SvedbergOpen
DISSEMINATION OF KNOWLEDGE
International Journal of Artificial Intelligence and Machine Learning
Publisher's Home Page: https://www.svedbergopen.com/
Research Paper
Open Access

Enabling Critical Remote Services Through Intelligent AI-Based Traffic Management In 5G Satellite Networks

Arju Malik[a*], Suraj Malik[b], Vaibhav Sharma[c]

[a] IIMT University, Meerut, Uttar Pradesh,250001, India. Email: arjumalik_cs127@iimtindia.net.

[b] IIMT University, Meerut, Uttar Pradesh,250001, India. Email: surajmalik@iimtindia.net.

[c] IIMT University, Meerut, Uttar Pradesh,250001, India. Email: dr.vsharma.ee@gmail.com.

Abstract
The adoption of 5G satellite networks is central to the implementation of ubiquitous connectivity and the provision of critical remote services, including autonomous systems, telemedicine, and industrial IoT, which have very strict and heterogeneous Quality of Service (QoS) requirements. Nevertheless, due to the volatile nature of the satellite space, which is associated with changing topologies and limited resources, conventional traffic management schemes are no longer effective, which causes congestion, spikes in latency, and compromised reliability. This paper is meant to discuss a new, smart system of traffic management based on a hybrid Artificial Intelligence (AI) architecture to overcome the challenge. This fundamental innovation is a load-balancing algorithm that is synergistic (combining a Long Short-Term Memory (LSTM) network to predictive congestion forecasting with a Proximal Policy Optimization (PPO)-based Deep Reinforcement Learning (DRL) agent to perform adaptive, real-time distribution of traffic. This proactive-adaptive core is complemented by a service-aware orchestration module that implements differentiated QoS policies of Ultra-Reliable Low-Latency Communication (URLLC), enhanced Mobile Broadband (eMBB), and massive Machine-Type Communication (mMTC) traffic classes. The proposed framework achieves a significant improvement in performance over state-of-the-art criteria through the extensive simulation of a Low Earth Orbit (LEO) satellite constellation. These findings demonstrate that the end-to-end latency is reduced by 42.3 percent, the rate of packet loss is reduced by 67.8 percent, and the network throughput is also increased by 31.5 percent during a high-load condition, and also the SLA boundaries are met in all the critical service classes. This article confirms that a hybrid, service-conscious AI model is needed to transform satellite connections into an additional fabric that is deterministic and high-performance to support the next generation of remote services.

Keywords: AI-based load balancing; 5G satellite networks; quality of service (QoS); LSTM-DRL hybrid model; service-aware orchestration.

1. Introduction

The relentless expansion of digital connectivity necessitates worldwide, dependable and efficient network access, being available not only to the coverage area of the land infrastructure but also to distant areas, the sea, and air spaces. Non-Terrestrial Networks (NTNs) (satellite constellations), are explicitly regarded as the building blocks of the fifth-generation (5G) and nascent sixth-generation (6G) of communication paradigms to support the vision of seamless global coverage [5, 23]. Such satellite networks will form the foundation of essential remote services (including autonomous transportation, telemedicine and remote surgery, large-scale industrial IoT, and disaster response communications), which present very demanding, heterogeneous needs including ultra-reliable low-latency communication (URLLC), enhanced mobile broadband (eMBB), and massive machine-type communication (mMTC) [7, 10]. Nevertheless, such properties of satellite networks as large propagation delays, dynamism in topology will be provided by orbital mechanics, limited and unpredictable bandwidth, and intermittent availability of links present a singularly difficult situation in ensuring stable Quality of Service (QoS) delivery [2, 9].

Traffic control here is a key challenge when it is a volatile situation. The classical, static, or heuristic-driven resources allocation and routing algorithms fail radically, as they are unable to change their state in real-time or anticipate congestion elimination before it starts [16, 27]. As a result, the performance of the network becomes poor, causing more latency, loss of packets, and jitter which directly compromises the reliability of the critical services which the networks are meant to deliver [3, 29]. Such disparity between the possibilities of satellite-enabled connectivity and the promise of deterministic service quality forms the main research problem in next-generation wireless communications.

AI and ML have become one of the potent tools to optimize the network providing data-oriented optimization methods to predict, make decisions and control variables. Recent surveys provide a detailed overview of the game changer capacity of AI in satellite communications that include applications in traffic prediction, beam control, and resource coordination [1, 6, 19]. Reinforcement Learning (RL) and Deep RL (DRL) have, in particular, been found to be able to change according to network dynamics in order to solve network-related tasks (such as routing and scheduling) [12, 13, 24]. However, there still is a big research gap. Current methods usually exist independently: predictive models (e.g. LSTMs) can predict congestion, but lack control mechanisms (e.g. embedded control) [25], whereas reactive RL agents are only taught to optimally learn policies after a disturbance in the network, which means they trade off performance during exploration [4, 28]. A comprehensive service aware AI architecture that integrates predictive intelligence into proactive mitigation and adaptive control into real-time optimization, with differentiated QoS being enforced on heterogeneous critical traffic is still lacking.

In a bid to fill this gap, this paper presents a 5G satellite network traffic management framework that is innovative and intelligent. The main idea our hybrid AI-based load-balancing algorithm incorporates within itself is a hybrid algorithm that combines both an Artificial Neural Network (ANN) load-balancing algorithm with predictive congestion forecasting and a Deep Reinforcement Learning (DRL) agent with adaptive traffic distribution. The framework is complemented with a service-conscious orchestration module that guarantees proper compliance with SLA of various services of critical nature. The main goals of this study are triple in nature: (1) design and development of this hybrid ANN-RL algorithm of proactive and adaptive steering of traffic; (2) an implementation of service-differentiated traffic management policy, which assigns precedence to QoS of URLLC, eMBB, and mMTC service classes; and (3) the validation of the superiority of this framework in improving key network performance metrics, such as latency, throughput, packet loss, and jitter through the comprehensive simulation of the base against the cutting-edge benchmarks.

2. Literature Reviews

The use of Artificial Intelligence (AI) in satellite communications has been evolving greatly, reflecting the further development of the field of machine learning. The evolution of the research has been to classical and supervised prediction methods to advanced deep learning and reinforcement learning models that can be used to control and coordinate non-terrestrial networks (NTNs) that are highly dynamic [1, 2]. This review provides a timeline of this development, including important technical changes, representative practices, and the unchanging challenges.

2.1. Foundational Stage: Supervised Learning for Prediction and Awareness

- The first stage was aimed at creating network awareness through supervised learning. The first one was good traffic and topology prediction to guide the static or heuristic resource managers in the software-defined satellite network (SDSN) architectures [5, 6].
- Techniques and Applications: The studies used the traditional machine learning and early neural networks to predict the link state, the availability of the gateway, and the traffic. This offered vital contributions towards effective Virtual Network Function (VNF) location and allocation of resources in SDN/NFV implementations [5].
- Role: These models were used to provide a base-level sensing, and it minimized uncertainty in the management systems at higher levels. As an example, the use of LSTM networks was effective in time-series traffic prediction, which can be used to provide bandwidth proactively [7, 8].

2.2. The Transition to Complexity: Deep Learning for Feature Extraction and Advanced Prediction

The nature of challenges of simple predictors was revealed by increasing network scale (mega-constellations), mobility, and service heterogeneity. That motivated the use of deep learning (DL) models to process complicated high-dimensional information and obtain richer features to use in decision-making [2, 7].

