BEST: Blockchain-Enabled Sustainable Task Scheduling for Optimizing Large AI Model Workloads at the Edge

Haoxiang Luo, Runhua Chen, Gang Sun, *Senior Member, IEEE*,
Hongfang Yu, *Senior Member, IEEE*, Schahram Dustdar, *Fellow, IEEE*

***Abstract*—The proliferation of Large AI Models (LAMs) to the network edge promises unprecedented intelligence in real-time applications. It also introduces a critical sustainability challenge due to the immense energy footprint of model inference on resource-constrained devices. While model-level optimizations have made progress, they fail to address the systemic issue of aggregate energy consumption across a network. This paper introduces BEST (Blockchain-Enabled Sustainable Task Scheduling), a novel framework that reimagines edge resource management to explicitly tackle this sustainability crisis. BEST establishes a decentralized marketplace for underutilized edge resources by introducing a novel application of Real-World Asset (RWA) tokenization to abstract and commoditize computational capacity and battery power. These tokenized resources can be securely and transparently traded among peer devices via a blockchain-based infrastructure. To navigate this dynamic marketplace, we develop a cooperative multi-agent deep reinforcement learning (MADRL) algorithm based on Proximal Policy Optimization (PPO). The agents learn a sophisticated scheduling policy that dynamically balances task latency, system-wide energy consumption, and task success rate. Extensive simulations against compared baselines demonstrate BEST's unique strengths. It reduces total energy consumption by up to $26\% - 46\%$ against other schemes, while maintaining a high task success rate and latency competitive with the state-of-the-art scheme. By creating a market-driven, collaborative ecosystem, BEST provides a scalable and effective pathway toward achieving truly sustainable LAM at the edge.**

***Impact Statement*—This research provides a market-driven paradigm for achieving sustainable LAMs at the network edge, a critical step as AI's energy footprint becomes a global concern. By creating a decentralized micro-economy for computational and energy resources, our work not only addresses the pressing sustainability challenge but also introduces novel economic incentives for participation in edge networks. This can unlock a new generation of complex, long-running AI applications in power-constrained environments such as autonomous vehicle fleets, industrial IoT, and remote sensing networks. The BEST framework moves beyond insufficient device-centric optimizations to a systemic, collaborative solution. It directly confronts the critical barrier of aggregate energy consumption that limits the scalability and long-term viability of intelligent edge systems, paving the way for a more resilient and efficient decentralized digital infrastructure.**

***Index Terms*—Sustainable Large AI Models (LAMs), edge Computing, blockchain, Real-World Asset (RWA), edge computing power network.**

I. Introduction

A. Background

THE field of Artificial Intelligence (AI) has been revolutionized by the advent of Large AI Models (LAMs), which have demonstrated remarkable capabilities across a spectrum of complex tasks [1]. These models, with their billions of parameters, have demonstrated unprecedented capabilities in natural language understanding, content generation, and complex problem-solving [2], [3]. Initially confined to powerful, centralized cloud data centers, there is a strong and growing imperative to push these LAMs to the network edge, closer to where data is generated and consumed [4], [5]. The motivations for this shift are compelling. It is significantly reducing latency for real-time applications, enhancing data privacy by keeping sensitive information local, and reducing reliance on constant, high-bandwidth cloud connectivity [6].

However, the environmental and operational cost of AI is a well-documented and escalating concern [7]. While the substantial energy required for model training has garnered significant attention, with models like GPT-3 consuming an estimated $1,287$ megawatt-hours (MWh) for a single training run [8], a more pervasive challenge is emerging. Recent analyses reveal that the inference phase, where a trained model is used to make predictions, is the dominant contributor to the model's lifecycle energy footprint, accounting for as much as $80-90\%$[1] of its total energy use [9]. This is because training is a one-time or infrequent expense, whereas inference occurs continuously and on a massive scale throughout a model's operational lifecycle [10]. A single query to a generative AI model can consume five to ten times more electricity than a traditional web search, and projections indicate that by 2027,

This work was supported in part by the National Science and Technology Major Project (2025ZD1302700) and in part by the National Natural Science Foundation of China (62394324).

H. Luo, G. Sun, and H. Yu are with the School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China (e-mail: lhx991115@163.com; {gangsun, yuhf}@uestc.edu.cn).

