A Modified Lyapunov-Based MEC Offloading Algorithm for Severity-Aware QoS in 5G/B5G IoT Systems

Modified Lyapunov-Based Severity-Aware QoS in 5G/B5G IoT Systems

Sahana S Reddy*, R. Sukumar
Department of Electronics and Communication Engineering, JAIN (Deemed to be University), Bengaluru, 560 034, India

***Abstract*—The rapid growth of latency-sensitive and computation-intensive IoT applications in 5G and Beyond-5G (B5G) networks has increased the demand for efficient Multi-access Edge Computing (MEC) offloading strategies. Current MEC frameworks have several limitations: 1) binary QoS modeling without considering deadline violation severity, 2) a lack of severity-aware optimization in IoT applications, 3) insufficient consideration of different task criticality, and 4) poor handling of dynamic latency and energy trade-off in large-scale IoT environments. This study proposes a modified Lyapunov-based severity-aware MEC offloading framework for heterogeneous 5G/B5G IoT systems. The proposed framework utilizes task deadlines, task criticality, queue states, wireless channel conditions, and MEC resource availability as input for adaptive offloading optimization. A QoS Violation Severity Index is introduced to jointly capture deadline violation magnitude and task criticality. Furthermore, severity-aware virtual queues are integrated with a modified Lyapunov Drift-Plus-Penalty optimization framework to dynamically minimize QoS violation severity while balancing latency and energy consumption. Experimental evaluation demonstrates that the proposed framework significantly reduces average task delay to 82 ms, energy consumption to 6.0 mJ, and QoS violation rate to 4.8%, while improving long-term system stability compared with existing MEC offloading approaches in dynamic 5G/B5G IoT environments.**

***Keywords*—5G and Beyond-5G; multi-access edge computing; quality of service; task offloading; severity estimation; lyapunov optimization**

I. Introduction

The evolution of 5G and Beyond-5G (B5G) networks have greatly improved the capabilities of the Internet of Things (IoT) by facilitating ultra-reliable communication, massive device connectivity and low-latency services [1]. These networks enable real-time applications like smart healthcare, autonomous transportation, industrial automation, intelligent surveillance, and smart city infrastructures, which produce and process huge amounts of heterogeneous data [2, 3]. To facilitate such latency-sensitive applications, Multi-access Edge Computing (MEC) has been embedded into 5G/B5G architectures to introduce computation and storage to the end-users and thereby shorten the communication latency and improve service responsiveness [4, 5]. Although the 5G/B5G IoT technologies have made great strides, modern network has still many challenges in operation and resource management [6]. The massive growth of connected devices introduces network congestion, complexity of resource allocation, and changes in traffic patterns, making the efficient management of tasks highly challenging [7, 8]. Moreover, IoT devices also have limited battery power, computation capability, and storage capacity, which increases the reliance on edge servers [9, 10]. Furthermore, heterogeneous IoT applications show a wide variety of Quality of Service (QoS) demands related to latency, reliability, bandwidth, and energy efficiency, making uniform resource management ineffective [11, 12]. Therefore, achieving reliable QoS under various latency, energy consumption, and resource utilization constraints remains a significant challenge in the modern 5G and B5G IoT environments [13].

Several existing studies analyze MEC-based task offloading and resource optimization techniques to improve QoS performance in 5G and B5G IoT networks [14, 15]. For instance, a joint optimization framework for energy consumption and time delay in dependency-aware task offloading for 5G applications was developed [16]. It used a Directed Acyclic Graph (DAG) to analyze the dependencies between tasks and an improved Particle Swarm Optimization (PSO) algorithm to optimize task offloading decision and MEC server selection optimization. To efficiently manage the task, a 5G-MEC task offloading framework was developed, which minimizes dropped task ratio, computational latency, and communication latency through Mixed Integer Linear Programming (MILP), PSO, and Genetic Algorithm (GA) [17]. Despite the progress, several key research gaps remain. 1) Existing works usually model QoS violations in a binary way without considering the severity of deadline misses, making it ineffective in handling latency-sensitive IoT applications [18,19]; 2) Most existing schemes focus on optimizing average latency or energy consumption rather than explicitly minimizing severe QoS violations, which expose critical tasks to risks [20]; 3) Current MEC offloading frameworks consider limited heterogeneous task criticality, leading to unfair QoS degradation for high-priority applications [21, 22]; 4) Many existing solutions assume relatively homogeneous IoT environments and fail to effectively address the diverse latency, reliability, and energy requirements of large-scale heterogeneous IoT applications in dynamic 5G and B5G networks [23, 24]. To overcome these, the proposed work develops a severity-aware MEC task offloading framework for 5G and Beyond-5G IoT networks that minimizes

*Corresponding author.

severe QoS violations while balancing latency and energy consumption for heterogeneous IoT applications. To achieve this, a modified Lyapunov Drift-Plus-Penalty (LDPP) optimization framework is developed to enable dynamic and real-time offloading decisions by jointly stabilizing severity-aware virtual queues and optimizing system performance under varying network conditions. The key novelty of the proposed framework is as follows:

Severity-Aware QoS Violation Modeling: A novel QoS Violation Severity Index (QVSI) is proposed to quantify the severity of the delay violation by combining the task criticality and the magnitude of delay violation, overcoming the limitations of traditional binary QoS models in 5G/B5G IoT networks.

Modified Lyapunov-Based Optimization Framework: A severity-aware virtual queue model is incorporated into a modified LDPP optimization framework to jointly minimize the severity of QoS violation, LATENCY, and energy consumption while ensuring long-term stability.

Adaptive MEC Offloading Decision Strategy: An intelligent offloading mechanism is developed to dynamically select local execution, MEC offloading, and cooperative MEC execution based on the real-time queue states, channel conditions, and the available MEC resources in the large-scale heterogeneous IoT environment.

Efficient Support for Heterogeneous 5G/B5G IoT Applications: The proposed framework effectively handles heterogeneous IoT applications with different latency requirements and criticality levels, while minimizing the occurrence of severe QoS violations, deadline miss ratio, latency, and energy consumption compared with existing MEC offloading approaches.

The article is structured as follows. A review of the relevant works is presented in Section II. Section III includes a detailed process of the proposed approach for task offloading in MEC-enabled IoT systems. Section IV exhibits the findings and discussion, and Section V concludes with the findings of the study.

II. Related Work

This section reviews the related work on existing MEC-based task offloading and resource allocation methods in 5G/B5G IoT environments. Zhai et al. [25] modeled a task-offloading scheme for hybrid edge computing networks, where the tasks can be executed locally, offloaded to edge servers, or forwarded to neighboring outgoing devices. It used Deep Reinforcement Learning (DRL) with a Double Deep Q Network (DDQN) algorithm to optimize the task offloading decisions, reducing the overall task delay and prioritizing the high-priority tasks. However, the framework mainly focused on delay-aware offloading and priority scheduling, whereas the severity of QoS violations, heterogeneous task criticality, and long-term queue stability were not explicitly modeled in the optimization process. Benbraika et al. [26] introduced the Hungarian Algorithm for Task Offloading (HATO) for effective task scheduling in 5G-enabled Vehicular Edge Computing (VEC) systems. The model effectively distributed computational workloads, reduced delay, and improved task processing performance in autonomous vehicles. However, the approach mainly depended on stable 5G connectivity and central edge server support, which may affect performance in highly dynamic real-time environments.

To balance the trade-off between energy consumption and computation time, Alam et al. [27] introduced task offloading and resource allocation for the Internet of Vehicles (IoV) using PSO. The method improved QoS, reduced latency, and minimized energy consumption during computational task execution. However, the model was estimated under limited network settings, and it could not capture task criticality during offloading optimization, which limits its applicability to delay-sensitive applications. Younis et al. [28] introduced an Energy-Latency-aware Task Offloading and Approximate Computing (ETORS) framework to optimize the trade-off between energy consumption and latency using the Dual-Decomposition Method (DDM). This framework intelligently balances the energy consumption and shortened completion time by processing the workloads without overloading. However, the framework primarily focused on energy-latency trade-off optimization without considering severity-aware QoS degradation and long-term queue stability. Luo et al. [29] introduced a joint task offloading and resource allocation framework for MEC using the Lyapunov optimization with the Potential Minimum Point (PMP) algorithm. The model achieved increased QoS, lowered latency, balanced energy consumption, and better system stability. However, the framework focused mainly on resource optimization and system stability, while the effect of deadline violation and differentiated QoS handling for critical IoT tasks remains unexplored. Yang et al. [30] introduced a Deep Reinforcement Learning-based Task Offloading (DRTO) framework in MEC systems using Deep Neural Networks (DNNs) and Reinforcement Learning (RL) techniques. The method helped reduce task delay, improved computation efficiency, and increased the speed of offloading decisions under changing wireless network conditions. However, the framework mainly focused on delay optimization and could not jointly optimize QoS violation severity, latency, and energy consumption under long-term queue stability constraints in heterogeneous IoT environments.

Mathi et al. [31] introduced a Distributed Collaborative Learning Framework to detect Black Hole Attacks in the Routing Protocol for Low-power and Lossy networks (RPL) based Internet of Things (IoT) networks. The framework reduces communication overhead, energy consumption, and computational complexity. However, the framework is mainly focused on detecting black hole attacks; performance in IoT deployments cannot be discussed. Xiao and Dong [32] introduced a lightweight CNN-based deep learning model for graphic recognition in the Internet of Things (IoT) ecosystem. The model was suitable for resource-limited IoT devices and supports simultaneous graphic recognition applications. However, the impact of varying network conditions and device heterogeneity was not analyzed. Vidyaningtyas et al. [33] introduced a Sequential Power Allocation (SePA) algorithm to optimize power distribution within each sector. The method minimizes intra-sector interference and improves power efficiency, and the framework supports reliable communication in high-density environments such as Internet of Things (IoT)

networks. However, the machine learning-based optimization techniques are not integrated.

Thus, by analyzing extensive research on MEC-based task offloading and resource optimization in 5G and B5G IoT networks. Existing approaches mainly focus on minimizing the average latency, energy consumption, or overall task failure rate without explicitly considering the severity of QoS violations. Most existing approaches consider QoS satisfaction in a binary manner and ignore the extent to which deadlines are missed, which is critical for latency-sensitive and mission-critical IoT applications. Moreover, there is a limited consideration of heterogeneous task criticality and diverse QoS requirements, leading to suboptimal and sometimes unfair resource allocation for high-priority tasks. As a result, current MEC offloading frameworks remain insufficient in effectively handling severe QoS degradation under dynamic and large-scale IoT network conditions. Therefore, the proposed framework develops a severity-aware LDPP-based offloading algorithm that minimizes severe QoS violations while balancing latency and energy consumption.