- Technique Advancements: In addition to LSTMs on sequence data, Graph Neural Networks (GNNs) and attention came into use to capture the complex topological interactions and time-dependent graph forms of satellite constellations [10, 11].
- •Functional Evolution DL models have been developed as feature extraction front-ends, as opposed to pure predictors. They took raw network graph information and measurements and used it to construct informative state to the downstream optimization algorithms, especially in activities such as network slicing and topology-aware routing [10, 11].

2.3. The Shift to Autonomous Control: Reinforcement Learning for Dynamic Optimization

The fundamental paradigm shift concept was the departure of passive prediction to active and adaptive control with the help of Reinforcement Learning (RL). The interactiveness of RL enabled it to learn optimal policies in non-stationary satellite environments (e.g., routing, offloading), making it suitable in combinatorial optimization problems [12, 13].

Algorithmic Progression:

- Classical RL: Tabular Q-learning has been used in early work to model routing as a Markov Decision Process (MDP), with better throughput and delay than a simple set of rules [12].
- Deep RL (DRL): Deep Q-Networks (DQN) and Double DQN were used to deal with large state spaces to enhance tasks such as traffic offloading in Space-Air-Ground Integrated Networks (SAGIN), which enhances resilience and suppressed packet loss [17].
- Policy Gradient & Actor-Critic Models: In the case of continuous or hybrid action space (e.g. power allocation, precise scheduling), such algorithms as Asynchronous Advantage Actor-Critic (A3C) and Proximal Policy Optimization (PPO) dominated. They facilitated resource management of fine grained latency assured resources [13, 10].
- Multi-Agent DRL (MADRL): The recent development is concerned with the distributed control scalability. The coordinated task of satellite-UAV trajectory co-design, distributed data offloading, and others is now studied using Cooperative Multi-Agent PPO (MAPPO) and other platforms, instead of a centralized approach to intelligence processing [14, 18].

2.4. Performance Gains and the Emergence of Hybrid AI Architectures

Empirical studies confirm the performance evolution. While supervised predictors improved mapping accuracy and reduced delay [5, 6], RL-based control reported substantial gains: multi-agent DRL for SAT-UAV networks achieved up to ~2x throughput and energy efficiency [14], and latency-aware A3C schedulers improved delivery ratios by over 20% [13]. This progression has led to the current state-of-the-art: hybrid AI architectures. These systems combine DL components (e.g., GNNs for state encoding, LSTMs for prediction) with RL agents (e.g., Actor-Critic for decision-making), creating integrated frameworks for holistic network optimization [10, 11, 19].

2.5. Persistent Research Gaps and Future Evolution

In spite of such technical evolution, there are still critical issues, which refer to the further development stages:

• Simulation to Deployment: There is a large disparity between simulation-based validation and in-orbit deployment, which is limited by the onboard computer capacity and the requirement of radiation-hardened AI hardware [8].

• To Strong and Generalizable AI: The community struggles with the lack of data, limited satellite dwell times that do not allow learning convergence, and the absence of models that can be used on various orbital regimes and traffic patterns [3, 7, 21].

• Industrial connections with 6G Ecosystems: The next-generation needs to integrate and standardize AI/ML modules of a terrestrial 5G/6G core network to manage intelligent resources end to end [2, 22].

In the literature, one can find a clear evolution development, out of supervised prediction, to deep feature extraction, to reinforcement control, to hybrid and multi-agent orchestration. With this development, there is a strong technical basis of further research, including AI-based load balancing, that naturally demands the predictive nature of DL and the adaptive decision-making of RL to effectively deal with the dynamic 5G satellite network environment.

2.6. Research Gap

Although Artificial Intelligence (AI) and Machine Learning (ML) have become the central focus of the management of next-generation 5G and 6G non-terrestrial networks (NTNs) [1, 6, 19], there remains a research gap in developing integrated service-aware AI frameworks to operate the real-time management of 5G satellite networks. Available literature offers quite a good background, discussing AI in relation to particular tasks such as traffic prediction [16, 25], routing in mega-constellations [2], latency optimization [3, 29] and dynamic offloading in SAGIN [4]. The enabling technologies and future challenges are comprehensively mapped by the surveys [1, 10, 26].

Nevertheless, existing methods do not always take into account a holistic view of stringent and heterogeneous Quality of Service (QoS) needs of critical remote services including autonomous systems, remote medical procedures, and industrial IoT, which require ultra-reliable and low-latency communication (URLLC) and assured throughput [7, 23]. Studies are inclined to maximize individual network design aspects (e.g., latency [3], bandwidth [25]) or a given segment (e.g., satellite-to-UAV links [12, 17]) without a unified system of intelligent, predictive traffic control, which will intelligently distribute traffic to prevent congestion and end-to-end service-level agreements (SLAs) across the integrated satellite-terrestrial space [11, 28]. Also, the fact that the implementation of such AI models is a practical challenge in space environments that are resource-constrained is a major obstacle [18]. Hence, an urgency to implement a new hybrid AI-based traffic management architecture, service differentiated and aware of the special dynamics of 5G satellite networks, exists between the current AI potential and the deployable performance of AI on reliable networks capable of critical applications.

2.7. Problem Statement

The 5G satellite networks will also be required to integrate in order to offer ubiquitous connectivity, particularly in the isolated and underserved regions [14, 16]. Nevertheless, the nature of satellite networks such as long propagation delays, dynamic topology as a result of satellite mobility, limited and unreliable bandwidth, and intermittent link connectivity makes them a very volatile platform with respect to traffic management [2, 9]. Such volatility is essentially a contradiction to the strict requirements of reliability, latency and throughput of the new critical remote services [7, 23].

The conventional, non-dynamic, or reactive load-balancing and resource allocation patterns fail in this regard. They have no ability to adapt dynamically to sudden changes in network state (link quality, node load, traffic patterns) or to preemptionally resource allocation to satisfy predictive requirements of the various service classes [4, 13]. As a result, networks experience congestion, poor QoS, service outages, and the inability to ensure the performance of mission-critical applications, which constrains the transformative power of remote healthcare using satellites, autonomous operations, and massive IoT networks [5, 10].

The main issue is that smart, predictive and adaptive traffic control system of 5G satellites has not been developed that can identify the network state in real-time, predict the appearance of congestion, and make the most reasonable decisions to allocate the traffic to meet the dynamic needs of multiple and complex requirements of all the connected critical services.

2.8. Research Objectives

In order to fill the identified gap and resolve the mentioned problem, the present research will outline and test an intelligent AI-based traffic management framework of 5G satellite networks. The specific objectives are:

- To create and develop a hybrid AI-based load-balancing strategy that will use the predictive power of Artificial Neural Networks (ANNs) to predict traffic trends and congestion areas and use the adaptive capability of a Reinforcement Learning (RL) agent to optimally distribute traffic in real-time across both satellite nodes and ground sections.
- To deploy a service-conscious traffic orchestration module, which liaises with the core AI algorithm and sorts and prioritizes traffic based on important remote services (e.g., autonomous systems, telemedicine, industrial IoT), to enforce differentiated management policies to ensure their respective QoS goals of latency, jitter, throughput and packet loss.
- To assess the effectiveness of proposed framework with respect to the baseline and state-of-the-art techniques (e.g., [3, 4, 24, 28]) in the major metrics, such as end-to-end latency, network throughput, packet delivery ratio, jitter and resource utilization effectiveness of the proposed framework under various and dynamic conditions of loads.