R. Chen is with the Glasgow College, University of Electronic Scienceand Technology of China, Chengdu 611731, China. (e-mail: 2839936c@student.gla.ac.uk).

S. Dustdar is with ICREA, Barcelona 08002, Spain (e-mail: schahram.dustdar@upf.edu).

The corresponding author: Gang Sun.

[1] https://www.hpcwire.com/2019/03/19/aws-upgrades-its-gpu-backed-ai-inference-platform/

the electricity consumption of the AI sector could rival that of entire countries[2].

This reality presents a fundamental paradox for the future of edge LAMs. Edge devices are inherently constrained in processing power, memory, and most critically, battery life. Deploying energy-intensive LAMs onto these platforms creates a direct and unsustainable conflict between the functional requirement for advanced intelligence and the operational constraint of limited energy [11]. This is not merely a technical inconvenience. It is a critical barrier that threatens the scalability, economic viability, and environmental responsibility of the next generation of intelligent edge applications [12].

B. Research Motivation

Current efforts to mitigate the energy consumption of edge LAM primarily focus on model-level optimizations. Techniques such as pruning [13], which removes redundant model parameters; quantization [14], which reduces the numerical precision of calculations; and knowledge distillation [15], which trains smaller models to mimic larger ones, have proven effective at reducing the computational load of a single inference task. These methods are essential first steps. However, they are often insufficient on their own [16]. While these techniques make individual LAM operations more energy-efficient, the concurrent explosion in the deployment and complexity of AI models can lead to a net increase in aggregate energy demand across the ecosystem. This suggests that optimizing the model in isolation is not enough [17]. A system-level solution that manages the collective resources of the entire network is required.

The key observation motivating our work is that a network of edge devices, while individually constrained, collectively represents a massive, albeit fragmented and underutilized, pool of computational and energy resources. At any given moment, many devices are idle or operating well below their maximum capacity [18]. The challenge lies in creating a mechanism to aggregate and share these distributed resources in a trusted, efficient, and scalable manner.

This is where blockchain technology and the concept of a decentralized marketplace offer a promising solution [19]. Blockchain provides a foundation for secure, tamper-proof, and decentralized coordination among peers, eliminating the need for a central authority [17]. Building on this, smart contracts can be used to automate the logic of a marketplace [20], matching resource suppliers with demanders through transparent and verifiable rules. The most profound innovation, however, comes from extending the concept of Real-World Asset (RWA) tokenization to this domain [21]. While RWA tokenization is typically associated with tangible assets like real estate or financial instruments [22], we propose a novel application, the tokenization of abstract, ephemeral resources. By representing standardized units of computational work (e.g., Giga-FLOPS) and stored energy (e.g., Watt-hours) as fungible, tradable digital tokens on a blockchain, we can create a liquid and dynamic micro-economy for edge resources. This abstraction transforms a heterogeneous collection of physical devices into a homogeneous pool of tradable assets, creating a programmable and economically-driven substrate for sustainable, distributed computation.

C. Our Contributions

To address the aforementioned challenges, this paper introduces **BEST**, a **B**lockchain-**E**nabled **S**ustainable **T**ask scheduling framework for LAM workloads at the edge. BEST integrates a decentralized resource marketplace with an intelligent, learning-based scheduler to create a collaborative ecosystem that explicitly prioritizes sustainability. To the best of our knowledge, this is the first study to tokenize device resources into RWA, enabling the sustainable operation of edge LAMs. The primary contributions of this work are fourfold:

- **A Novel System-Level Framework for Sustainable Edge AI:** We propose BEST, a holistic architecture that moves beyond device-centric model optimization to address the systemic challenge of aggregate energy consumption in a distributed edge network.
- **Tokenization of Edge Compute and Power as Real-World Assets:** We introduce a novel RWA tokenization model that abstracts and commoditizes the ephemeral computational and energy resources of heterogeneous edge devices. This creates a transparent, liquid, and decentralized marketplace, enabling a fluid exchange of resources among peers.
- **A Multi-Agent DRL Approach for Collaborative Task Scheduling:** We formulate the complex, multi-objective scheduling problem as a Decentralized Partially Observable Markov Decision Process (Dec-POMDP). We then propose a cooperative multi-agent deep reinforcement learning (MADRL) algorithm, based on Proximal Policy Optimization (PPO), which learns an optimal policy to dynamically balance task latency, energy consumption, and overall system utility.
- **Comprehensive Performance Evaluation:** We conduct extensive, reproducible simulations to validate the efficacy of the BEST framework. Our results demonstrate its significant superiority over three distinct baseline methods across four key performances, confirming its potential to enable sustainable LAM deployment at scale.