III. System Overview

The proposed system considers a 5G/B5G-enabled MEC architecture consisting of multiple heterogeneous IoT devices, a 5G/6G base station, and an edge server connected through wireless communication links, as shown in Fig. 1. Each IoT device creates computation-heavy and latency-sensitive tasks that are stored in local task queues. The IoT devices have different battery life, computing power, and QoS needs, making the network highly dynamic and resource-constrained. Depending on various wireless channel conditions, queue status, task deadline, and device energy level, each task may be processed locally on the IoT device or offloaded to the MEC/edge cloud server for faster processing. The wireless communications between the IoT devices and the MEC server are represented through time-varying channel states, which represent real-world network dynamics. To effectively provide delay-sensitive and critical IoT applications in 5G/B5G environments, the proposed framework introduces a severity-aware task management scheme that measures the QoS violation, optimizes offloading decisions adaptively to minimize the latency, energy consumption, and service reliability.

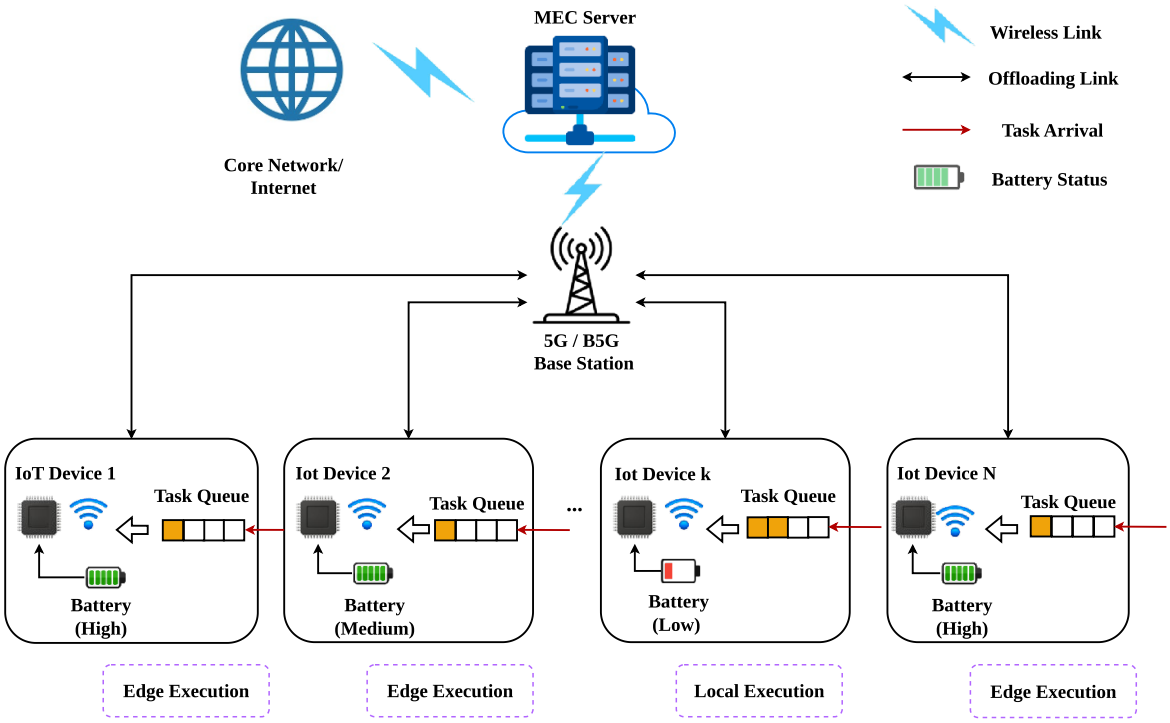

Fig. 1. System model for multi-access edge computing-assisted task offloading in heterogeneous 5G/B5G IoT environments.

A. Problem Formulation

The considered 5G/B5G MEC-enabled IoT system supports multiple heterogeneous IoT devices that generate computation-intensive and latency-sensitive tasks to be executed locally or offloaded to the MEC server. The primary goal is to reduce the QoS violation severity, latency, and energy consumption, while maintaining long-term system stability under stochastic task arrival. In contrast to traditional binary QoS models, a severity-aware optimization framework is incorporated to account for the magnitude of deadline misses and for task importance. The optimal long-term problem is formulated as shown in Eq. (1).

$$min\ lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \mathrm{E}\left[\sum_{i=1}^{N}\left(\alpha S_i(t) + \beta T_i(t) + \gamma E_i(t)\right)\right] \quad (1)$$

where, $\mathrm{E}[\cdot]$ representing long-term average system behavior, $S_i(t)$, $T_i(t)$, and $E_i(t)$ represent the severity of QoS violation, latency and energy consumption, respectively, N is the total number of IoT devices, T is the total simulation, and t is the discrete time slot. The weighting factors α, β, and γ are such that $\alpha + \beta + \gamma = 1$, controlling the balance between severity,

delay, and energy efficiency. System stability is guaranteed by resource constraints as the sum of computation capacity of MECs is bounded by $\sum_{i\epsilon N} f_i(t) \leq F^{max}$ and the sum of communication bandwidths of MECs is bounded by $\sum_{i\epsilon N} b_i(t) \leq B^{max}$. The components $f_i(t)$ and $b_i(t)$ are the allocated computing and bandwidth resources, and F^{max} and B^{max} are the maximum capacities of computing and bandwidth, respectively. To achieve long-term stability, the virtual queue should be bounded as in Eq. (2).

$$lim_{T\to\infty} sup \frac{1}{T} \sum_{t=1}^{T} E[Z_i(t)] < \infty \quad (2)$$

where, $Z_i(t)$ is the severity-aware queue length. Lastly, exactly one execution mode is chosen for each task, in which the decision variables are binary. The problem is formulated as a stochastic optimization problem that minimizes the severity of the long-term QoS violations, the system latency, and the system energy consumption under the system constraints.

B. Proposed Severity-Aware Offloading Framework

The proposed severity-aware MEC offloading framework sequentially and iteratively operates to efficiently manage computation tasks in 5G/B5G IoT environments, as depicted in Fig. 2. Initially, the IoT devices generate heterogeneous tasks, characterized by data size, deadline constraints, and levels of criticality. The system estimates the local execution and MEC offloading delay, including transmission and computation for each task. Based on these delays, the QVSI is calculated to reflect the degree of possible violation of the deadlines while incorporating task importance. The obtained severity values are then used to update severity-aware virtual queues, which capture the accumulated QoS degradation over time. These queue states are then embedded in an LDPP optimization framework to optimize latency, energy use, and severity-aware QoS performance. This optimization dynamically switches between local processing mode, MEC offloading mode, and cooperative MEC execution mode based on the system conditions of each task. The selected decisions are then implemented, and the resulting delay and energy consumption are recorded for performance assessment. Finally, the severity-aware queues are updated based on the observed outcomes, and the process repeats over the successive time slots, which makes real-time adaptive and stable offloading decisions under the dynamic network conditions.

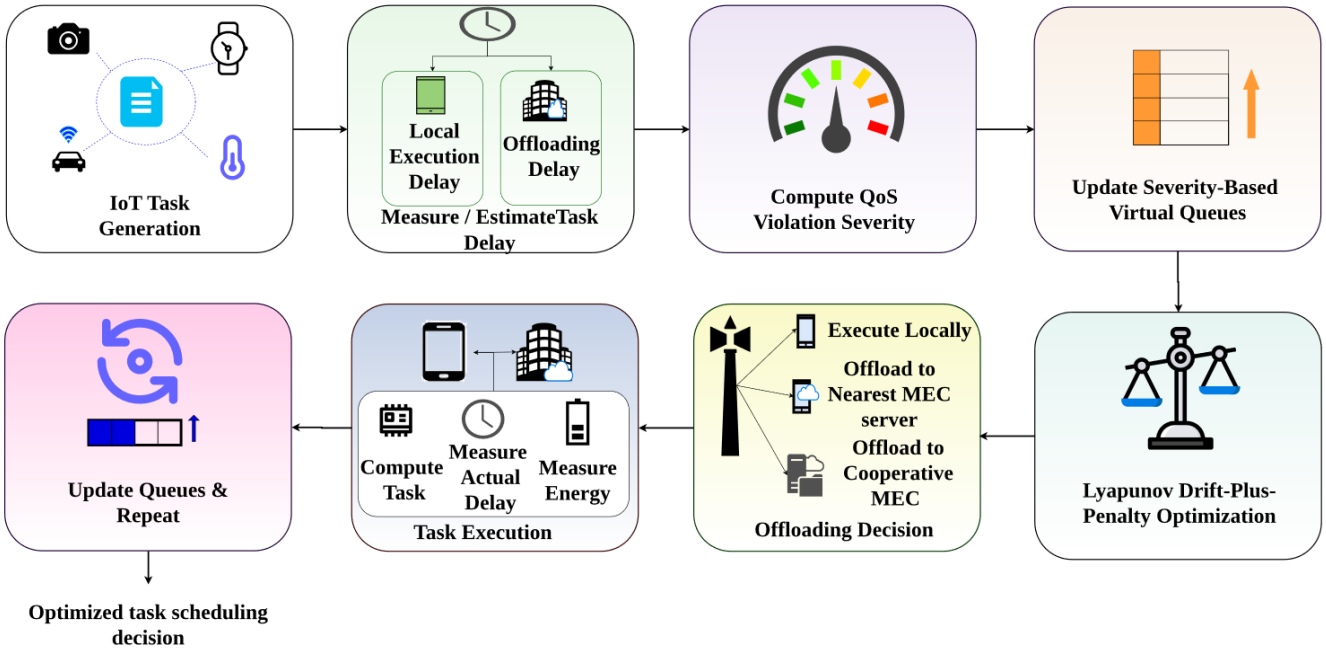

Fig. 2. Architecture of the proposed severity-aware Lyapunov-based MEC offloading system for 5G/B5G IoT networks.