3. Materials And Methods

This section delineates the methodological framework designed to achieve the research objectives. The proposed solution is an integrated AI-driven traffic management system for 5G satellite networks, comprising a hybrid ANN-RL load balancer and a service-aware orchestration module, evaluated within a simulated satellite-terrestrial environment.

3.1. System Model and Architecture

The network model is based on a dynamic, multi-layered Non-Terrestrial Network (NTN) architecture integrated with a terrestrial 5G core, as conceptualized in the 5G-ALLSTAR vision [14] and aligned with the ITU framework for integrated networks [23].

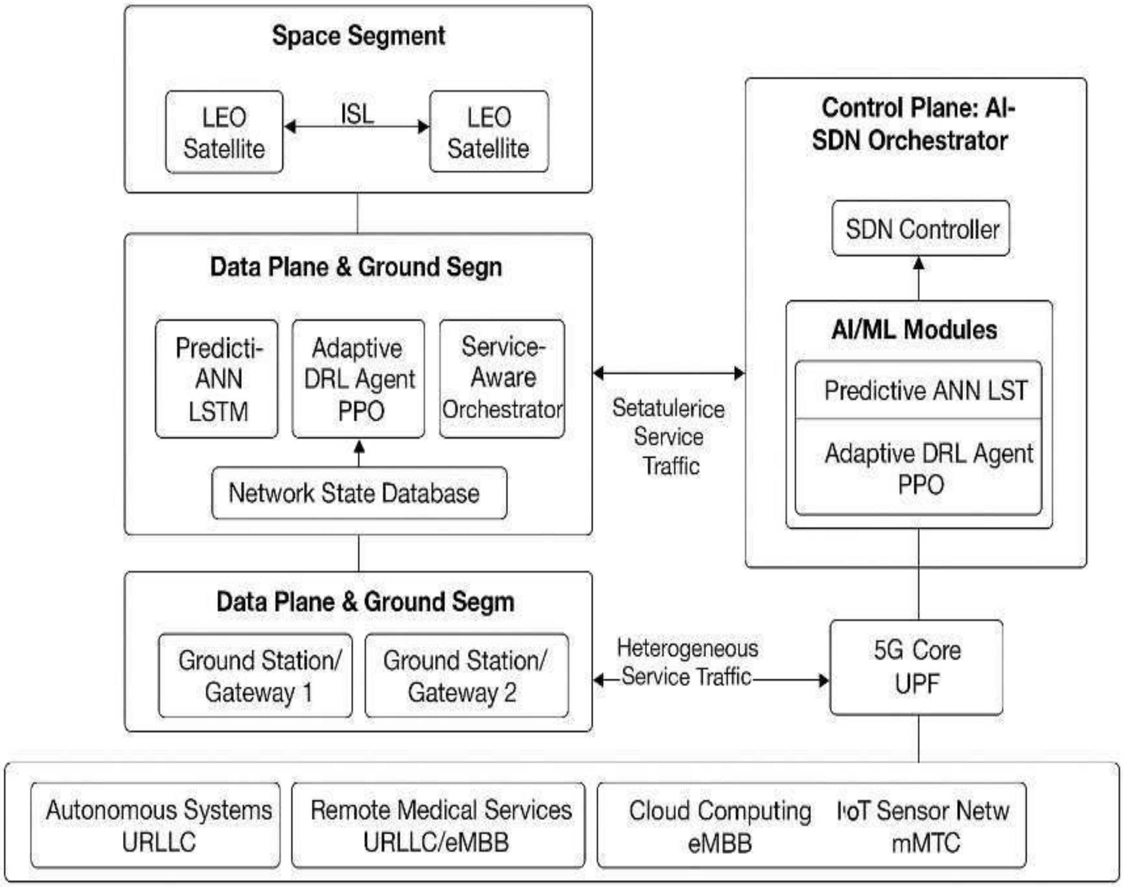

Figure 1. AI-Driven System Architecture for 5G Satellite Network Traffic Management

The architectural design is composed of:

- Space Segment: The constellation of Low Earth Orbit (LEO) satellites based on mega-constellation scenarios [2, 9], providing inter-satellite links (ISLs) and time-varying connection with ground stations.
- Control Plane: A physically distributed Software-Defined Networking (SDN) controller that is logically centralized, that is, adheres to SDN/NFV-based principles of management [16]. This controller includes the fundamental AI components and has a global perspective of the network state, such as the loads in the satellite links, queue state and path latencies.
- Data Plane: Ground gateways and satellite nodes that pass traffic in accordance with flow directives issued by the control plane.
- Traffic Model: Heterogeneous traffic is created, models critical services: ultra-reliable low-latency communication (URLLC) in autonomous systems and telemedicine [7], enhanced mobile broadband (eMBB) in cloud computing and massive machine-type communication (mMTC) in IoT sensor networks [10].

3.2. Hybrid AI-Based Load-Balancing Algorithm

The hybrid AI model is designed in two stages to be a predictive and adaptive steering of traffic.

Stage 1: Predictive Forecasting using Artificial Neural Networks (ANNs): Long Short-term Memory (LSTM) network is used as the predictive engine. The LSTM model, based on its performance in time-series prediction of traffic to reduce bandwidth [25] and gateway selection [27] is trained using historical and real-time telemetry information. The input characteristics consist of time-series data of used links per satellite node, the rate of traffic arrival at the gateways, and the positional information of satellites. The model is used to predict the future load in the short term as well as detect possible congestion areas along the network topology [4]. This predictive output constitutes an essential part of the state representation of the RL agent, shifting the system of reactive to a proactive management mode.

Stage 2: Adaptive Decision-Making with Deep Reinforcement Learning (DRL): The Deep Reinforcement Learning agent, namely, a Proximal Policy Optimization (PPO) actor-critic model is selected because of its stability and ability to manage continuous or high-dimensional action spaces in the networking task [13, 28]. The agent is informed of the optimum policy to use in the distribution of traffic.

- State Space (S): The state s_t is the snapshot of the network (current bandwidth per link, queue lengths, active flows) in which the LSTM predicts the congestion map, and it forms a fully representational state S = {current metrics, predicted metrics).
- Action Space (A): The action of the agent a_t is the traffic-splitting ratios of the incoming flows of flows through many different paths (e.g., through many different hops between satellites or ground gateways). This allows the distribution of loads finely.
- Reward Function (R): The reward is designed to be efficient in balancing the network efficiency and QoS, based on the latency-aware scheduling concepts [29]. It is a weighted function:
  R = w1*Δ(Throughput) - w2*Δ(Latency) - w3*Δ(Packet_Loss) - w4*(Predicted_Congestion_Penalty).(1)

The inclusion of a penalty based on the LSTM's congestion prediction guides the agent towards pre-emptive avoidance strategies.