D. Paper Structure

The remainder of this paper is organized as follows. Section II reviews related work in edge LAM optimization, sustainable AI, and decentralized resource management. Section III presents the detailed system architecture of the BEST framework and formulates the multi-objective optimization problem. Section IV describes the DRL-based scheduling methodology, including the MDP formulation and the MAPPO algorithm. We conduct a security analysis of the BEST framework in Section V. Then, Section VI details the simulation setup, performance metrics, and presents a thorough analysis of the results. Finally, Section VII concludes the paper and discusses avenues for future research.

[2] https://www.polytechnique-insights.com/en/columns/energy/generative-ai-energy-consumption-soars/

This article has been accepted for publication in IEEE Transactions on Artificial Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAI.2026.3669544

II. Related Works

This section contextualizes our work by reviewing three key areas of research: the optimization of LAMs for edge devices, methodologies for green and sustainable AI, and the use of blockchain for decentralized resource management. We analyze the state of the art in each area to identify the research gap that the BEST framework is designed to fill.

A. Optimizing Large Model Inference on Edge Devices

The primary challenge in deploying LAMs on edge devices stems from the mismatch between the models' significant resource requirements and the devices' limited capabilities [23], [24]. Consequently, a substantial body of research has focused on model-centric optimization techniques to bridge this gap. These techniques can be broadly categorized as follows.

- Model Compression: It is the most common approach and includes two main strategies. Pruning or inducing sparsity involves identifying and removing redundant or less important parameters [13], [25], namely weights or neurons, from a neural network. Thus, inference can reduce its size and the number of computations required.
- Quantization: It aims to reduce the numerical precision of the model's weights and activations [14], for example, by converting 32-bit floating-point numbers to 8-bit integers. This not only shrinks the model's memory footprint but also allows for faster computations on hardware that supports lower-precision arithmetic.
- Knowledge Distillation: This technique involves a teacher-student paradigm. A large, complex, and highly accurate teacher model is used to train a much smaller and computationally cheaper student model. The student learns to mimic the output distribution or intermediate representations of the teacher, thereby inheriting its knowledge while remaining lightweight enough for edge deployment [15].
- Efficient Architecture Design: It involves designing neural network architectures from the ground up with efficiency in mind. Architectures like MobileNet and EfficientNet [26], utilize building blocks such as depthwise separable convolutions to achieve a favorable balance between accuracy and computational cost, measured in FLOPS.

While these techniques are foundational and highly effective, they are fundamentally device-centric. They optimize a model for execution on a single, isolated device. This approach, however, does not leverage the collective power of a network of devices. Our work is complementary to these methods. BEST operates at a higher, system-level orchestration layer, making intelligent decisions about where and how to execute a task. It transforms the problem from "*how can one device run this task?*" to "*how can the network of devices collectively run this task sustainably*?".

B. Green and Sustainable AI Methods

As awareness of AI's environmental impact has grown, so has the field of Green AI or Sustainable AI [27]. The core principle of Green AI is to incorporate efficiency and environmental impact as key metrics alongside traditional ones like accuracy. Current strategies in this domain largely focus on centralized, data-center-based AI and include:

- Hardware and Data Center Optimization: This involves using more energy-efficient hardware, such as Google's Tensor Processing Units (TPUs) or specialized GPUs with higher FLOPS-per-watt ratings [28]. It also includes optimizing data center infrastructure through advanced cooling systems, efficient power delivery, and strategic load balancing [29].
- Carbon-Aware Computing: This strategy involves scheduling computationally intensive tasks, such as model training, at times or in geographical locations where the electricity grid is supplied by a higher proportion of renewable energy sources. This reduces the operational carbon footprint of the computation, even if the energy consumption remains the same [30].

The work on Green AI provides a foundation but exhibits a data center-centric blind spot. The principles designed for large, monolithic, and professional data centers do not directly translate to the network edge. Sustainability at the edge is not about optimizing a single facility but about coordinating a dynamic swarm of independent devices [5], [7]. This represents a distinct and largely unexplored research area, which BEST directly addresses the sustainability challenges of a decentralized computing fabric.