1) System initialization and task modelling: In the considered 5G/B5G MEC enabled IoT environment, a collection of heterogeneous IoT devices $\mathcal{N} = \{1,2,\dots,N\}$ dynamically generates computation tasks over time. Every task generated by the i-th device $\mathcal{T}_i(t)$ is expressed as follows in Eq. (3).

$$\mathcal{T}_i(t) = \{S_i^{data}(t), D_i^{max}, C_i\} \quad (3)$$

where, $S_i^{data}(t)$ is the task size, D_i^{max} is the deadline constraint, and C_i is the task criticality. The task criticality factor C_i represents the priority level of each IoT task and is assigned based on the application type and service requirements. In the proposed framework, C_i is predefined at the time of task generation and remains static throughout the task execution process. Specifically, high-criticality tasks (e.g., URLLC-based applications) are assigned higher weights, medium-criticality tasks are assigned moderate weights, and low-criticality tasks are assigned lower weights. This static assignment reflects the inherent priority differences among heterogeneous IoT services and ensures consistent severity-aware scheduling and offloading decisions within the Lyapunov optimization framework. Each task is initially stored in the local queue of each IoT device upon arrival, and then a decision is taken to either execute or offload the task. The local task dynamics at each device are modeled as indicated in Eq. (4).

$$Q_i(t+1) = max[Q_i(t) - \mu_i(t), 0] + A_i(t) \quad (4)$$

where, $Q_i(t)$ is the queue backlog of the device i, $A_i(t)$ is the number of newly arrived tasks at the time slot t, $\mu_i(t)$ is the number of tasks processed or offloaded at the time slot t, and $Q_i(t+1)$ is the updated queue backlog of IoT devices i at the next time slot $t+1$. This queue formulation represents the stochastic network conditions with dynamic arrival and service processes of IoT tasks. The defined queueing model serves as the foundation for subsequent severity-aware optimization and Lyapunov-based offloading decisions in the proposed framework.

2) Task characterization and delay estimation: For each incoming task i, the system evaluates its execution characteristics under two different modes: local execution and MEC offloading [34]. The local execution delay $D_i^{loc}(t)$ is dependent on the computational power of the IoT device and is represented as given in Eq. (5).

$$D_i^{loc}(t) = \frac{S_i^{data}(t).k}{f_i^{loc}(t)} \quad (5)$$

where, $S_i^{data}(t)$ is the size of the task, k is the number of CPU cycles needed per bit of data, and $f_i^{loc}(t)$ is the local CPU frequency. In the case of MEC offloading, the overall delay consists of the transmission delay and the edge computation delay as indicated in Eq. (6).

$$D_i^{mec}(t) = D_i^{trans}(t) + D_i^{edge}(t) \quad (6)$$

where, $D_i^{mec}(t)$ denotes total MEC offloading delay, $D_i^{trans}(t)$ denotes uplink data transmission delay, and $D_i^{edge}(t)$ denotes the delay in data processing. The system chooses the minimum possible delay based on the network condition, channel state, and available resources. The resulting estimated delay $D_i(t)$ is referred as shown in Eq. (7).

$$D_i(t) = min\left(D_i^{loc}(t), D_i^{mec}(t)\right) \quad (7)$$

This process is subsequently applied in QoS violation severity computation and offloading optimization decisions.

3) QoS violation severity modeling: In conventional MEC offloading systems, QoS is generally considered in a binary way, either a task meets or fails its deadline constraint. However, such modeling fails to capture the extent of violation, an important aspect in 5G/B5G IoT systems supporting heterogeneous and latency-sensitive applications. To overcome this limitation, the delay violation function $V_i(t)$ [35] of task i at the time slot t is defined as shown in Eq. (8).

$$V_i(t) = max(0, D_i(t) - D_i^{max}) \quad (8)$$

where, $D_i(t)$ is the actual or estimated time to complete the task and D_i^{max} is the maximum permissible task completion time. The function guarantees that only positive deadline deviations are considered, and the penalty for non-violated tasks is zero.

To further model heterogeneity in the IoT applications, each task is given a criticality factor C_i denoting the priority level and importance of the task. By combining delay violation magnitude and the importance of the task, the QVSI [36] is defined as follows in Eq. (9).

$$S_i(t) = C_i \cdot V_i(t) \quad (9)$$

where, $S_i(t)$ is the severity of the QoS violation of the task i. The final severity model is given as follows in Eq. (10).

$$S_i(t) = C_i \cdot max(0, D_i(t) - D_i^{max}) \quad (10)$$

A QoS violation is considered to occur when the task completion delay exceeds its deadline, i.e., $D_i(t) > D_i^{max}$. However, unlike conventional binary models, the proposed framework does not treat QoS violation as a binary event. Instead, the severity of the violation is quantified using the QVSI $S_i(t)$, which jointly captures both the magnitude of deadline violation and the task criticality. Therefore, although violation detection is based on deadline exceeding, the decision-making process is entirely driven by the severity value $S_i(t)$, rather than a binary QoS indicator. This formulation makes it possible to jointly consider the degree of violation of the deadline and the criticality of the task in the system design. The proposed QVSI serves as a key input for the Lyapunov-based optimization framework by updating virtual queues according to the severity and influencing real-time offloading decisions.

4) Severity-aware virtual queue construction: To guarantee long-term QoS violation control, a severity-aware virtual queue is introduced for each IoT device. The idea behind this virtual queue is to redefine the problem of managing QoS violations as a stochastic queue stability problem, allowing the use of Lyapunov optimization for offloading decisions at real-time [37].

In particular, let $S_i(t)$ be the QVSI for the task i in the time slot t. The severity-based virtual queue evolution $Z_i(t+1)$ is defined as shown in Eq. (11).

$$Z_i(t+1) = max[Z_i(t) + S_i(t), 0] \quad (11)$$

where, $Z_i(t)$ represents the accumulated severity backlog of the device i at time t with initial condition zero. The virtual queue accumulates the QoS violation severity over time, such that the higher the value indicates severe or persistent the deadline violations. This allows prioritization of high-severity tasks during scheduling and offloading decisions.

By including $Z_i(t)$ into the Lyapunov framework, the QoS violation minimization problem is converted into a queue stability problem. This enables real-time optimization of latency, energy consumption, and severity-aware QoS performance. This formulation dynamically penalizes continually violated tasks and provides long-term stability of the system with respect to its QoS.

5) Lyapunov drift-plus-penalty formulation: The proposed severity-aware MEC offloading framework minimizes the long-term QoS violation severity by stabilizing the severity-aware virtual queues, while simultaneously optimizing latency and energy consumption through the penalty component of the Lyapunov Drift-Plus-Penalty framework [38]. The system stability is characterized using a Lyapunov function $L(t)$

defined over the severity-aware virtual queues as shown in the Eq. (12).

$$L(t) = \frac{1}{2}\sum_{i=1}^{N} Z_i^2(t) \quad (12)$$

where, N is the number of the total IoT devices and $Z_i(t)$ is the backlog of virtual queues for IoT device i at time t determined by its severity. The term $Z_i^2(t)$ helps to ensure system stability by providing a higher penalty if the queue backlog is larger.

To jointly optimize system performance and queue stability, the one-slot conditional Lyapunov drift $\Delta L(t)$, is defined as shown in Eq. (13).

$$\Delta L(t) = \mathbb{E}[L(t+1) - L(t) | Z(t)] \quad (13)$$

where, $L(t+1)$ is the Lyapunov function value at the next time slot $t+1$ and $\mathbb{E}[\cdot]$ denotes the expectation taken over the system randomness, including the arrival of tasks, channel variations, and offloading decisions.

The Lyapunov drift inherently captures the accumulated QoS violation severity through the severity-aware virtual queue $Z_i(t)$, whose evolution is governed by the QoS Violation Severity Index $S_i(t)$ in Eq. (11). Therefore, minimizing the Lyapunov drift implicitly minimizes the long-term QoS violation severity. Consequently, the penalty term is introduced only for latency and energy consumption, avoiding redundant optimization of the severity objective. The resulting Drift-Plus-Penalty (DPP) function is expressed in Eq. (14).

$$\mathcal{O}_t = \Delta L(t) + V \cdot \mathbb{E}[\lambda P(t) + \mu T(t) | Z(t)] \quad (14)$$

where, $\mathcal{O}_t$ represents the combined Lyapunov drift and penalty function, $P(t) = \sum_{i=1}^{N} P_i(t)$ represents total energy consumption at time t, and $T(t) = \sum_{i=1}^{N} T_i(t)$ represents the total task latency in the system. The parameter $V > 0$ determines the trade-off between maintaining queue stability and optimizing system performance, while λ and μ ($\lambda + \mu = 1$) control the relative importance of latency and energy consumption in the penalty term. Specifically, smaller values of V place greater emphasis on stabilizing the severity-aware virtual queues, thereby reducing long-term QoS violation severity, whereas larger values prioritize the optimization of latency and energy consumption. In this work, the value of V was selected based on preliminary simulation experiments to achieve a balanced trade-off between QoS violation severity, latency, and energy consumption. Unlike the overall optimization objective in Eq. (1), which considers QoS violation severity, latency, and energy consumption, the Drift-Plus-Penalty formulation optimizes the QoS violation severity implicitly through the Lyapunov drift via the severity-aware virtual queues, while latency and energy consumption are optimized explicitly through the penalty term. Thus, the proposed optimization remains fully consistent with the long-term objective while enabling adaptive real-time MEC offloading in 5G/B5G IoT networks. This detailed process is shown in Fig. 3.

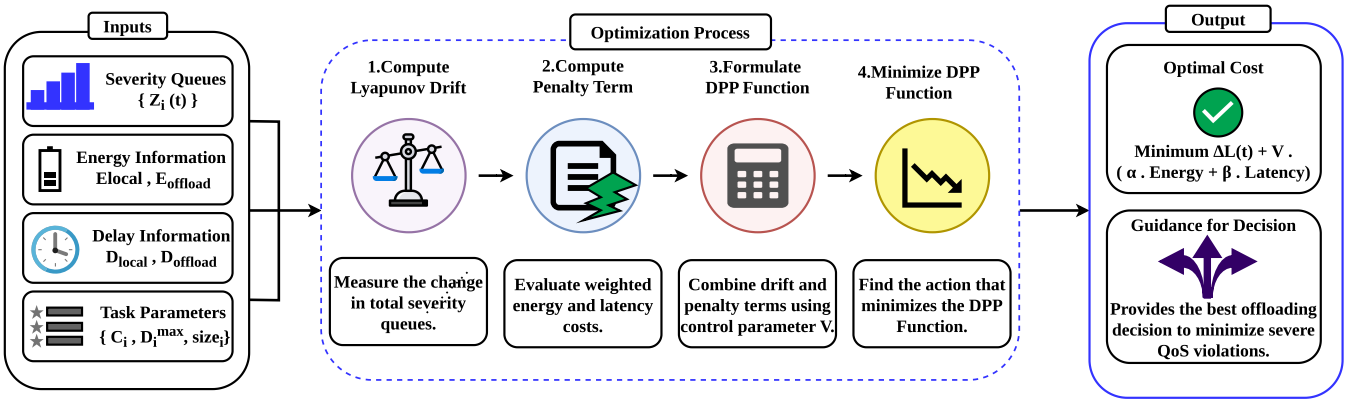

Fig. 3. Severity-aware Lyapunov Drift-Plus-Penalty (LDPP) optimization process in MEC-enabled 5G/B5G IoT networks.