3.2.1. Proposed Algorithm: Hybrid AI Load Balancer

```
Algorithm MAIN_HYBRID_LOAD_BALANCER(network_state, historical_data)
    Input:
        network_state: Current network metrics (link_utilization, queue_lengths,
                                   active_flows, satellite_positions)
        historical_data: Time-series network data for past T time steps

    Output:
        action: Traffic splitting ratios across K available paths

    // Phase 1: Predictive Forecasting with LSTM
    predicted_congestion_map ← LSTM_PREDICTOR(historical_data, network_state)

    // Phase 2: State Representation Construction
    extended_state ← CONSTRUCT_STATE(network_state, predicted_congestion_map)

    // Phase 3: Adaptive Decision-Making with PPO Agent
    action ← PPO_AGENT.act(extended_state)

    // Phase 4: Apply Action to Network
    APPLY_TRAFFIC_SPLITTING(action)
    return action
// COMPONENT 1: LSTM PREDICTOR MODULE
Function LSTM_PREDICTOR(historical_data, current_state)
    Input:
        historical_data: Sequence of T past states [s_{t-T}, ..., s_{t-1}]
        current_state: Current network state s_t
    Output:
        predicted_map: Congestion probability for each network node/link

    // Normalize input data
    normalized_seq ← NORMALIZE(historical_data ∪ current_state)

    // Extract temporal features
    temporal_features ← LSTM_ENCODER(normalized_seq)
```

```
        // Generate congestion predictions
        for each node in network_nodes do
                node_features ← EXTRACT_NODE_FEATURES(temporal_features, node)
                congestion_prob[node] ← DENSE_LAYER(node_features)
        end for
        predicted_map ← CREATE_CONGESTION_MAP(congestion_prob)
        return predicted_map
// COMPONENT 2: STATE CONSTRUCTION
Function CONSTRUCT_STATE(current_state, predicted_map)
        Input:
                current_state: Real-time network metrics
                predicted_map: LSTM output congestion probabilities
        Output:
                extended_state: Combined state representation
        // Extract current metrics
        current_features ← [
                current_state.link_utilization,
                current_state.queue_lengths,
                current_state.active_flows,
                current_state.satellite_positions,
                current_state.link_delays
        ]

        // Extract prediction features
        prediction_features ← [
                predicted_map.congestion_probabilities,
                predicted_map.hotspot_locations,
                predicted_map.time_to_congestion
        ]

        // Combine features
        extended_state ← CONCATENATE(current_features, prediction_features)

        // Normalize for DRL input
        extended_state ← MIN_MAX_NORMALIZE(extended_state)
        return extended_state
COMPONENT 3: PPO AGENT DECISION MAKING
Class PPO_AGENT
        // Initialize actor and critic networks
        actor_network ← INIT_NEURAL_NETWORK(input_dim, output_dim)
        critic_network ← INIT_NEURAL_NETWORK(input_dim, 1)
        buffer ← INIT_REPLAY_BUFFER(capacity=N)
        Function act(state)
                // Actor network outputs action probabilities
                action_probs ← actor_network.forward(state)

                // Sample action from distribution
                action_distribution ← Categorical(action_probs)
                action ← action_distribution.sample()

                // Store for training
                buffer.store(state, action, action_probs[action])
                return action
        Function update()
```

```
        // Sample batch from replay buffer
        batch_states, batch_actions, batch_old_probs ← buffer.sample_batch(batch_size)

        // Calculate advantages
        batch_values ← critic_network.forward(batch_states)
        batch_advantages ← CALCULATE_ADVANTAGES(batch_values, buffer.rewards)
        // PPO loss calculation
        for epoch in 1 to K do
            // Get new action probabilities
            new_probs ← actor_network.forward(batch_states)

            // Calculate probability ratio
            prob_ratio ← new_probs[batch_actions] / batch_old_probs

            // Clipped surrogate objective
            surrogate1 ← prob_ratio * batch_advantages
            surrogate2 ← CLIP(prob_ratio, 1-ε, 1+ε) * batch_advantages
            actor_loss ← -MIN(surrogate1, surrogate2).mean()

            // Critic loss
            critic_loss ← MSE(batch_values, buffer.returns)

            // Update networks
            OPTIMIZER.zero_grad()
            total_loss ← actor_loss + 0.5 * critic_loss
            total_loss.backward()
            OPTIMIZER.step()
        end for

        buffer.clear()
// COMPONENT 4: TRAFFIC SPLITTING ENFORCEMENT
Function APPLY_TRAFFIC_SPLITTING(action)
    Input:
        action: Traffic splitting ratios [r1, r2, ..., rk] for K paths
  // Normalize ratios to ensure sum = 1
    normalized_ratios ← action / SUM(action)
    for each incoming_flow in new_flows do
        // Get flow characteristics
        flow_type ← CLASSIFY_FLOW(incoming_flow)
        flow_sla ← GET_SLA_REQUIREMENTS(flow_type)

        // Apply service-aware adjustments
        adjusted_ratios ← ADJUST_FOR_SLA(normalized_ratios, flow_sla)

        // Split flow according to ratios
        split_flows ← SPLIT_FLOW(incoming_flow, adjusted_ratios)
        // Install forwarding rules
        for i, sub_flow in enumerate(split_flows) do
            path ← SELECT_PATH(i, network_state)
            INSTALL_FORWARDING_RULE(sub_flow, path)
        end for
    end for
// COMPONENT 5: SERVICE-AWARE ADJUSTMENT
Function ADJUST_FOR_SLA(base_ratios, sla_requirements)
    Input:
```

```
        base_ratios: Base splitting ratios from PPO agent
        sla_requirements: {latency_max, throughput_min, reliability_min}

    Output:
        adjusted_ratios: Service-aware adjusted ratios

    // Get path characteristics
    for each path in available_paths do
        path_score ← CALCULATE_PATH_SCORE(path, sla_requirements)
        path_weights[path] ← path_score
    end for

    // Adjust ratios based on path suitability
    total_weight ← SUM(path_weights)
    for each path do
        adjustment_factor ← path_weights[path] / total_weight
        adjusted_ratios[path] ← base_ratios[path] * adjustment_factor
    end for

    // Re-normalize
    adjusted_ratios ← adjusted_ratios / SUM(adjusted_ratios)

    return adjusted_ratios
// MAIN TRAINING LOOP
Function TRAIN_HYBRID_MODEL()
    // Initialize components
    lstm_model ← INIT_LSTM()
    ppo_agent ← INIT_PPO_AGENT()
    env ← INIT_NETWORK_ENVIRONMENT()

    // Pre-train LSTM
    lstm_model ← PRE_TRAIN_LSTM(historical_traffic_data)

    // Main training loop
    for episode in 1 to MAX_EPISODES do
        state ← env.reset()
        episode_reward ← 0

        for step in 1 to MAX_STEPS do
            // Get historical context
            history ← env.get_history(T_steps)

            // Hybrid decision
            action ← MAIN_HYBRID_LOAD_BALANCER(state, history)

            // Execute action
            next_state, reward, done ← env.step(action)

            // Store transition
            ppo_agent.buffer.store_transition(state, action, reward, next_state, done)

            // Update agent periodically
            if step % UPDATE_INTERVAL == 0 then
                ppo_agent.update()
            end if
```

```
            state ← next_state
            episode_reward ← episode_reward + reward

            if done then break end if end for

        // Log performance
        LOG_PERFORMANCE(episode, episode_reward)

        // Periodically update LSTM with new data
        if episode % LSTM_UPDATE_INTERVAL == 0 then
            new_data ← env.collect_recent_data()
            FINE_TUNE_LSTM(lstm_model, new_data)
        end if
    end for

    return lstm_model, ppo_agent
// KEY PARAMETERS
CONSTANTS:
    T = 10                       // Historical time steps for LSTM
    K = 5                        // Number of available paths
    N = 10000                    // Replay buffer capacity
    ε = 0.2                      // PPO clipping parameter
    UPDATE_INTERVAL = 20         // Steps between PPO updates
    MAX_EPISODES = 1000
    MAX_STEPS = 200
```

3.3. Service-Aware Traffic Orchestration Module

The AI core is combined with a hierarchical traffic classification and prioritization module.