C. Decentralized Resource Management in Edge Computing

The need for coordination in distributed edge networks has led researchers to explore decentralized technologies, with blockchain being a prominent candidate. Several works have proposed using blockchain to facilitate resource sharing and create decentralized marketplaces in edge computing.

- Blockchain for Secure Data and Resource Exchange: Researchers have proposed using blockchain to maintain secure logs of data sharing, verify device identities, and ensure the integrity of transactions in computing power networks [31], and other scenarios where network resource sharing.
- Tokenized Edge Marketplaces: The concept of creating a marketplace to match suppliers of edge infrastructure with demanders has been explored [32]. In these models, smart contracts often automate an auction process where tenants can bid for resources from providers.

These works validate the foundational premise of BEST, using a blockchain to enable a decentralized marketplace. However, they often face two key limitations. First, the resource allocation mechanisms are typically based on simple heuristics or classical auction models, e.g., lowest-bid-wins [33]. These may not be optimal for scheduling complex, multi-faceted LAM workloads with strict latency and energy constraints. Second, their primary optimization goal is often resource monetization or latency reduction, rather than an explicit focus on system-wide energy sustainability.

Furthermore, a critical challenge is the inherent latency and overhead of consensus [34], which can be prohibitive

TABLE I: Key Notations

Notations	Definitions
N	The total number of edge devices in the network
$\mathcal{E}$	The set of all edge devices, $\mathcal{E} = \{e_1, \ldots, e_N\}$
e_i	An individual edge device, where $i \in \{1, \ldots, N\}$
C_i	The maximum computational capacity of device e_i (in FLOPS)
B_i	The current battery level of device e_i (in Joules)
t_j	An individual LAM inference task.
W_j	The computational workload of task t_j (in FLOPs)
D_j	The data size associated with task t_j (in bytes)
L_j^{max}	The maximum tolerable latency (deadline) for task t_j
L_j	The actual end-to-end latency for completing task t_j
E_j	The total energy consumed for processing task t_j
π	The scheduling policy learned by the DRL agents
w_l, w_e	Weighting coefficients for latency and energy in the objective function
γ	The discount factor for future rewards in the DRL formulation
CTK	Compute Token, representing a unit of computational resource
PTK	Power Token, representing a unit of energy

for real-time, resource-constrained edge devices [19]. BEST addresses this by integrating a learning-based DRL agent as the core decision-maker and adopting a hybrid architecture. High-frequency scheduling decisions are made off-chain by the fast DRL agents, while the blockchain is reserved for lower-frequency, high-value operations like identity management, token ownership, and final transaction settlement. This pragmatic design makes the framework both intelligent and feasible.

III. The BEST Framework and Problem Formulation

This section details the architecture of the BEST framework, its core components, the novel mechanism for resource tokenization, and the formal mathematical formulation of the optimization problem. For clarity, Table I summarizes the key notations used throughout this paper.

A. System Model and Components

The BEST framework operates within a distributed environment consisting of a set of heterogeneous Edge Devices (EDs), a shared blockchain network, and a stream of incoming LAM inference tasks. The key components, illustrated in Fig. 1, are as follows:

We consider a set of N heterogeneous EDs, denoted by $\mathcal{E} = \{e_1, e_2, \ldots, e_N\}$. Each device $e_i \in \mathcal{E}$ is characterized by its computational capacity C_i (measured in Floating Point Operations Per Second, FLOPS), its current battery level B_i (in Joules), and its network connection quality. EDs are autonomous entities that can act as both *Task Requesters* (when they have a LAM workload to execute) and *Resource Suppliers* (when they have idle capacity).

Tasks arrive at the EDs according to a stochastic process [35]. A task t_j is defined by a tuple (W_j, D_j, L_j^{max}), where W_j is the computational workload (in total FLOPs), D_j is the size of the input/output data (in bytes), and L_j^{max} is the maximum tolerable latency, or deadline.

Additionally, a decentralized, permissionless ledger serves as the trust anchor for the entire system. It hosts the smart contracts that govern the resource marketplace and maintains an immutable record of all transactions, e.g., token transfers, task agreements [36]. To be suitable for an edge environment, this is assumed to be a high-throughput, low-fee blockchain.

Then, a suite of smart contracts deployed on the blockchain that codify the logic of the marketplace. This includes contracts for token minting/burning, managing the order book for resource auctions, and settling payments upon successful task completion.