Theoretical Stability Analysis: To theoretically justify the long-term stability of the proposed severity-aware MEC offloading framework, we analyze the system using Lyapunov optimization under stochastic network conditions.

The following standard assumptions are considered: 1) task arrivals follow a stochastic process with bounded first and second moments, 2) wireless channel states are time-varying and stochastic but remain bounded within a finite support, 3) task sizes, computational workloads, and delay deadlines are finite and bounded, and 4) system resources including computation and communication capacities are finite and constrained as defined in the system model.

Theorem 1 (Queue Stability Guarantee): Under the above assumptions, the proposed severity-aware Lyapunov Drift-Plus-Penalty (DPP) based offloading policy ensures that the severity-aware virtual queues $Z_i(t)$ are mean-rate stable. Furthermore, the time-average expected queue backlog remains bounded, guaranteeing long-term stability of the MEC system under stochastic task arrivals and time-varying wireless channel conditions.

Proof:

The Lyapunov function is defined in Eq. (15).

$$L(t) = \frac{1}{2}\sum_{i=1}^{N} Z_i^2(t) \quad (15)$$

The conditional Lyapunov drift is given by Eq. (16).

$$\Delta L(t) = \mathbb{E}[L(t+1) - L(t) \mid Z(t)] \quad (16)$$

Using the queue evolution, it is expressed as shown in Eq. (17).

$$Z_i(t+1) = \max[Z_i(t) + S_i(t), 0] \quad (17)$$

The drift can be upper-bounded as shown in Eq. (18).

$$\Delta L(t) \leq B + \sum_{i=1}^{N} Z_i(t)\mathbb{E}[S_i(t) \mid Z(t)] \quad (18)$$

where, B is a finite constant due to bounded severity values. The proposed DPP policy minimizes as expressed in Eq. (19).

$$\Delta L(t) + V \cdot \mathbb{E}[\lambda P(t) + \mu T(t) \mid Z(t)] \quad (19)$$

This ensures that the Lyapunov drift is driven toward stability while simultaneously optimizing latency and energy consumption. Since the severity-aware virtual queue is updated using bounded QVSI values, repeated minimization of the drift guarantees that the time-average queue backlog remains finite, as expressed in Eq. (20).

$$\lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}[Z_i(t)] < \infty \quad (20)$$

which implies mean-rate stability of all virtual queues. Thus, the proposed framework ensures long-term queue stability under stochastic task arrivals and time-varying wireless channels while jointly optimizing QoS violation severity, latency, and energy consumption in real time.

6) Severity-aware offloading decision policy: Based on the instantaneous system state, each IoT task chooses one of the execution modes among local processing, MEC offloading, or cooperative MEC execution. The decision is formulated as an LDPP minimization problem that combines queue stability, latency and energy consumption.

Let the three binary decision variables $x_i^{loc}(t)$, $x_i^{mec}(t)$, and $x_i^{coop}(t)$ for the three modes of execution. The constraint is enforced by the system, which is represented by Eq. (21).

$$x_i^{loc}(t) + x_i^{mec}(t) + x_i^{coop}(t) = 1, \forall_i \in N \quad (21)$$

At each time slot t, the system solves the following instantaneous optimization problem under the LDPP framework as shown in Eq. (22).

$$\min_{x_i^{loc}(t), x_i^{mec}(t), x_i^{coop}(t)} \Delta L(t) + V \cdot \mathbb{E}[\alpha E(t) + \beta T(t) \mid Z(t)] \quad (22)$$

subject to execution mode constraint (15), computation resource constraint: $\sum f_i(t) \leq F^{max}$, communication bandwidth constraint: $\sum b_i(t) \leq B^{max}$, binary decision constraints: $x_i^{loc}(t), x_i^{mec}(t), x_i^{coop}(t) \in \{0,1\}$. Where $Z(t)$ represents the severity-aware virtual queue vector. Tasks with higher queue backlog values $Z_i(t)$ are implicitly prioritized due to their stronger contribution to the Lyapunov drift term. The optimal decision is obtained by evaluating the drift-plus-penalty cost corresponding to each execution mode and selecting the mode that minimizes the objective function. This converts the long-term stochastic optimization problem into a real-time per-slot control policy suitable for dynamic 5G/B5G MEC environments.

In local execution, tasks are processed directly at the IoT device using its available computational resources. In MEC offloading, tasks are transmitted to the nearest edge server via the base station for remote execution. In cooperative MEC execution, tasks are partitioned into smaller subtasks and distributed across multiple MEC servers according to their current computational load and queue states. Coordination among MEC servers is achieved through lightweight exchange of control information, such as queue length and resource availability, via an edge orchestration layer. This coordination introduces negligible overhead compared to task execution time and is not explicitly modeled in delay analysis. Inter-edge communication is assumed to occur over a high-speed wired backhaul, and its delay is incorporated into the overall MEC execution delay while remaining significantly lower than wireless transmission delay. The proposed approach dynamically responds to channel conditions, queue states, and task urgency, allowing real-time severity-aware offloading, while reducing the severity of QoS violation, energy consumption, and latency.

7) Task execution and system operation: After the offloading decision $x_i^{loc}(t)$, $x_i^{mec}(t)$, and $x_i^{coop}(t)$ is determined, each task on the IoT is implemented based on the selected mode. In the local execution mode, the tasks are executed locally at the IoT device using its available computational resources. In the MEC offloading mode, tasks are transmitted to the nearest edge server via the base station, while in the cooperative MEC mode, tasks are distributed among multiple edge servers for load balancing. The overall delay of the executions is controlled by a transmission delay and a computation delay, depending on the mode used. Each task requires a certain amount of energy given by Eq. (23).

$$E_i(t) = \begin{cases} E_i^{loc}(t), & if\ executed\ locally \\ E_i^{tx}(t) + E_i^{mec}(t) & if\ offloaded\ to\ MEC \end{cases} \quad (23)$$

where, $E_i^{tx}(t)$ is the transmission energy, $E_i^{loc}(t)$ is the local computation energy, and $E_i^{mec}(t)$ is the energy consumption at MEC. For cooperative MEC execution, the total energy consumption is modeled as the sum of distributed computation energy across participating MEC servers, along with the corresponding (aggregated) communication cost among them. Lastly, the proposed severity model is used to evaluate the severity of QoS violation $S_i(t)$ to represent performance degradation. All metrics are stored over time and evaluated the system performance under the dynamic 5G/B5G IoT environments.

8) Queue update and continuous operation: After task execution, the severity-aware virtual queues are updated to reflect the new observed level of QoS violations, which increases the contribution of tasks with higher delays or recurring violations to the future scheduling decisions. The system operates in a continuous time-slot-based manner where the arrival of IoT tasks, channel conditions, and resources change dynamically over time. At every time slot, updates value of the queues and system parameters are returned to the Lyapunov-based optimization framework, allowing adaptive

and real-time offloading decisions. This continuous feedback mechanism ensures the proposed framework remains responsive to the changing network conditions in 5G/B5G IoT environments while maintaining the long-term stability of the system. As a result, the system effectively balances the tradeoff between latency, energy consumption, and severity-aware QoS requirements, ensuring high performance in highly dynamic and heterogeneous IoT workloads.

IV. Experimental Analysis and Discussion

The simulation is implemented in a Python-based discrete-time environment that models a 5G/B5G IoT network consisting of 100 heterogeneous IoT devices and 5 MEC servers over 1000 time slots. Each IoT device generates stochastic computation tasks with varying task sizes, CPU cycle requirements, latency deadlines, and criticality levels. The task arrivals are modeled using a Poisson arrival process with an arrival rate ranging from 0.05 to 0.15 tasks/ms, which captures the stochastic nature of heterogeneous IoT traffic. Task sizes are uniformly distributed between 0.1 MB and 10 MB, while CPU cycle requirements and latency deadlines are randomly assigned within the ranges listed in Table I to represent diverse URLLC and mMTC applications. The simulation primarily considers independent stochastic task arrivals rather than bursty or temporally correlated traffic, enabling a controlled evaluation of the proposed severity-aware MEC offloading framework under heterogeneous workload conditions. To ensure statistical reliability, all results are obtained by performing multiple independent simulation runs under identical settings, and the final reported values are computed as the average over repeated trials with different random seeds.

TABLE I. System Hyperparameters used in the Proposed Severity-Aware MEC Offloading in 5G/B5G IoT Environment

Category	Parameter	Value
Lyapunov Control Parameters	V (drift penalty weight)	10
	α (energy weight)	0.5
	β (latency weight)	0.5
	Queue update factor	severity-based (QVSI)
QVSI (Severity Model)	Criticality weights	High=3, Medium=2, Low=1
Network / Communication Settings	Bandwidth	20 MHz
	Transmission rate	100 Mbps
	Tx power	0.5 W
	Channel model	stable AWGN (typical assumption)
Compute Resources	IoT Device CPU	1 GHz
	MEC Server CPU	20 GHz
	MEC servers	5 nodes
	Task arrival rate	0.05 – 0.15 tasks/ms
Task Model	Task size	0.1 – 10 MB
	CPU cycles	0.5 – 2 GHz cycles equivalent
	Deadline range	50 – 500 ms
	Task types	URLLC + mMTC mixed

This approach captures system variability arising from stochastic task arrivals, channel conditions, and queue dynamics, and improves the robustness of the performance evaluation. For each task, both local execution delay and MEC execution delay, including transmission, computation, and queueing delays, are estimated to emulate realistic MEC operations. Table I details the parameters employed in the experimental analysis.