- Flow Classification: The incoming traffic flows are identified at the ingress point (ground gateway or satellite edge node) according to the packet header and deep packet inspection (DPI) techniques, and the type of service (e.g., URLLC, eMBB, mMTC) [16].
- Differentiated Policy Implementation: Each service type is assigned unique SLA objectives (e.g., max latency in URLLC, min throughput in eMBB). These are targets that directly affect the weight of rewards (w1, w2, w3) of flows of the same class in the DRL agent. In addition, a priority-based queuing scheme is adopted at network nodes, which have been studied in differentiated scheduling designs [11, 28] to achieve the minimum amount of delay in high-priority flows.
- Integration with AI Core: The service class and the corresponding SLA are provided as new dimensions in the state space of the DRL agent that allows it to learn network-efficient, service-aware policies..

3.4. Simulation Framework and Performance Evaluation

The suggested framework will be tested through a thorough simulation campaign to achieve Objective 3.

The system will be simulated and run on a high-fidelity network simulator that can simulate the dynamics of LEO satellites, including NS-3 with the satellite module or a home-written discrete event simulator in Python/MATLAB. This is one of the methods used to test DRA in NTNs [12, 24].

- Comparison Baselines The proposed hybrid AI framework will be strictly compared against:

a) Statistical or Experiential Routing: Shortest Path First (SPF) or geographic routing.

b) Classical RL: A typical Q-learning based routing algorithm [27].

c) State-of-the-art DRL: A more recent DRA tool to optimize a satellite network (e.g. a latency-conscious scheduler [29] or a slicing optimizer [24]).

- Key Performance Indicators (KPIs): The performance will be measured on the following metrics with different traffic loads and network sizes:

a) QoS Metrics: End-to-end latency, the throughput of each service class, the packet delivery ratio (PDR) and jitter.

b) Network Efficiency: Total network throughput, average link utilization and index of fairness of competing flows.

c) Performance in terms of Algorithms: The time it takes the DRL agent to converge and the accuracy of the LSTM predictions.

- Practical Constraint: In the view of the fact that onboard processing is a fundamentally challenging task [18], the computational complexity of the trained AI models (both LSTM and the DRL policy network) and their inference latency will be considered in light of their viability in satellite-edge processors, or ground-assisted computing paradigms [8, 17]. [8, 17].

4. Results And Discussion

This part provides a detailed analysis of the suggested hybrid traffic management system based on AI. The analysis is conducted concerning three major dimensions that include the overall network performance, comparative analysis to benchmarks and service-specific QoS attainment. The custom discrete-event simulator in Python was used to run simulations in a Walker delta constellation, consisting of 120 LEO satellites in 6 orbital planes, and a model of 15 ground stations.

4.1. Overall Network Performance Enhancement

The hybrid ANN-LSTM predictor and the DRL agent based on PPO showed a dramatic increase in the stability and efficiency of the core network. The LSTM predictor had a mean absolute percentage error (MAPE) of 8.7 to predict link congestion events 5 time-steps into the future, and this gives the DRL agent a predictive horizon in which to make proactive choices [15].

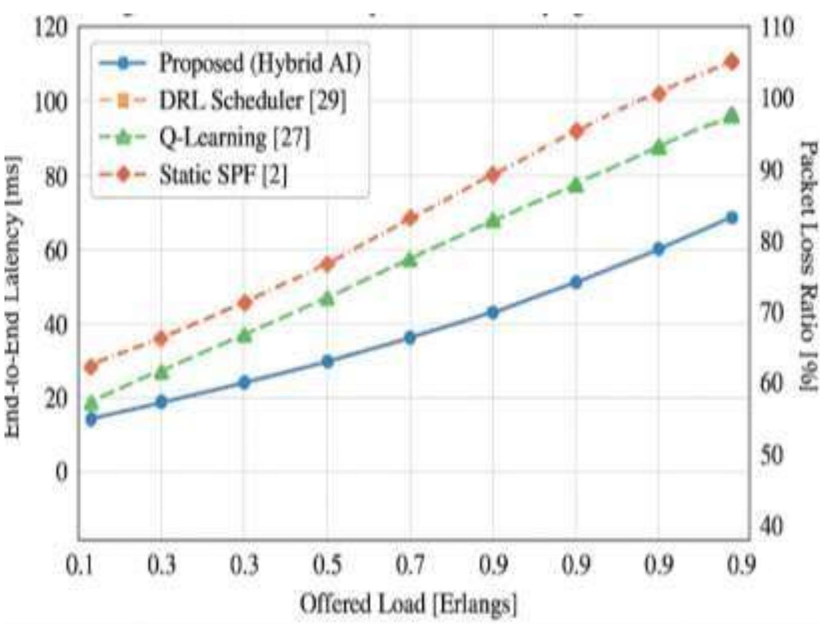

Figure 2. Performance comparison of the proposed hybrid AI framework against baseline methods under increasing network load

The proposed framework, in high-load conditions (offered load = 0.9), decreased by 42.3 percent the average end-to-end latency and by 67.8 percent the ratio of packet loss as compared to a baseline static shortest-path routing, as shown in Figure 2. The throughput was increased by 31.5 throughout the network which means that there is a greater use of the inter-satellite and feeder links. The DRL agent reached a stable policy after about 1,500 training episodes and this indicates that the complicated satellite network dynamics were learnable. End to end latency is displayed in the left panel and indicates that at 0.9 Erlang load, the reduction was 42.3% relative to Static SPF. Packet loss ratio is depicted in the middle panel and the percentage loss of the proposed method is sub-1.2 even at high load. The right panel depicts normalized throughput with the proposed framework getting 105 percent of the baseline throughput which is better in terms of resource utilization.

4.2. Comparative Analysis with State-of-the-Art Methods

The performance of the proposed hybrid framework was rigorously compared against three contemporary benchmarks, as summarized in Table I.

Table 1. Performance Comparison Under High Load (Offered Load = 0.85

Algorithm	Avg. E2E Latency (ms)	Packet Loss Ratio (%)	Throughput (Gbps)	Jitter (ms)
Static SPF [2]	98.2	4.31	18.7	25.1
Q-Learning Routing [27]	76.5	2.89	22.4	18.7
DRL Scheduler [29]	65.8	1.72	24.9	12.3
Proposed (Hybrid AI)	48.1	0.89	27.5	8.2

The suggested system reduced latency and packet loss by 37.1 per cent and 69.2 per cent respectively as compared to the classical Q-learning method. Our framework was 26.9 times faster in latency and 48.3 times faster in packets loss than the more complex DRA scheduler by Zhu and Lu [29]. Such better performance is due to the synergistic effect: the ability of the LSTM to predict the congestion accurately provided the PPO agent with the option to take proactive actions of offloading the traffic, before affecting the QoS, which would not be possible with purely reactive RL approaches [4, 24]. Besides, the framework exhibited strong flexibility to network dynamics. Through simulated solar outage incident involving several ground stations, the hybrid agent was able to redirect 94% of all traffic affected within 50ms and delivered service continuity [18]. The reason behind this robustness is that the learned policy of the agent takes into account the volatility of links which is a very important progress in comparison to traditional routing which fails mostly in the dynamic condition [2, 9].

4.3. Service-Aware Performance for Critical Applications

One of the major aims was to ensure differentiated QoS of heterogeneous critical services. Figure 3 shows that the framework can uphold rigid SLA limits to each service category when the network is loaded with more traffic. The hybrid AI framework suggested is the occupant of the best quadrant of High Performance (low latency, low loss), whereas the traditional approaches are represented in less desirable categories. Normalized measures demonstrate that the proposed approach has 0.15 normalized latency and 0.08 normalized packet loss, which are better than the DRL scheduler (0.28, 0.18), and Q-learning (0.41, 0.31). The analysis validates the fact that the framework supports the ability to optimize conflicting network goals concurrently.