Finally, a lightweight DRL agent runs on each ED. This agent is the brain of the device, responsible for making real-time scheduling decisions for its tasks based on its local observations and the state of the marketplace [37].

B. RWA Tokenization of Edge Compute and Power Resources

A core innovation of BEST is the abstraction of heterogeneous physical resources into standardized, tradable digital assets. This is achieved through a novel application of Real-World Asset (RWA) tokenization, following a structured process [38].

- **Asset Abstraction and Specification:** We define two fundamental, fungible assets that underpin LAM execution: **1) Computational Capacity:** Represented by **Compute Tokens (CTK)**. One CTK corresponds to a standardized unit of computational work, for example, 10^9 floating-point operations (1 GFLOP); **2) Energy:** Represented by **Power Tokens (PTK)**. One PTK corresponds to a standardized unit of energy consumption, for example, 3600 Joules (1 Watt-hour). These tokens are implemented as standard fungible tokens, e.g., ERC-20, or others[3]. This standardization is crucial, which allows the system to treat 1 GFLOP of computation from a powerful smartphone CPU as equivalent to 1 GFLOP from a low-power edge accelerator, creating a level playing field for trade.
- **Off-Chain Verification via Oracles:** The value of these tokens depends on them being verifiably backed by real, available resources. To prevent fraud, e.g., a device minting tokens for resources it does not have, we propose a secure off-chain verification mechanism. Each ED is assumed to have a trusted monitoring component, ideally running within a hardware-based Trusted Execution Environment (TEE) [39]. This component securely measures the device's idle CPU cycles and available battery energy. This verified data is then relayed to the blockchain via a decentralized oracle network. This process serves as a Proof of Resource (PoR), analogous to the Proof of Asset (PoA) used in DeFi to verify the backing of asset-backed stablecoins [40], but applied for the first time to ephemeral computational resources. Specifically, the interaction process between PoR and TEE is as follows:

[3] https://ethereum.org/en/developers/docs/standards/tokens/erc-20/

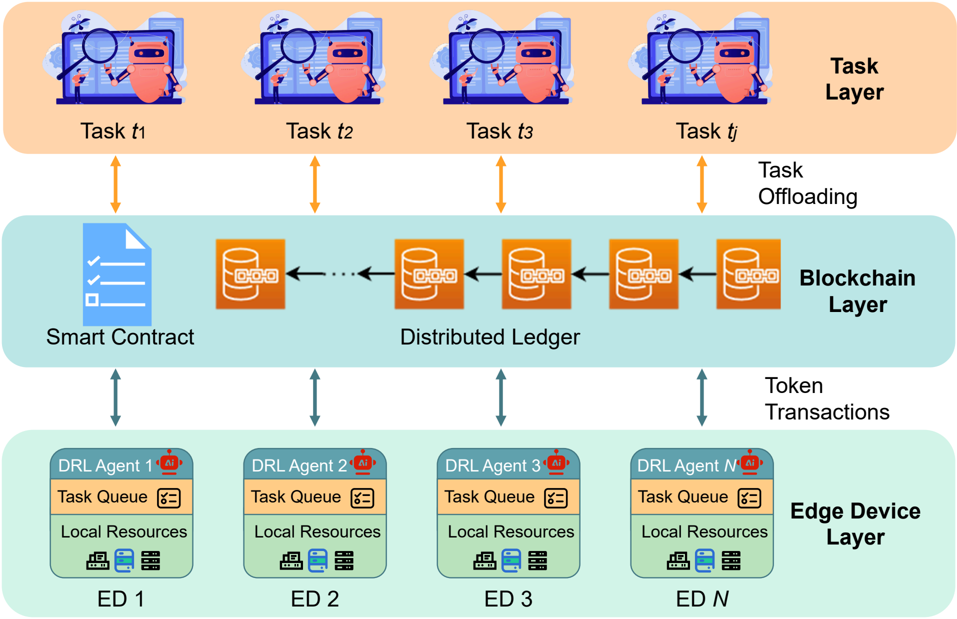


Fig. 1: System model and components. The task layer completes token transactions and identity verification through the blockchain layer. The DRL agent makes decisions on whether to execute the task locally or to offload it across devices based on the local resource status and market information.