A. Performance Evaluation of the Proposed Framework

The performance evaluation reveals that the proposed MEC offloading mechanism with severity awareness results in substantial improvements in terms of the various QoS metrics and the efficiency of the system usage scenarios for the 5G/B5G IoT environment. The system has an average task delay of 82ms and a maximum worst-case delay of 280ms. This indicates that the latency can be controlled even during high network traffic.

The 4.8% QOS violation rate confirms the effectiveness of the QoS Violation Severity Index (QVSI) to reduce deadline misses. The 98.5% SLA satisfaction rate shows reliable service delivery for different IoT applications. The obtained system throughput of 103 tasks per second shows strong scalability when tasks are densely injected into the system. Energy efficiency is also significantly improved, with an average consumption of 6.0mJ per slot due to optimized Lyapunov drift-plus-penalty control. The workload distribution of the CPU between MEC (70%) and IoT devices (38%) shows effective workload offloading and a lower burden on the IoT devices.

The Jain fairness index value of 0.955 indicates that the resources are distributed fairly for the tasks. An offloading ratio of 72% shows that most of the tasks get completed through the utilization of MEC. The queue lengths are observed as stable, being between 2.8 and 5.0, which suggests stability and convergence in the network. This proves that the proposed framework maintains controlled congestion and efficient long-term operation.

1) Overall system performance evaluation:

Fig. 4 evaluates the proposed severity-aware Lyapunov-based MEC offloading algorithm in 5G/B5G IoT networks.

Fig. 4(a) shows that the system throughput steadily increases. This proves the efficiency of the framework to manage tasks and improve resource efficiency.

Fig. 4(b) exhibits a steady decline in average energy consumption per slot. This indicates the energy-efficient offloading decisions.

Fig. 4(c) illustrates a decrease in QoS violation rate. It validates the effectiveness of the proposed QVSI and severity-aware optimization.

Fig. 4(d) represents a rapid rise in the task completion rate. It proves improved reliability and latency control.

Fig. 4(e) shows a very high Jain's Fairness Index, which confirms fairness in resource sharing. The results prove that the proposed framework successfully minimizes severe QoS violations while improving throughput, energy efficiency, reliability, and fairness in MEC-enabled 5G/B5G IoT systems.

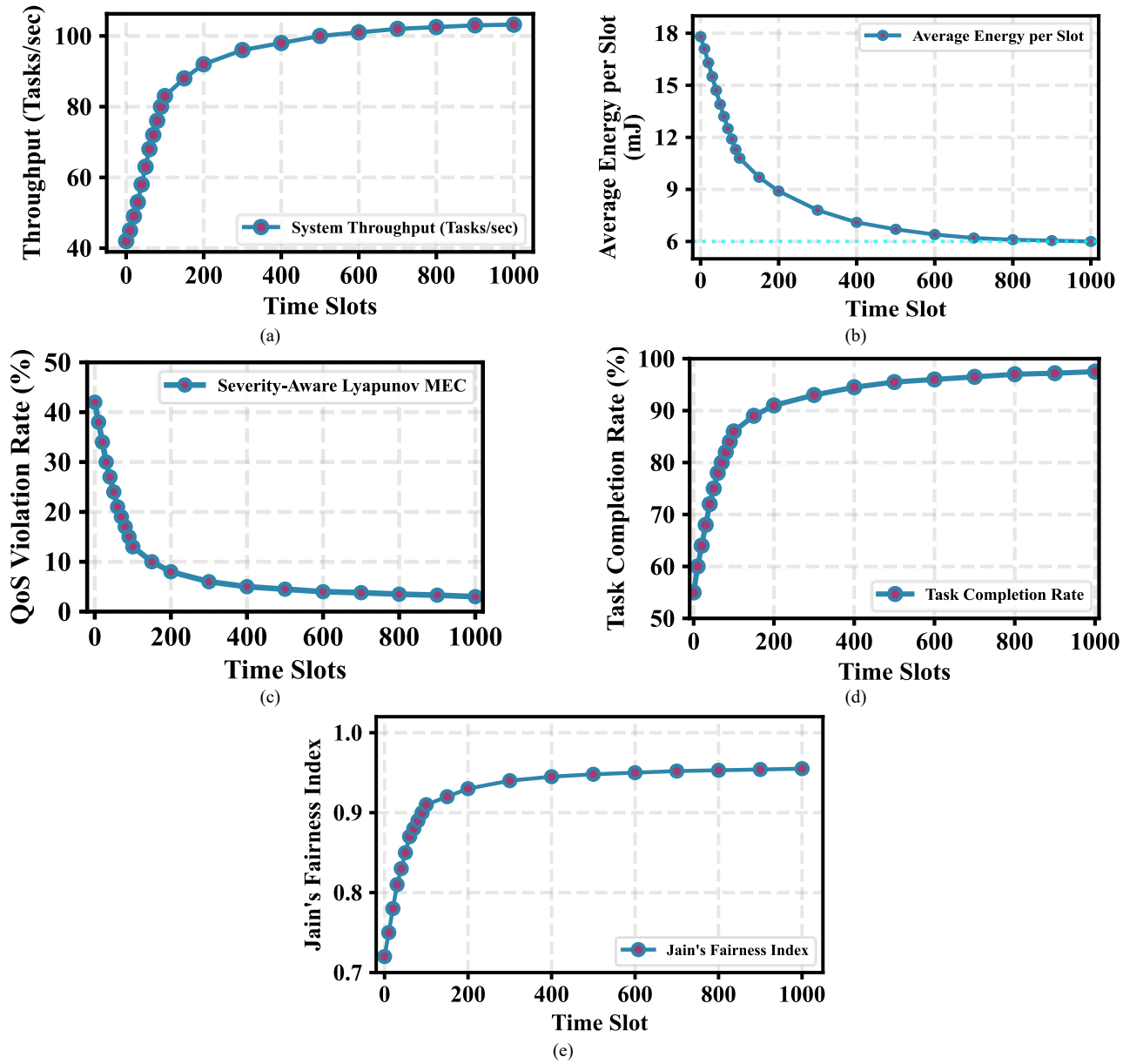

Fig. 4. Performance evaluation of the proposed severity-aware Lyapunov MEC offloading framework: (a) system throughput, (b) average energy per slot, (c)QoS violation rate, (d) task completion rate, and (e) Jain's fairness index over time slots.

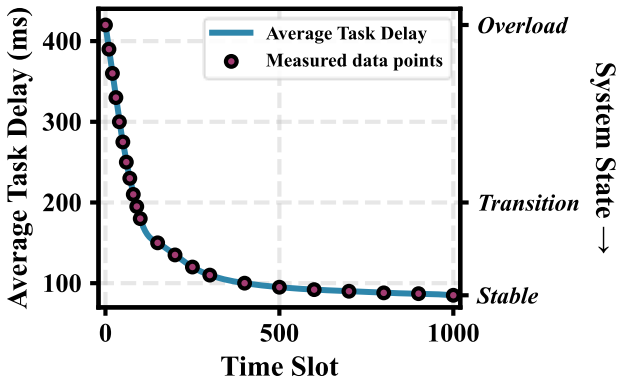

Fig. 5. Convergence behavior of the proposed severity-aware Lyapunov MEC offloading algorithm.

2) Convergence and queue stability analysis: Fig. 5 shows how effectively the proposed severity-aware LDPP MEC offloading approach converges in a 5G/B5G IoT environment. Initially, the system operates under overload conditions with a high average task delay. As the optimization process progresses, the severity-aware virtual queues stabilize. Hence, the delay is significantly reduced as we move towards the stability period. Finally, the system converges to a steady state with minimal delay. This proves that the proposed approach provides minimal task delays, stability, and effective MEC offloading performance for heterogeneous IoT applications.

Fig. 6 depicts the normalized virtual queue length change under the severity-aware Lyapunov MEC offloading framework in a 5G/B5G IoT environment. When overload occurs, the queue

rapidly increases because of the congestion and high task arrival rate in the overload phase. As the Lyapunov Drift-Plus-Penalty optimization becomes active, the queue length gradually decreases and converges toward a stable low-backlog state.

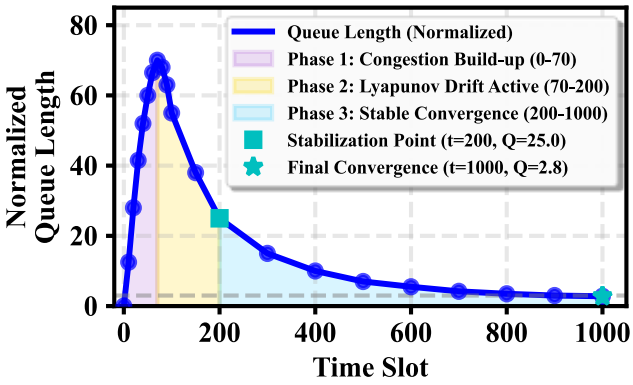

Fig. 6. Convergence of severity-aware virtual queue length in the proposed Lyapunov MEC framework.

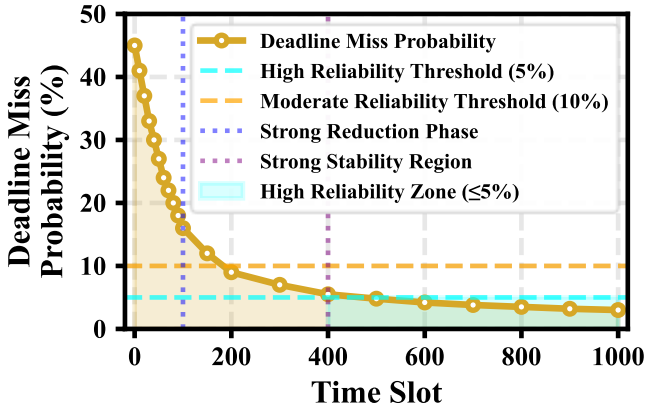

Fig. 7. Temporal reduction of deadline miss probability in the proposed severity-aware MEC offloading framework.

The stabilization and final convergence points indicate good queue stabilization and system stability. This validates the proposed algorithm's ability to overcome congestion, to achieve congestion-aware queue stabilization, and ensure reliable MEC task scheduling under dynamic IoT workloads.

Fig. 7 depicts the assessment of the proposed severity-aware Lyapunov-based MEC offloading method using the evaluation of deadline miss probability over 1000 time slots in a 5G/B5G IoT environment.