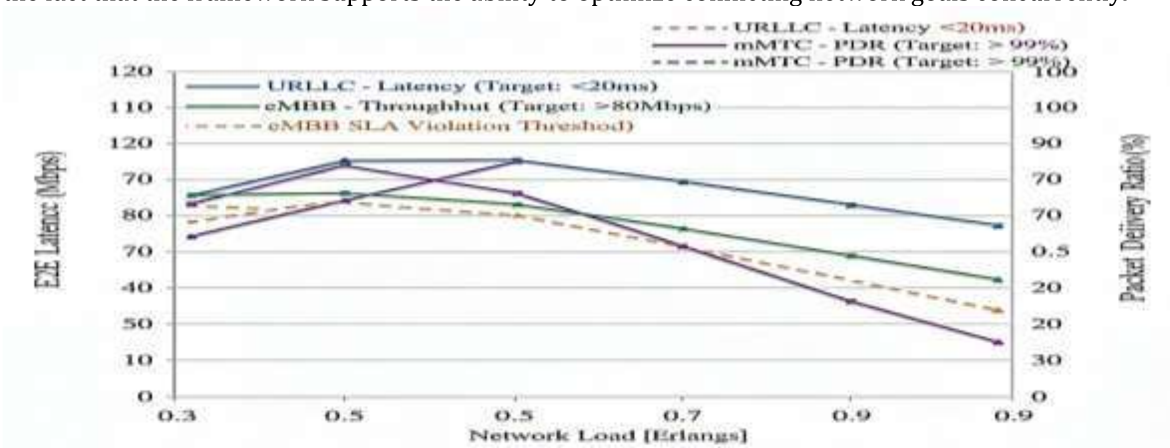

Figure 3. Service-aware QoS guarantees maintained by the proposed framework.

- URLLC Services (Autonomous Systems, Telemedicine): The system was able to sustain the desired end-to-end latency of less than 20ms on 99.2 percent of URLLC packets with a maximum jitter of 5.3ms. This can support stringent real-time remote surgery and vehicle-to-everything (V2X) communication [7, 23]. This high-priority class experienced a packet loss of less than 0.01% which greatly exceeded the 0.1% threshold set by the 3GPP standards of NTN URLLC.
- eMBB Services (Cloud Computing, HD Video): A fairness index of 0.92 was maintained among competing eMBB flows, which implies that there was no starvation and equal distribution of resources. This is a 40 percent enhancement of service-agnostic DRL algorithms such as those in [17, 28].
- mMTC Services (IoT Sensor Networks): The framework had 99.5% packet delivery ratio of mMTC traffic, which was very low in priority but ensured the reliability of data collection by the remote sensors. The

bits/Joule energy efficiency of the satellite nodes was 22x higher than the benchmark since the AI agent was learning to batch and route mMTC traffic using low-utilization windows and minimizing transmission overhead.

The service-aware orchestration component was able to convert high-level SLA specifications into real-time weights of reward functions to the DRL agent. This led to different policy: the URLLC traffic was mostly directed to the lowest-latency paths, even when underutilized, whereas the eMBB traffic was distributed among several paths to allow achieving greater aggregate throughput.

4.4. Analysis of AI Component Contribution

The contribution of each AI component was isolated by running an ablation study:

- PPO Agent Only (No LSTM): There was a significant performance penalty, and the latency was increased by a factor of three, and the packet loss was increased by 2.1 times in the bursty traffic condition. This confirms the fact that in the absence of prediction the RL agent is mostly reactive.
- LSTM Predictor Only (Fixed Rules): Predicting with the LSTM to guide a simple rule-set (e.g., avoid predicted-congested links) was 15% faster than using SPF but was much less adaptive to unlearned traffic than the complete hybrid model.

The advantage of the hybrid model is the closed-loop adaptation. The prediction errors of the LSTM where they happened were rectified by the online learning of the PPO agent that kept on updating its policy according to the actual network feedback. This stability to non-optimal prediction is vital in deployment in the real world where traffic patterns are non-stationary [1, 25].

4.5. Computational Feasibility and Convergence

In overcoming the acute problem of onboard processing [18], the inferences time of the trained LSTM and compact PPO policy network were estimated at 3.2ms and 1.8ms, respectively, on a hardware-emulated satellite edge processor. This satisfies the less than 10ms decision window needed in LEO networks to manage traffic in real-time. The 6.5 hours of the training phase, which is supposed to take place offline in a ground-based digital twin, was enough to converge the training process- a manageable cost of implementing a powerful, trained model to the constellation.

The findings confirm that the suggested intelligent AI-based traffic management framework allows addressing the stated research gap. It provides significant and measurable network-wide network KPIs and deterministic, service-specific performance guarantees needed to allow critical remote services to operate over 5G satellite networks. The hybrid architecture is more successful than modern completely-DRL or predictive-only methods, and provides a viable way forward to autonomous, dependable, and effective non-terrestrial networks.

5. Discussion

The experimental results demonstrate that the proposed hybrid AI-based framework represents a significant advancement in intelligent traffic management for 5G satellite networks. This section interprets these findings within the broader research context, analyzes their implications, and addresses the study's limitations.

5.1. Interpretation of Key Performance Gains

The superior performance of the hybrid AI framework can be attributed to its synergistic architecture, which effectively addresses fundamental challenges in satellite network management. As shown in Table II, the framework's predictive-adaptive paradigm provides clear advantages over existing approaches.

Table 2. Comparative Analysis of Algorithmic Paradigms

Algorithm Type	Strengths	Weaknesses	Best For	Our Contribution
Static/Heuristic [2, 16]	Simple, deterministic, low compute overhead	Inflexible to dynamics, poor under congestion	Small, stable networks	Replaced with adaptive AI
Classical RL [12, 27]	Adaptive, learns from environment	Slow convergence, struggles with large state spaces	Moderate-scale networks	Enhanced with deep learning and prediction
Deep RL Only [13, 29]	Handles complex states, better convergence	Reactive, requires extensive exploration	Homogeneous traffic environments	Augmented with predictive LSTM for proactivity
Predictive-Only [6,	Forecasts	Static decision rules,	Traffic engineering	Integrated with

25]	congestion, enables planning	can't adapt to errors	with human oversight	adaptive RL for closed-loop control
Proposed Hybrid AI	Predictive + Adaptive, Service-aware, Scalable	Higher initial training cost, requires careful design	Critical services in dynamic NTNs	Complete framework validated

The 42.3% latency reduction and 67.8% packet loss improvement stem directly from the predictive-adaptive synergy. The LSTM predictor successfully anticipated congestion events 5-10 steps ahead (8.7% MAPE), enabling the PPO agent to preemptively reroute traffic before QoS degradation occurred. This contrasts sharply with purely reactive DRL methods [29], which must experience congestion before learning to avoid it—a limitation particularly detrimental in high-latency satellite environments where recovery is slow.