1) Resource Data Collection: TEE-integrated monitoring modules collect idle CPU cycles and remaining battery energy, excluding system process overhead.

2) Data Encryption and Signing: Collected data is encrypted within TEE, signed with the device's TEE endorsement key to ensure integrity.

3) Decentralized Oracle Reporting: Encrypted data is transmitted to a decentralized oracle network, with multi-oracle aggregation to prevent single-point tampering.

4) On-Chain Validation: Smart contracts verify TEE signatures and oracle consensus results, enabling token minting only if resources are verifiably available.

- **Issuance (Minting) and Burning:** An ED can signal its intent to become a supplier by interacting with the PoR and smart contract. Based on the data received from the oracle, the contract allows the ED to mint a corresponding number of CTK and PTK, which are credited to its wallet. These tokens now represent a verifiable claim on the device's idle resources. Conversely, when a requester spends tokens to have a task executed, those tokens are transferred to the supplier and subsequently burned, namely removed from circulation, by the contract to signify that the underlying resource has been consumed.

This tokenization process creates a vital layer of abstraction. It decouples the physical hardware from the service it provides, transforming a complex resource allocation problem into a more straightforward economic transaction problem. It can be efficiently managed by autonomous DRL agents.

C. Blockchain-Powered Decentralized Scheduling Marketplace

The tokenized resources are traded on a decentralized marketplace governed by a set of smart contracts. The marketplace functions as an automated exchange, facilitating resource allocation without a central coordinator.

The core of the marketplace is an on-chain order book.

- **Suppliers (Sellers):** An ED with surplus minted tokens (CTK and PTK) can place *ask* orders on the marketplace, specifying the quantity of tokens they are willing to sell and at what price, such as.
- **Requesters (Buyers):** An ED that needs to offload a task determines the required amount of CTK and PTK. Its DRL agent can then either place a *bid* order to buy these tokens at a certain price or execute a *market buy* to purchase them at the lowest available *ask* price.

Meanwhile, the smart contract of the marketplace acts as a trustless intermediary. When a deal is matched, the requester's payment and the supplier's resource tokens can be locked in escrow. Upon successful completion of the offloaded task, which can be verified, for instance, by the requester signing a confirmation message, the contract releases the payment to the supplier and finalizes the token transfer. This automated, smart contract-driven process, inspired by proposed edge auction systems, ensures fairness and security for all participants [20]. Moreover, at the beginning of the system, each device is initially allocated 100 CTK and 50 PTK as the start-up funds. Additionally, the smart contract includes a micro-credit pool. That is to say, new devices can pledge their device ID hash to borrow up to 30 PTK. If the repayment is overdue, their minting rights will be frozen until the debt is repaid.

When the task is executed locally, only CTK needs to be paid to record the CPU usage. There is no need to pay PTK. Since the energy cost will be internalized within the reward function as described below, this not only avoids unnecessary expenses from self-transactions but also strengthens the directionality of economic incentives. That is, PTK is specifically used to compensate for the external energy transfer in the offloading scenario, thereby suppressing meaningless offloading and ensuring that market liquidity serves the real scarcity of resources. Specifically, if the task is offloaded to device ei, both CTK and PTK need to be paid to calculate and compensate for battery consumption.

D. Multi-Objective Optimization Problem Formulation

The overarching goal of the BEST framework is to find a scheduling policy π that maps the system state to a set of actions for all incoming tasks, that is, local execution or offloading. The policy should optimize for multiple, often conflicting, objectives. We formulate this as the minimization of a weighted cost function, aggregated over all tasks t_j processed by the system.

$$\begin{aligned} \pi^* = \arg\min_{\pi} \quad & \mathbb{E}\left[\sum_{j=1}^{M}(w_l \cdot L_j + w_e \cdot E_j)\right] \\ \text{s.t.} \quad & (C1): L_j \leq L_j^{\max}, \quad \forall j \\ & (C2): C_i^{\text{committed}}(t) \leq C_i^{\text{avail}}(t), \quad \forall i, t \\ & (C3): B_i(t) \geq B_i^{\min}, \quad \forall i, t \end{aligned} \tag{1}$$