The system starts with relatively large deadline miss probabilities due to large task arrivals and queue instabilities. However, as the optimization framework adapts, the deadline misses probability falls sharply in the strong reduction period. It provides efficient decisions for task scheduling and offloading. The system reaches the optimal high-reliability state after around 400 time slots, with the probability of missing deadlines kept below 5%. This shows that the proposed QVSI-based approach reduces severe QoS violations. The results assure a high reliability and stability level for heterogeneous and latency-sensitive IoT systems. The performance of the proposed severity-aware Lyapunov-based MEC offloading approach in mitigating the severity of QoS violation has been illustrated in Fig. 8. The average value of QVSI is large in the beginning because of heavy deadline violations.

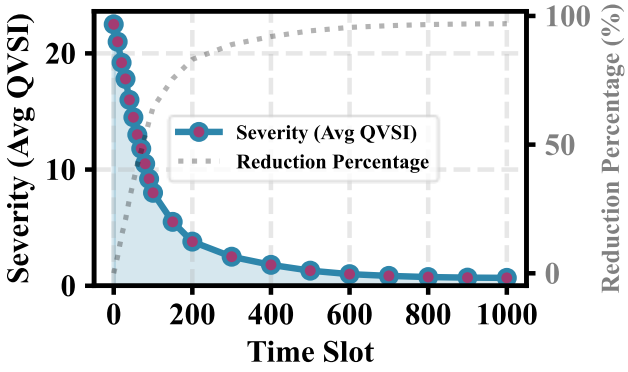

Fig. 8. Convergence of average QoS violation severity (QVSI) and reduction percentage over 1000 time slots.

3) QoS violation severity and reliability analysis: But it rapidly decreases as the algorithm learns optimal offloading decisions and prioritizes critical IoT tasks. Simultaneously, the reduction percentage steadily increases from approaching near-optimal performance. This proves the proposed framework reduces severe QoS violations, stabilizes the system, and improves service reliability for heterogeneous latency-sensitive IoT applications in 5G/B5G MEC networks while maintaining resource management. The distribution of IoT tasks is shown in Fig. 9, according to different QVSI ranges in the proposed severity-aware MEC framework.

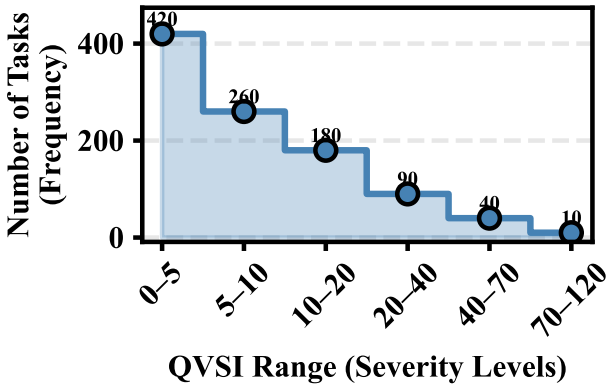

Fig. 9. Frequency distribution of QoS violation severity index (QVSI) across IoT tasks in the proposed severity-aware Lyapunov-based MEC offloading framework.

The majority of tasks are in the low-severity range (0-5), which indicates that most tasks do not suffer from high or even any deadline violations. The number of tasks decreases as the QVSI level increases, with only a few tasks suffering from severe QoS violations above 70. This shows the capability of the proposed Lyapunov-based MEC offloading algorithm in mitigating critical cases of deadline misses and prioritizing critical IoT tasks in congested 5G/B5G network environments.

4) Delay performance analysis under dynamic network conditions: Fig. 10 shows the average task delay for executing

the task locally and offloading it to the MEC in 5G/B5G IoT networks for different network loads. As the level of network load increases from low load to maximum congestion, the delay of local execution rises from 380ms to 620ms due to limited resource capabilities. In comparison, MEC offloading has lower delays, which grow incrementally from 120ms to 210ms. This shows that MEC-enabled offloading effectively reduces latency and improves the QoS performance under heavy traffic conditions, thus supporting latency-sensitive and critical IoT applications.

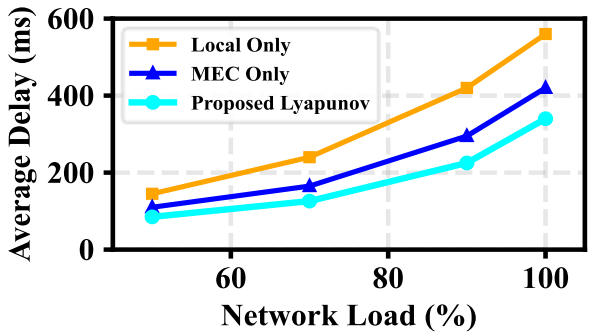

Fig. 10. Performance comparison of local computing, MEC offloading, and proposed severity-aware Lyapunov MEC offloading algorithm in terms of average delay under increasing network load.

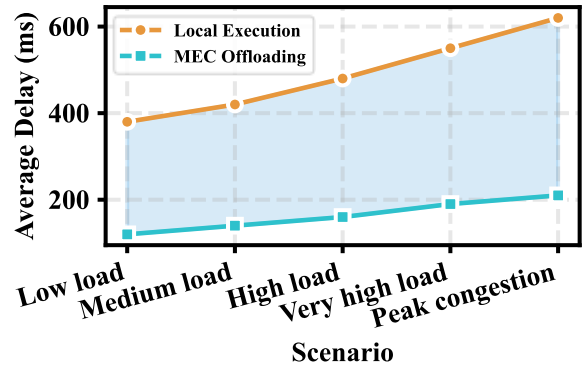

Fig. 11. Average delay comparison between local execution and MEC offloading under different network load conditions.

Fig. 11 shows the average task delay versus the increasing network load for Local Only, MEC Only, and the proposed Lyapunov scheme. The delay of all schemes increases with the network load due to congestion and limited resources. The Local Only scheme has the largest delay due to the low processing capability of IoT devices. The MEC-only scheme reduces the delay using the edge servers but still suffers from the heavy load. The proposed severity-aware Lyapunov algorithm achieves the lowest delay by intelligently balancing local execution and MEC offloading while prioritizing critical tasks. This proves that the proposed framework provides better QoS and latency performance in 5G/B5G IoT networks.

Fig. 12 depicts the delay variation of critical and non-critical IoT tasks over different time slots with the proposed framework. At first, the critical tasks suffer from higher delay due to their strict QoS requirement and higher processing priority, but as time progresses, the delays for both types of tasks are significantly reduced due to the severity-aware Lyapunov optimization that efficiently handles the queue congestion and resource allocation. The delay gap between critical and non-critical tasks also reduces over time, which depicts fair and stable scheduling.

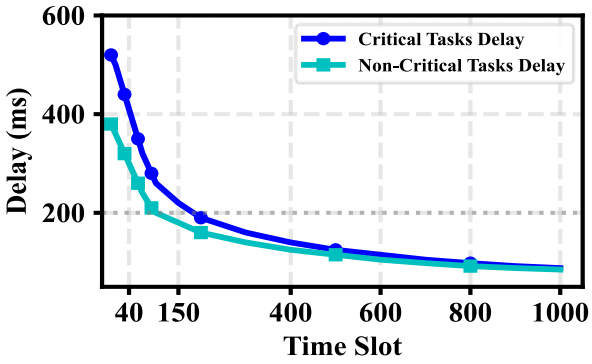

Fig. 12. Delay performance comparison between critical and non-critical IoT tasks over time slots using the proposed severity-aware Lyapunov MEC offloading framework in 5G/B5G networks.

This validates the effectiveness of the proposed algorithm to prioritize critical tasks while simultaneously maintaining the overall QoS stability and low latency in 5G/B5G IoT networks.

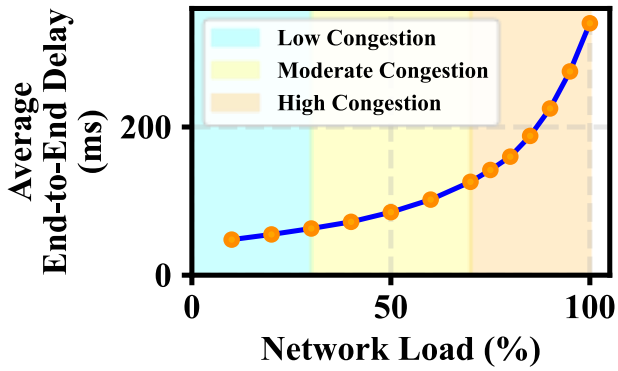

Fig. 13. Impact of network congestion levels on average end-to-end delay in the proposed severity-aware MEC offloading framework for 5G/B5G IoT networks.

Fig. 13 shows how the average end-to-end delay increases with network load under low, moderate, and high congestion conditions. In low congestion, the delays will be kept to a minimum due to abundant network and MEC resources. With an increase in congestion in terms of low, medium, and high, the delay will increase because of the increased queue formation, transmission delays, and scarcity of resources. The sudden jump towards the point where the network is full denotes the impact of heavy congestion on the QoS. This proves network congestion affects delay and demonstrates the importance of the proposed severity-aware Lyapunov MEC offloading framework in managing congestion and maintaining QoS in 5G/B5G IoT systems.

5) Energy efficiency analysis: Fig. 14 compares the comparison between the energy consumption for computation and transmission of the Local offloading, the conventional MEC, and the newly proposed severity aware MEC offloading

strategy. The results show that local execution consumes the highest computation energy (18.5mJ), while the proposed method reduces both computation and transmission energy compared to conventional MEC. This proves that the modified Lyapunov-based severity-aware offloading algorithm efficiently balances local processing and MEC offloading decisions, minimizing overall energy consumption while maintaining QoS requirements for heterogeneous 5G/B5G IoT applications.

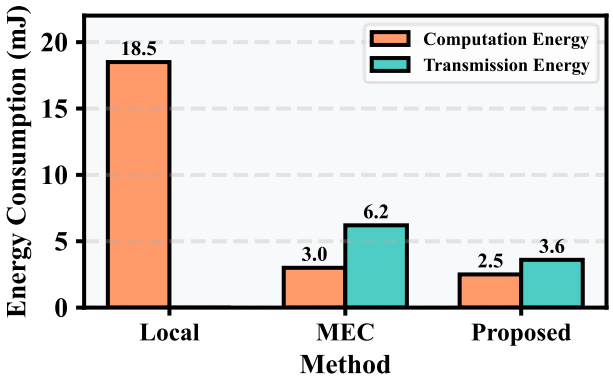

Fig. 14. Energy consumption comparison of local execution, conventional MEC, and proposed severity-aware MEC offloading method.