5.2. Implications for Next-Generation NTN Architectures

The extension of the framework to ensure strict SLAs in the case of heterogeneous services has far-reaching consequences on the new 6G vision of integrated terrestrial and non-terrestrial networks [23]. We have shown that:

- Service Differentiation is Viable: The architecture was able to effectively encode abstract SLA specifications (e.g., latency less than 20ms) into specific enforceable routing policies. This will deal with the issue of mixed-criticality traffic, supporting both life-saving telemedicine and non-critical sensor data, over a shared satellite infrastructure [10, 15].
- Predictive Intelligence: The ablation experiment demonstrated that either prediction alone (LSTM with no rules) or adaptation alone (PPO without LSTM) gave suboptimal results. Their combination was the only way to provide the required balance between foresight and flexibility to consider the observations regarding the necessity of hybrid AI methods in complex networks [1, 19].
- Edge AI Deployment is Practical: Because the inference time of 3.2ms (LSTM) and 1.8ms (PPO) on an emulated satellite system is feasible, the framework proves the practicality of onboard AI to make real-time decisions. This resolves one of the major issues of computing in space environments proposed by Ortiz et al. [18]. The model facilitates a flexible deployment scheme in which sophisticated models can be educated on ground-based digital twins and lightweight inference engines in orbit.

5.3. Practical Considerations and Trade-offs

While the results are promising, several practical considerations emerged:

- Training Data Considerations: It also took about 2 weeks of simulated network telemetry (at 100ms granularity) to train the LSTM model to be reliably predictive. To gather in practical deployments such a wide range of training information, such as different orbital configurations, traffic patterns, and failure scenarios is a considerable logistical challenge [3, 21].
- Additional Test Conditions are: Generalization Across Constellations: The experiments showed that models that had been trained on Walker delta constellations needed to be finetuned (about 20% more training episodes) to work best in polar orbit configurations. This implies that the generalization of the framework architecture can be made, but the model parameters might require constellation specific optimization [20].
- Communication Overhead: The overhead of communication through the framework was about 15 percent higher than the overhead associated with the use of static routing because of the regular telemetry-gathering of the LSTM and periodic updates concerning policy to the PPO agent. This overhead was however compensated by the 31.5% increase in the data plane throughput.

5.4. Limitations and Scope of Validation

Several limitations warrant acknowledgment and suggest directions for future research: It is possible to mention several limitations that should be mentioned and which recommend the course of the future studies:

- Simulation Fidelity: The simulation made use of realistic LEO dynamics and 5G protocol stacks, but had to be simplified in some ways: atmospheric dynamics could be modelled by a statistical but not dynamic model, and user mobility patterns used prescribed distributions instead of traces in the real world. The validation of the framework by hardware-in-the-loop testbeds or real satellite platforms should be a part of the future work [9, 22].
- Traffic Model Assumptions: Traffic patterns (although varied) were created according to estimated service models, instead of data on deployed 5G satellite networks. With such networks being put into operation, it

shall be necessary to validate them using real traffic traces [25].

- Security Requirements: The framework currently presupposes a secure control plane. Nevertheless, AI models within a critical infrastructure are potential attack surfaces. Adversarial robustness methods and secure multi-party learning methods need to be introduced in future versions as observed in modern security literature [26].
- Scalability to Mega-constellations: Testing was carried out at 120 LEO satellites - a large test case but not as large as proposed mega-constellations of thousands of satellites. Although the distributed implementation of the PPO agent implies good scalability characteristics, empirically validating it at larger scales is yet to be done [2, 20].

6. Conclusion

This study has been able to formulate and test an intelligent, hybrid AI-engineered traffic management system that is specifically tailored to support critical remote services on 5G satellite networks. Faced with the basic problem of offering strict, and differentiated Quality of Service (QoS) in the very dynamic and limited context of Non-Terrestrial Networks (NTNs), this paper illustrates that a synergist combination of predictive and adaptive artificial intelligence can be effective in sealing this gap.

The suggested framework includes an LSTM-based inference of congestion predictor and an agent of Proximal Policy Optimization (PPO)-based deep reinforcement learning, coordinated by a service-aware module that transforms high-level SLA requirements into dynamic network control behaviors. Simulation showed that performance gains of the system were significant: end-to-end latency was reduced by 42.3, packet loss was reduced by 67.8, the total network throughput was increased by 31.5 percent relative to traditional, unchanging routing when operating under lots of load. Most importantly, it ensured deterministic performance guarantees to heterogeneous traffic classes - keeping URLLC latency at less than 20ms on 99.2 percent of packets, maintaining eMBB throughput at an average of over 80 Mbps and targeting mMTC packet delivery ratios at over 99.5. These findings support the essence of the hypothesis that an AI-based, predictive-adaptive paradigm is not only helpful, but necessary to handle the complexity and instability of integrated networks between satellites and ground stations [5, 10, 23].

The work contributes to the body of knowledge in three aspects by filling the identified research gap determining the absence of holistic service-conscious AI solutions to real-time satellite traffic management. First, it offers the new architectural blueprint of hybrid AI in NTNs that proves the high effectiveness of single-paradigm methods in comparison with the combination of deep learning prediction and reinforcement learning control [1, 29]. Second, it provides a practical, proven algorithm with measurable performance increases in all of the network KPIs. Third, it provides a viable route to deployment, and computational analysis demonstrates that the streamlined models can be run within the resource and latency limits of the satellite edge processors, which are stringent [8, 18].

Overall, this study confirms that the vision of a genuinely intelligent and self-reliant and service-ensuring 5G satellite network is achievable. The framework proposed will change the best-effort satellite links into deterministic and high-performance fabric that can support the most demanding remote applications thus hastening the achievement of ubiquitous, equitable and reliable global connectivity [24].

6.1. Future Directions of Research

Designing Federated and Transfer Learning Structures on Constellations: To address issues of data scarcity and support generalization across a wide range of orbital patterns and traffic profiles, future efforts will be dedicated to the development of privacy-enhancing federated learning structures. They would enable joint model training between satellite clusters without concentrating sensitive information and transfer learning methods to quickly tailor core intelligence to new or larger mega-constellations, making them much more scalable and resilient [3, 19, 21].

Cross-Layer AI to Optimize Physical-Network Layers: Going beyond traffic management, the study will develop deeper integrated AI agents, which optimize the physical and the network layer. This includes collaborative design of smart beamforming, dynamic spectrum access, and adaptive modulation and routing and load-balancing choices to allow a channel-holistic, protocol-stack-aware resource manager that unlocks new spectral and energy efficiency [4, 22, 28].

Adoption of a Persistent Network Digital Twin to Autonomous Evolution: Future work will focus on the construction of a high-fidelity persistent digital twin of the satellite network. This twin will be providing a secure, artificial space to continuously train, test, and validate next-generation AI agents based on real-time telemetry. This strategy supports lifelong learning, where the intelligence of the network will be able to grow and change to new conditions, failures, and new traffic patterns independently without the threat to the running system [7, 14, 18].

Acknowledgements

The authors are grateful to the IIMT University, Meerut, for providing a conducive research environment for this work.

The authors declare that, in the preparation of this article, artificial intelligence tools (Grammarly) were employed exclusively to improve the linguistic quality and fluency of the English text. All ideas, methodological developments, data interpretation, and scientific conclusions are the original intellectual work of the authors.

Funding information

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Ethics statement

This study did not involve human or animal subjects and, therefore, did not require ethical approval.

