org/10.3390/s24082536

[17] Li H., Tang L., Chen S., Zheng L., and Zhong S., "AoI-Aware Resource Scheduling for Industrial IoT with Deep Reinforcement Learning," *Electronics*, vol. 13, no. 6, pp. 1104, 2024. https://doi.org/10.3390/electronics13061104

[18] Lin H., Lin K., and Wei H., "Adaptive Age of Information Optimization in Rateless Coding-Based Multicast-Enabled Sensor Networks," *IEEE Journal of Selected Areas in Sensors*, vol. 1, pp. 73-92, 2024. DOI:10.1109/JSAS.2024.3407689

[19] Liu H., Lin K., and Wei H., "Improving IoT Device Power Efficiency: Discontinuous Reception for Mixed Traffic in Multicast and Broadcast Services," *IEEE Internet of Things Journal*, vol. 11, no. 10, pp. 18005-18019, 2024. DOI:10.1109/JIOT.2024.3361012

[20] Ma R., Cao J., Zhang Y., Shang C., Xiong L., and Li H., "A Group-Based Multicast Service Authentication and Data Transmission Scheme for 5G-V2X," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 12, pp. 23976-23992, 2022. DOI:10.1109/TITS.2022.3197767

[21] Mach P. and Becvar Z., "Device-to-Device Relaying: Optimization, Performance Perspectives, and Open Challenges Towards 6G Networks," *IEEE Communications Surveys and Tutorials*, vol. 24, no. 3, pp. 1336-1393, 2022. DOI:10.1109/COMST.2022.3180887

[22] Mahdi W. and Taşpınar N., "Resource Allocation Based on SFLA Algorithm for D2D Multicast Communications," *Computer Systems Science and Engineering*, vol. 45, no. 2, pp. 1517-1530, 2023. https://doi.org/10.32604/csse.2023.030069

[23] Mishra S. and Tyagi A., *Artificial Intelligence-Based Internet of Things Systems*, Springer, 2022. https://doi.org/10.1007/978-3-030-87059-1_4

[24] Mustafa E., Shuja J., Zaman S., Jehangiri A., Din S., Rehman F., Mustafa S., Maqsood T., and Khan A., "Joint Wireless Power Transfer and Task Offloading in Mobile Edge Computing: A Survey," *Cluster Computing*, vol. 25, no. 4, pp. 2429-2448, 2022. https://doi.org/10.1007/s10586-021-03376-3

[25] Ouyang R., Xiong X., Fu M., Wang J., Chen S., and Alfarraj O., "A Scalable Video Multicast Scheme Based on User Demand Perception and D2D Communication," *Sensors*, vol. 23, no. 17, pp. 7325, 2023. https://doi.org/10.3390/s23177325

[26] Pupo E., González C., Montalban J., Angueira P., Murroni M., and Iradier E., "Artificial Intelligence Aided Low Complexity RRM Algorithms for 5G-MBS," *IEEE Transactions on Broadcasting*, vol.

70, no. 1, pp. 110-122, DOI:10.1109/TBC.2023.3311337

[27] Tan X., Li S., Wang S., Liu Y., Zheng Q., and Yang J., "Cooperative Bargaining Game Based Adaptive Video Multicast over Mobile Edge Networks," *IEEE Transactions on Multimedia*, vol. 26, pp. 2380-2394, 2024. DOI:10.1109/TMM.2023.3295569

[28] Tripathi V., Talak R., and Modiano E., "Information Freshness in Multihop Wireless Networks," *IEEE/ACM Transactions on Networking*, vol. 24, no. 3, pp. 1336-1393, 2022. DOI:10.1109/COMST.2022.3180887

[29] Trung N. and Anh N., "Beamforming-as-a-Service for Multicast and Broadcast Services in 5G Systems and Beyond," *IEEE Access*, vol. 11, pp. 142794-142815, 2023. DOI:10.1109/ACCESS.2023.3343523

[30] Vaezi M., Azari A., Khosravirad S., Shirvanimoghaddam M., Azari, M., and Chasaki D., "Cellular, Wide-Area, and Non-Terrestrial IoT: A Survey on 5G Advances and the Road toward 6G," *IEEE Communications Surveys and Tutorials*, vol. 24, no. 2, pp. 1117-1174, 2022. DOI:10.1109/COMST.2022.3151028

[31] Wang T., Wang S., Lan X., Liu Y., Chen Q., and Xiao P., "On the AoI-Aware Status Update in Buffer-Aided Wireless Powered Internet of Things Network," *IEEE Internet of Things Journal*, vol. 11, no. 7, pp. 12551-12566, 2024. DOI:10.1109/JIOT.2023.3334820

[32] Zhou M., Li J., Yuan J., Xie M., Tan W., Yin R., and Yang L., "An Architecture for AoI and Cache Hybrid Multicast/Unicast/D2D with Cell-Free Massive MIMO Systems," *IEEE Access*, vol. 11, pp. 43080-43088, 2023. DOI:10.1109/ACCESS.2023.3271519

[33] Zuhra S., Chaporkar P., Karandikar A., and Poor, H., "Multi-Connectivity for Multicast Video Streaming in Cellular Networks," *Network*, vol. 4, no. 2, pp. 175-195, 2024. https://doi.org/10.3390/network4020009

Wael AlZoubi holds a Ph.D. of Computer Sciences from National University of Malaysia in 2013. He also received his B.Sc. and M.Sc. in Computer Science from Yarmouk University, Jordan in 2000 and 2004, respectively. He is currently an Assistant Professor at Computer Science Department in Al-Balqa Applied University, Ajloun, Jordan. His research interests include Meta-Heuristics, Global Optimization, Machine Learning, Data Mining, Bioinformatics, Graph Theory and Parallel Programming. He has published over 20 papers in international journals and conferences.

Hazem Hatamleh is an Associate Professor of Computers, Computing System and Network at the Applied Science Department, Ajloun University College, Al-Balqa Applied University in Jordan. He holds a Ph.D. in engineering sciences from the National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute," in 2007, with over 23 years of experience in teaching training and research. He has held various academic and administrative positions and contributed to curriculum development. His current research interests include Computer Networks, Wireless Networks, the IoT, Image Processing, and Computer Graphics.