6) Sensitivity analysis: Table II presents a sensitivity analysis of the Lyapunov control parameter V on key performance metrics, including average task delay, energy consumption, QoS violation rate, and Jain's fairness index. As V increases from 1 to 50, the system achieves improved energy efficiency and reduced QoS violation rates due to stronger emphasis on penalty optimization in the Drift-Plus-Penalty framework.

TABLE II. SENSITIVITY ANALYSIS OF LYAPUNOV CONTROL PARAMETER V

V Value	Average Task Delay (ms)	Energy Consumption (mJ)	QoS Violation Rate (%)	Jain's Fairness Index
1	96	6.8	7.1	0.931
5	88	6.3	5.6	0.944
10	82	6.0	4.8	0.955
20	80	5.9	4.6	0.952
50	79	5.8	4.5	0.947

However, higher values of V slightly affect delay performance and fairness after a threshold, indicating a trade-off between latency minimization and system stability. The results demonstrate that $V = 10\text{-}20$ provides a balanced performance across all metrics.

B. Ablation Analysis

Table III evaluates the contribution of each major component of the proposed severity-aware MEC offloading framework. Different models are compared with the complete MEC offloading framework. The removal of the QVSI variant shows higher delay, energy consumption, and QoS violation rate, which proves the significance of the QVSI for quantifying and minimizing severe deadline violations. Likewise, removing virtual queues based on severity causes an improvement in throughput and fairness, highlighting that severity queue modeling improves the stability of a virtual queue and prioritizes critical tasks.

TABLE III. ABLATION ANALYSIS OF THE PROPOSED SEVERITY-AWARE MEC OFFLOADING FRAMEWORK IN 5G/B5G IOT NETWORKS

Method Variant	Average Task Delay (ms)	System Throughput (tasks/sec)	Average Energy Consumption (mJ)	Jain's Fairness Index	QoS Violation Rate (%)
w/o QVSI	105.0 ± 5.8	88.0 ± 3.2	7.8 ± 0.4	0.910 ± 0.010	9.5 ± 0.7
w/o Severity Queues	112.0 ± 6.1	85.0 ± 3.0	8.2 ± 0.5	0.895 ± 0.012	11.2 ± 0.8
w/o LDPP	128.0 ± 7.0	79.0 ± 3.5	9.6 ± 0.6	0.860 ± 0.015	14.8 ± 1.0
w/o Task Criticality	118.0 ± 6.5	83.0 ± 3.1	8.7 ± 0.5	0.882 ± 0.011	12.6 ± 0.9
Local Execution Only	155.0 ± 8.2	65.0 ± 2.8	12.5 ± 0.8	0.790 ± 0.018	22.0 ± 1.3
Full Proposed Model	**82.0 ± 4.6**	**103.0 ± 3.4**	**6.0 ± 0.3**	**0.955 ± 0.008**	**4.8 ± 0.5**

By removing the LDPP configuration, the worst performance is obtained among the algorithmic variants in terms of delay and QoS violation rate, showing that the optimal configuration of LDPP is a crucial and effective component for efficient online offloading and resource management. Without the Task Criticality model, performance begins to drop significantly, especially in fairness and QoS violation rate, indicating the significance of addressing heterogeneous task priority in latency-sensitive IoT applications. The Local Execution Only baseline also suffers from the highest delay, energy consumption, and QoS violations as tasks are executed only on IoT devices without the support of MEC, causing resource congestion and inefficient task execution. In contrast, the full proposed model achieves the highest energy efficiency, the highest throughput, the lowest delay, improved fairness, and a minimal QoS violation rate. These results show that the combination of using QVSI, severity-aware queues, Lyapunov optimization, and task criticality improves the reliability and QoS performance in 5G/B5G MEC-supported IoT networks.

C. Comparative Analysis

To ensure a fair comparison, all existing methods are implemented under identical simulation settings as the proposed framework. Table IV shows a comparison between the proposed severity-aware MEC offloading framework and existing state-of-the-art methods in terms of task delay, throughput, energy consumption, fairness, and QoS violation rate in 5G/B5G IoT networks. The results indicate that the traditional methods, such

as DRL-DDQN, HATO, PSO, Lyapunov optimization with PMP, and DNNs-RL, result in moderate performance gains, but still incur high latency and QoS violations due to limited accounting for the severity of the violation and the criticality of the task. Conversely, the proposed method achieves the lowest average task delay (82ms), minimum energy consumption (6.0mJ), the highest throughput (103 tasks/sec), and the best fairness index (0.955), and also reduces the rate of violation of QoS to 4.8%. The improvements demonstrate that the proposed framework enables more efficient and reliable MEC task offloading in 5G/B5G environments for heterogeneous and latency-sensitive IoT applications.

TABLE IV. COMPARATIVE PERFORMANCE ANALYSIS OF THE PROPOSED SEVERITY-AWARE MEC OFFLOADING METHOD AGAINST EXISTING APPROACHES

Reference	Method	Average Task Delay (ms)	System Throughput (tasks/sec)	Average Energy Consumption (mJ)	Jain's Fairness Index	QoS Violation Rate (%)
Zhai et al. [25] (2024)	DRL-DDQN	95.0 ± 5.2	92.0 ± 3.0	7.2 ± 0.4	0.920 ± 0.009	8.5 ± 0.6
Benbraika et al. [26] (2024)	HATO	110.0 ± 6.0	85.0 ± 2.9	8.5 ± 0.5	0.900 ± 0.011	11.0 ± 0.8
Alam et al. [27] (2024)	PSO	108.0 ± 5.8	87.0 ± 3.1	8.1 ± 0.4	0.905 ± 0.010	10.2 ± 0.7
Luo et al. [29] (2025)	Lyapunov optimization with PMP algorithm	100.0 ± 5.5	90.0 ± 3.0	7.6 ± 0.4	0.915 ± 0.009	9.0 ± 0.6
Yang et al. [30] (2024)	DNNs-RL	92.0 ± 5.0	94.0 ± 3.2	7.0 ± 0.3	0.925 ± 0.008	7.8 ± 0.5
Proposed	**LDPP**	**82.0 ± 4.6**	**103.0 ± 3.4**	**6.0 ± 0.3**	**0.955 ± 0.008**	**4.8 ± 0.5**

D. Discussion

The complex environment of 5G and B5G IoT introduces significant challenges in designing an efficient and reliable MEC task offloading strategy. Current frameworks mainly use binary models of QoS evaluation, which do not consider task performance as being either satisfied or violated. This approach fails to capture the level of deadline violations, especially for latency-sensitive and mission-critical IoT applications. Furthermore, most of the conventional approaches consider optimizing mean QoS metrics without the explicit consideration of extreme QoS degradation scenarios. Moreover, lack of information about task heterogeneity, in particular the criticality of the tasks, often leads to unfair scheduling decisions and poor performance of high-priority tasks. The dynamic nature of networks, such as varying wireless channels, computation loads, and resource-limited IoT devices, further complicates efficient task offloading and requires relying on static decision-making strategies in real-world MEC systems. To tackle these challenges, this study introduces a severity-aware MEC task offloading framework for heterogeneous 5G/B5G IoT environments. The main idea is to introduce a QVSI that not only estimates whether a task misses its deadline but also the magnitude of the violation, while including task criticality in the evaluation process. This achieves a more realistic representation of QoS degradation in the real IoT scenario. Moreover, a severity-aware virtual queue model is designed to capture system backlog as violation intensity instead of just tasks, enabling fine-grained control of system stability. These components are combined in a modified LDPP optimization framework to allow real-time and adaptive offloading decisions. The framework jointly minimizes latency, energy, and QoS violation severity while ensuring long-term stability of queues under dynamic network conditions.

The proposed objectives are verified by extensive simulation experiments conducted in a Python-based discrete-time MEC environment. The results show that the framework has significant improvements in system performance across various metrics. The average task delay is decreased to 82ms, and the mean energy consumption is minimized to 6.0mJ/slot. The deadline miss rate is significantly reduced to 4.8%, indicating an effective deadline miss mitigation. Moreover, the system achieves an efficient and fair performance with a Jain's fairness index of 0.955 and a throughput of 103 tasks/second among the heterogeneous IoT devices. The QoS violation severity is reduced by about 97%, confirming good convergence performance and the effectiveness of the severity-aware Lyapunov optimization mechanism in reducing the severity of the QoS violation under various workloads.

Overall, these results validate that the proposed model achieves its objectives of minimizing severe QoS violations while optimizing latency, energy efficiency, and fairness. However, the proposed framework relies on accurate knowledge of system state information, such as channel conditions, task characteristics, and computational requirements, which may not always be fully reliable in highly dynamic or partially observable environments. In addition, the current study assumes an AWGN-based communication model for performance evaluation, which provides a simplified and controlled baseline but does not capture realistic wireless fading dynamics in practical 5G/B5G systems. Future work could include the application of learning techniques such as deep reinforcement learning (DRL) or federated learning to improve adaptability under uncertainty and incomplete system information. Moreover, the framework can be extended to more realistic wireless channel models such as Rayleigh and Rician fading, as well as real-world edge computing testbeds. Mobility-aware and energy harvesting IoT models can also be incorporated to further enhance the applicability, robustness, and practical utility of the proposed framework for future 5G/B5G networks.

V. CONCLUSION

This work proposes a severity-aware MEC task offloading framework for 5G/B5G IoT that effectively resolves the limitations of the conventional binary QoS and non-criticality-aware task offloading methods. The framework introduces a QVSI and severity-aware virtual queuing integrated with a modified LDPP optimization model for real-time adaptive offloading decisions. This allows joint optimization of latency,

energy, and violation of QoS, and ensures stability of the system under dynamic network conditions. The excellent performance of the proposed approach is confirmed through simulation, where the average task delay is 82 ms, the energy consumption per slot is 6.0 mJ, the violation of the QoS of 4.8%, the system throughput is 103 tasks/sec, and the fairness index is 0.955. The significant decrease in QoS severity of violation illustrates strong convergence and reliability. Overall, the proposed framework ensures efficient, stable, and severity-aware resource allocation in MEC-enabled 5G/B5G IoT, offering a promising solution for future intelligent edge computing applications.

STATEMENTS AND DECLARATIONS

Author contributions: All authors contributed to the conception of the problem setting and overall design of the work. S.S.R built the conceptualization and methodology. R.S. implemented the work and writing. Authors involved in the visualization of concepts. This version was approved by all authors, who also read and approved the manuscript.

Funding: No funding was received for conducting this study.