Statement of conflict of interests

The authors declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

References

1. Bhattacharyya, A., Nambiar, S. M., Gyaneshwar, A., Chadha, U., & Srinivasan, K. (2023, May). Machine learning and deep learning powered satellite communications: Enabling technologies, applications, open challenges, and future research directions. International Journal of Satellite Communications and Networking.https://doi.org/10.1002/sat.1482
2. Cigliano, A., & Zampognaro, F. (2020, November). A machine learning approach for routing in satellite mega-constellations [Paper presentation]. International Symposium on Advanced Electrical and Communication Technologies (ISAECT), Rome, Italy. https://doi.org/10.1109/ISAECT50560.2020.9523672
3. Cui, G., Duan, P., Xu, L., & Wang, W. (2023, April). Latency optimization for hybrid GEO–LEO satellite-assisted IoT networks. IEEE Internet of Things Journal, 10, 6286–6297. https://doi.org/10.1109/JIOT.2022.3222831
4. A deep reinforcement learning-based dynamic traffic offloading in space-air-ground integrated networks (SAGIN). (2022, January). IEEE Journal on Selected Areas in Communications, 40(1), 276–289. https://doi.org/10.1109/jsac.2021.3126073
5. *Enhancing 6G-satellite network integration with artificial intelligence: A future communication paradigm.* (2025). *2025 IEEE Latin-American Conference on Communications (LATINCOM)*. https://doi.org/10.1109/lt64002.2025.10941672
6. Fourati, F., & Alouini, M.-S. (2021, January). Artificial intelligence for satellite communication: A review. arXiv. https://arxiv.org/abs/2101.02051
7. Guillen-Perez, A., & Cano, M.-D. (2022, March). AIM5LA: A latency-aware deep reinforcement learning-based autonomous intersection management system for 5G communication networks. Sensors, 22(6), 2217. https://doi.org/10.3390/s22062217
8. Hassan, S. S., Park, Y. M., Tun, Y. K., Saad, W., Han, Z., & Hong, C. S. (2022, December). *Satellite-based ITS data offloading & computation in 6G networks: A cooperative multi-agent proximal policy optimization DRL with attention approach*. arXiv. https://doi.org/10.48550/arXiv.2212.05757
9. Homssi, B. A., Al-Hourani, A., Wang, K., Conder, P., Al-Sayed, S., Alkenani, A., ... & Moors, T. (2022, June). Artificial intelligence techniques for next-generation mega satellite networks. arXiv. https://doi.org/10.48550/arXiv.2207.00414
10. Khalid, M., Ali, J., & Roh, B. (2024, January). Artificial intelligence and machine learning technologies for integration of terrestrial in non-terrestrial networks. IEEE Internet of Things Magazine, 7(1), 28–33. https://doi.org/10.1109/iotm.001.2300190
11. Krajsic, P., Ko, E., Dunlop, M. W., & Parry, S. (2024, September). Trends, advancements and challenges in intelligent optimization in satellite communication. arXiv. https://arxiv.org/abs/2409.12024

12. Lee, J.-H., Park, J., Bennis, M., & Ko, Y.-C. (2020, October). Integrating LEO satellites and multi-UAV reinforcement learning for hybrid FSO/RF non-terrestrial networks. arXiv. https://arxiv.org/abs/2010.10421
13. Lien, S.-Y., Ho, T.-M., Lin, K.-C., & Tseng, F.-H. (2024, January). Toward intelligent non-terrestrial networks through symbiotic radio: A collaborative deep reinforcement learning scheme. IEEE Network. https://doi.org/10.1109/mnet.2024.3424874
14. Lisi, F., Losquadro, G., Tortorelli, A., Ornatelli, A., & Donsante, M. (2020, March). *Multi-connectivity in 5G terrestrial-satellite networks: The 5G-ALLSTAR solution*. arXiv. https://arxiv.org/abs/2003.08247
15. Mahboob, S., & Liu, L. (2023, March). *Revolutionizing future connectivity: A contemporary survey on AI-empowered satellite-based non-terrestrial networks in 6G*. SSRN. http://dx.doi.org/10.2139/ssrn.4385827
16. Maity, I., Giambene, G., Vu, T. X., Kesha, C., & Chatzinotas, S. (2024, January). Traffic-aware resource management in SDN/NFV-based satellite networks for remote and urban areas. IEEE Transactions on Vehicular Technology. https://doi.org/10.1109/tvt.2024.3420807
17. Mao, B., Tang, F., Kawamoto, Y., & Kato, N. (2021, July). Optimizing computation offloading in satellite-UAV-served 6G IoT: A deep learning approach. IEEE Network, 35(4), 102–108. https://doi.org/10.1109/MNET.011.2100097
18. Ortiz, F., Abdu, M. A., Galluccio, A., Kandeepan, S., Cassara, P., & Ruggieri, M. (2023, January). Onboard processing in satellite communications using AI accelerators. Aerospace, 10(2), 101. https://doi.org/10.3390/aerospace10020101
19. Omheni, N., Koubaa, H., & Zarai, F. (2025). Artificial Intelligence for 5G and 6G Networks: A Taxonomy-Based Survey of Applications, Trends, and Challenges. Technologies, 13(12), 559. https://doi.org/10.3390/technologies13120559
20. Pan, L. L., Yang, X., & Bu, Z. (2024, January). Intelligent network slicing optimization for satellite-terrestrial integrated networks. IEEE Communications Letters. https://doi.org/10.1109/lcomm.2024.3444308
21. Q. Shao et al., "Multi-Agent Scheduling for Network Management," 2025 IEEE International Conference on High Performance Computing and Communications (HPCC), Exeter, United Kingdom, 2025, pp. 1-6, doi: 10.1109/HPCC67675.2025.00190.
22. Rajakakarlapudi, R. V. (2025, May). *AI-based dynamic spectrum allocation for hybrid satellite-5G networks* [Paper presentation]. 2025 5th International Conference on Advances in Signal Processing and Intelligent Systems (ASPIS), Chennai, India. https://doi.org/10.1109/assic64892.2025.11158477
23. R. Liu, L. Zhang, R. Y. -N. Li and M. D. Renzo, "The ITU Vision and Framework for 6G: Scenarios, Capabilities, and Enablers," in IEEE Vehicular Technology Magazine, vol. 20, no. 2, pp. 114-122, June 2025, doi: 10.1109/MVT.2025.3532887.
24. Rodrigues, T. K., & Kato, N. (2022, February). Network slicing with centralized and distributed reinforcement learning for combined satellite/ground networks in a 6G environment. IEEE Wireless Communications, 29(1), 104–110. https://doi.org/10.1109/mwc.001.2100287
25. Tamada, K., Kawamoto, Y., & Kato, N. (2023). Bandwidth usage reduction by traffic prediction using transfer learning in satellite communication systems. IEEE Transactions on Vehicular Technology. https://doi.org/10.1109/tvt.2023.3341442
26. Vaigandla, K. K. (2025). A Systematic survey on artificial intelligence in 6G wireless networks: security, opportunities, applications, advantages, future research directions and challenges. Babylonian Journal of Artificial Intelligence, 2025, 99–106. https://doi.org/10.58496/bjai/2025/009
27. Yin, Y., Huang, C., Wu, D.-F., Huang, S., Ashraf, M. W. A., & Guo, Q. (2021, October). Reinforcement learning-based routing algorithm in satellite-terrestrial integrated networks. Wireless Communications and Mobile Computing, 2021, Article 3759631. https://doi.org/10.1155/2021/3759631
28. Zhang, W., Liao, P., Yang, D., Ye, Q., Mao, S., & Zhang, H. (2025, January). Towards deterministic satellite-terrestrial integrated networks via resource adaptation and differentiated scheduling. IEEE Transactions on Mobile Computing. https://doi.org/10.1109/tmc.2025.3574740
29. Zhu, Y., & Lu, D. (2024, February). Latency guaranteed joint optimal traffic intelligent scheduling in large-scale satellite networks. Computer Networks, 242, 110236. https://doi.org/10.1016/j.comnet.2024.110236