Conflict of interest: The authors declare that they have no conflict of interest.

Availability of data and materials: The data set is not available; only a simulation with parameters is used.

Ethical approval: The research is original, and the authors of this manuscript created all the figures and tables.

Consent to participate: Not applicable.

Consent for publication: All authors agree with the submission of the manuscript to this journal and possible publication afterwards.

REFERENCES

[1] M. H. Alanazi, “Machine Learning-based Secure 5G Network Slicing: A Systematic Literature Review,” Int. J. Adv. Comput. Sci. Appl., vol. 14, pp.377, 2023.

[2] C. Zhiyi, M. Aman, and A. Hafizah, “A Review on NS Beyond 5G: Techniques, Applications, Challenges and Future Research Directions,” Int. J. Adv. Comput. Sci. Appl., vol. 15, pp. 117, 2024.

[3] T. Senevirathna, V. H. La, S. Marcha, B. Siniarski, M. Liyanage, and S. Wang, “A survey on XAI for 5G and beyond security: Technical aspects, challenges and research directions,” IEEE Commun. Surv. Tutor., vol. 27, pp. 941–973, Aug. 2024.

[4] Kaur and A. Godara, “Machine learning empowered green task offloading for mobile edge computing in 5G networks,” IEEE Trans. Netw. Serv. Manag., vol. 21, pp. 810–820, July 2023.

[5] L. Peng, P. H. Ho, and K. Zhao, “Task offloading in terrestrial-support-free multi-layer multi-access edge computing,” IEEE Commun. Mag., vol. 62, pp. 82–87, July 2024.

[6] M. A. Khan, N. Kumar, S. H. Alsamhi, G. Barb, J. Zywiołek, I. Ullah, F. Noor, J. A. Shah, and A. M. Almuhaideb, “Security and privacy issues and solutions for UAVs in B5G networks: A review,” IEEE Trans. Netw. Serv. Manag., vol. 22, pp. 892–912, Oct. 2024.

[7] A. Darwin Jose Raju, S. Solai Manohar, T. C. Jermin Jeaunita, Energy Prediction and Scheduling in Battery-Less Backscatter Sensor Network for Sustainable Network Management, Pervasive and Mobile Computing, 2026, 102234, ISSN 1574-1192.

[8] R. Huang, W. Wen, X. Chen, Z. Zhou, Q. Chen, and C. Dong, “Joint power allocation and task replication for reliability-sensitive services in NOMA-enabled vehicular edge computing,” IEEE Trans. Veh. Technol., vol. 73, pp. 4178–4193, Oct. 2023.

[9] H. Guo, Y. Wang, J. Liu, and C. Liu, “Multi-UAV cooperative task offloading and resource allocation in 5G advanced and beyond,” IEEE Trans. Wirel. Commun., vol. 23, pp. 347–359, May 2023.

[10] S. Awoyemi and B. T. Maharaj, “Adaptive power management for multiaccess edge computing-based 6G-inspired massive Internet of Things,” IET Wirel. Sens. Syst., vol. 15, p. e70000, Jan. 2025.

[11] D. Logeshwari, S. Asha, T. C. Jermin Jeaunita, M. Povaneswari, "Proactive Channel Assignment and Modified CR-AODV for Stability-Driven Multi-Hop Routing in Cognitive Radio Networks," Radioengineering, vol. 34, Dec 2025.

[12] E. F. Maleki, W. Ma, L. Mashayekhy, and H. J. La Roche, “QoS-aware content delivery in 5G-enabled edge computing: Learning-based approaches,” IEEE Trans. Mob. Comput., vol. 23, pp. 9324–9336, Feb. 2024.

[13] H. W. Kao and E. H. K. Wu, “QoE sustainability on 5G and beyond 5G networks,” IEEE Wirel. Commun., vol. 30, pp. 118–125, Mar. 2023.

[14] G. Nieto, I. De la Iglesia, U. Lopez-Novoa, and C. Perfecto, “Deep reinforcement learning techniques for dynamic task offloading in the 5G edge-cloud continuum,” J. Cloud Comput., vol. 13, pp. 94, May 2024.

[15] A. Elgendy, S. Meshoul, and M. Hammad, “Joint task offloading, resource allocation, and load-balancing optimization in multi-UAV-aided MEC systems,” Appl. Sci., vol. 13, pp. 2625, Feb. 2023.

[16] C. Xu, M. Lv, K. Zhang, K. Cao, G. Wang, M. Wei, and B. Peng, “Energy consumption and time-delay optimization of dependency-aware tasks offloading for industry 5.0 applications,” IEEE Trans. Consum. Electron., vol. 70, pp. 1590–1600, Dec. 2023.

[17] P. F. Moshiri, M. Simsek, and B. Kantarci, “Joint optimization of completion ratio and latency of offloaded tasks with multiple priority levels in 5G edge,” IEEE Trans. Netw. Serv. Manag., vol. 22, pp. 1357–1371, Jan. 2025.

[18] A. Abba Ari, F. Samafou, A. Ndam Njoya, A. C. Djedouboum, M. Aboubakar, and A. Mohamadou, “IoT-5G and B5G/6G resource allocation and network slicing orchestration using learning algorithms,” IET Netw., vol. 14, pp. e70002, Jan. 2025.

[19] P. Li, Y. Wang, Z. Wang, T. Wang, and J. Cheng, “Joint task offloading and resource allocation strategy for hybrid MEC-enabled LEO satellite networks: A hierarchical game approach,” IEEE Trans. Commun., vol. 73, pp. 3150–3166, Oct. 2024.

[20] N. Lassoued and N. Boujnah, “A comprehensive review of energy efficiency in 5G networks: Past strategies, present advances, and future research directions,” Computers, vol. 15, pp. 50, Oct. 2026.

[21] R. Zhang, L. Wu, S. Cao, D. Wu, and J. Li, “A vehicular task offloading method with eliminating redundant tasks in 5g HetNets,” IEEE Trans. Netw. Serv. Manag., vol. 20, pp. 456–470, Aug. 2022.

[22] A. Uddin, A. H. Sakr, and N. Zhang, “Adaptive prioritization and task offloading in vehicular edge computing through deep reinforcement learning,” IEEE Trans. Veh. Technol., vol. 74, pp. 5038–5052, Nov. 2024.

[23] F. Bahramisirat, M. A. Gregory, and S. Li, “Dynamic network slicing for resource allocation in 5G/B5G networks: An optimization-based approach,” Ad Hoc Netw., p. 104194, Feb. 2026.

[24] H. Hu, Z. Chen, F. Zhou, Z. Han, and H. Zhu, “Joint resource and trajectory optimization for heterogeneous-UAVs enabled aerial-ground cooperative computing networks,” IEEE Trans. Veh. Technol., vol. 72, pp. 8812–8826, Feb. 2023.

[25] H. Zhai, X. Zhou, H. Zhang, and D. Yuan, “Delay minimization in hybrid edge computing networks: A DDQN-based task offloading approach,” IEEE Trans. Veh. Technol., vol. 73, no. 10, pp. 15098–15108, May 2024.

[26] M. K. Benbraika, O. Kraa, Y. Himeur, K. Telli, S. Atalla, and W. Mansoor, “Enhancing 5g vehicular edge computing efficiency with the Hungarian algorithm for optimal task offloading,” Computers, vol. 13, pp. 279, Oct. 2024.

[27] A. Alam, P. Shah, R. Trestian, K. Ali, and G. Mapp, “Energy efficiency optimisation of joint computational task offloading and resource allocation using particle swarm optimisation approach in vehicular edge networks,” Sensors, vol. 24, pp. 3001, May 2024.

[28] A. Younis, S. Maheshwari, and D. Pompili, “Energy-latency computation offloading and approximate computing in mobile-edge computing

networks," IEEE Trans. Netw. Serv. Manag., vol. 21, pp. 3401–3415, Jan. 2024.

[29] K. Luo, Y. Wang, Y. Liu, and K. Zhu, "Collaborative integration of vehicle and roadside infrastructure sensor for temporal dependency-aware task offloading in the Internet of Vehicles," Int. J. Intell. Syst., vol. 2025, p. 8064086, May 2025.

[30] W. Yang, Z. Liu, X. Liu, and Y. Ma, "Deep reinforcement learning-based low-latency task offloading for mobile-edge computing networks," Appl. Soft Comput., vol. 166, p. 112164, Nov 2024.

[31] S. Mathi, G. Rohan Lal, L. C., Madala, K. A., Reddy, P. Jagadhabhiram, and G. Neelakanta Iyer, "FedBHAD: Energy-Efficient Federated Learning for Black Hole Attack Detection in RPL-Based Low-Power IoT Networks," Emerging Science Journal, vol. 9, pp.2884–2898, Dec 2025.

[32] Y. Xiao, and Y. Dong, "Optimizing AIGC Technology for IoT Devices with Deep Learning," HighTech and Innovation Journal, vol. 6, pp. 976–990, Sep 2025.

[33] H. Vidyaningtyas, H., Iskandar, A. A. Pramudita, "NOMA Performance Improvement with Downlink Sectorization," Emerging Science Journal, vol. 9, pp. 311-328, Feb 2025.

[34] A. Ateya, A. Muthanna, A. Koucheryavy, Y. Maleh, and A. A. A. El-Latif, "Energy efficient offloading scheme for MEC-based augmented reality system," Clust. Comput., vol. 26, pp. 789–806, Feb 2023.

[35] V. R. Chintapalli, R. Partani, and B. R. Tamma, "Energy efficient and delay aware deployment of parallelized service function chains in NFV-based networks," Comput. Netw., vol. 243, p. 110289, Apr. 2024.

[36] H. Yuan, T. Wang, M. Fu, and Y. Shi, "GIRP: Energy-efficient QoS-oriented microservice resource provisioning via multi-objective multi-task reinforcement learning," IEEE Trans. Mob. Comput., vol. 24, pp. 5793–5807, Mar. 2025.

[37] G. Wu, X. Chen, Y. Shen, Z. Xu, H. Zhang, S. Shen, and S. Yu, "Combining Lyapunov optimization with actor–critic networks for privacy-aware IIoT computation offloading," IEEE Internet Things J., vol. 11, pp. 17437–17452, Jan. 2024.

[38] M. An, T. Lu, X. Han, and Z. Ding, "An online distributed optimisation model solving the time-coupled conundrum by the Lyapunov drift plus penalty method," IET Gener. Transm. Distrib., vol. 19, p. e70019, Jan 2025.