where M is the total number of tasks; L_j denotes the end-to-end latency for task t_j, defined as the time from its arrival to the completion of its execution; w_l and w_e are non-negative weighting coefficients that represent the relative importance of latency and energy sustainability, respectively. These can be tuned to reflect system-wide priorities, for instance, a higher w_e for a green mode. E_j represents the total energy consumed for processing task t_j. This includes energy for local computation, communication energy for offloading data, and the energy represented by any purchased PTK. The total energy E_j includes local execution and execution on e_j two scenarios, which are composed as follows:

$$E_j = \begin{cases} E_j^{\text{local}} = P_i^{\text{active}} \cdot \frac{W_j}{C_i}, \\ E_j^{\text{offload}} = E_{ij}^{\text{tx}} + E_{ij}^{\text{rx}} + P_j^{\text{active}} \cdot \frac{W_j}{C_j} + E_j^{\text{PTK}}, \end{cases} \tag{2}$$

where P_i^{active} and P_i^{idle} are the active and idle power consumption of device e_i (in Watts); $E_{ij}^{\text{tx}} = P_{ij}^{\text{tx}} \cdot \frac{D_j}{R_{ij}}$ is the transmission energy from e_i to e_j, with P_{ij}^{tx} being the transmit power and R_{ij} the link data rate. $E_{ij}^{\text{rx}} = P_{ij}^{\text{rx}} \cdot \frac{D_j}{R_{ij}}$ is the receive energy. $E_j^{\text{PTK}} = \eta \cdot \text{PTK}_j$ is the monetary energy cost converted via PTK, where η is the PTK-to-Joules exchange rate.

This optimization is subject to the following constraints for each task t_j and device e_i:

1) **Deadline Constraint:** $L_j \leq L_j^{max}$. The task must be completed before its deadline.
2) **Resource Constraint:** The resources committed $C_i^{\text{committed}}(t)$ by any device e_i at any time cannot exceed its available local and purchased resources $C_i^{\text{avail}}(t)$.
3) **Battery Survival Constraint:** The battery level of any device e_i, $B_i(t)$, must remain above a minimum threshold, B_i^{min}, to ensure its continued operation.

This formulation captures the fundamental trade-offs in the system. Offloading a task might reduce its latency but increase energy consumption due to network communication, while processing locally might save communication energy but risk missing a deadline if the device is slow. The role of the DRL agent is to learn a policy that navigates these trade-offs intelligently.

IV. DRL-based Sustainable Task Scheduling

To solve the complex, dynamic, and multi-objective optimization problem formulated in the previous section, we turn to deep reinforcement learning. Specifically, we model the system as a cooperative multi-agent problem and employ an MADRL algorithm to learn an effective scheduling policy [41].

A. Decentralized Markov Decision Process Formulation

The problem is naturally modeled as a Dec-POMDP [42], as each edge device acts as an autonomous agent with only a local view of the overall system state. The goal is for these agents to learn to cooperate implicitly to optimize a global objective function. The Dec-POMDP is defined by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \mathcal{O}, N, \gamma)$.

- **State Space ($\mathcal{S}$):** The global state $s \in \mathcal{S}$ includes the complete information about all devices and tasks in the network. However, each agent i only perceives a local observation $o_i \in \mathcal{O}$. The observation vector o_i for agent e_i at a given timestep includes:
  1) Local Task Information: A feature vector describing the tasks in its local queue, e.g., (`workload`, `data_size`, `deadline`) of the task at the head of the queue.
  2) Local Resource Status: Its current available CPU capacity C_i^{avail} and battery level B_i.
  3) Market Information: A summary of the state of the resource marketplace, such as the current lowest *ask* prices for CTK and PTK, and the volume of available tokens.
  4) Token Holdings: The agent's current balance of CTK and PTK.
  The global state s exists completely in the environment and contains the precise resource values of all devices. However, during training, the centralized critic cannot directly access s. Instead, the critic's input is the concatenated vector of all o_i.
- **Action Space ($\mathcal{A}_i$):** For the task at the head of its queue, each agent i selects an action a_i from its discrete action space. The action space for agent i is $\mathcal{A}_i = \{0\} \cup \{j | e_j \in \mathcal{E}, j \neq i\}$, where $a_i = 0$ denotes execute locally, that is, the agent processes the task using its own resources; and $a_i = j$ represents offload to oeer j, that is, the agent initiates a transaction on the marketplace to purchase the necessary CTK and PTK, and then offloads the task data to device e_j for execution. Moreover, to address resource competition among multiple agents, a hierarchical conflict resolution strategy is integrated:

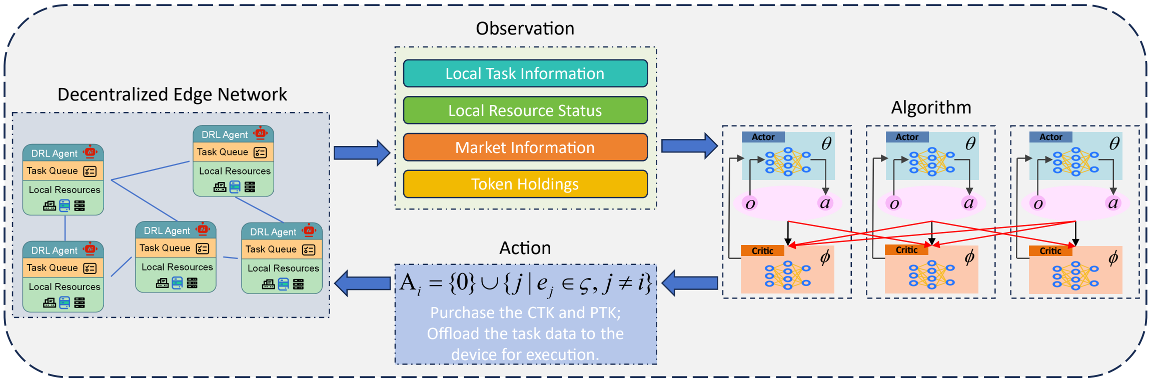


Fig. 2: MAPPO framework.

1) Priority Precedence: Tasks are prioritized by deadline urgency L_j^{max}. High-priority task agents gain exclusive resource access first.
2) Resource Splitting: For splittable LAM inference tasks, resources are allocated proportionally to agents' task workload demands, minimizing idle capacity.
3) Dynamic Bargaining: Conflicting agents negotiate off-chain to adjust bids. The agent offering a higher price and meeting priority requirements secures resources.
Furthermore, if the computing capacity of the edge devices is limited, when they are tasked with indivisible operations, there may be physical infeasibility leading to ineffective training. Therefore, we design a binary mask $M_i \subset A_i$ for each decision to filter out inactionable actions. Specifically, if $W_j/C_i > L_j^{max} - L_{margin}$, where L_{margin} is the reserved time for communication, then $a_i = 0$ is masked. Conversely, if the C_j^{avail} of the target device e_j is less than $W_j/(L_j^{max} - L_{margin})$, then $a_i = j$ is masked.

- **Transition Probability ($\mathcal{P}$):** The transition function $P(s'|s, \mathbf{a})$ defines the probability of transitioning from global state s to s' after the joint action $\mathbf{a} = (a_1, \ldots, a_N)$ is taken. This function is governed by the system dynamics, such as task completion times, network delays, and new task arrivals, and is unknown to the agents.
- **Reward Function ($\mathcal{R}_i$):** The reward function is crucial for guiding the learning process. Each agent receives a local reward based on the outcome of its action, which is designed to align its local incentive with the global optimization objective. The reward for agent i after completing a task is:

$$R_i(t) = -\left(w_l \cdot L_i(t) + w_e \cdot E_i(t)\right) \\ + \beta_{\text{succ}} \cdot \mathbb{I}(\text{success}) - \beta_{\text{fail}} \cdot \mathbb{I}(\text{failure}), \tag{3}$$

  where $L_i(t)$ and $E_i(t)$ are the latency and energy cost associated with the completed task, $\mathbb{I}(\cdot)$ is an indicator function, and β_{succ} and β_{fail} are large positive and negative constants rewarding successful completion before the deadline and penalizing failure, respectively. The energy cost $E_i(t)$ includes both computation and communication energy.
- **Discount Factor (γ):** A discount factor $\gamma \in [0, 1)$ can balance immediate and future rewards.

B. Multi-Agent Proximal Policy Optimization (MAPPO)

The DRL agent selection first generates an unsigned resource reservation request off-chain, which includes the task hash, target device j, the quantity, and time stamp t of the offered CTK/PTK. This request is broadcast to device j through the gossip protocol. After the device j locally verifies the resource availability, it signs the off-chain commitment and returns it to device i. At this point, the task can immediately start transmission without waiting for the chain-based confirmation. Subsequently, at the time $t + \Delta t$, both parties jointly submit the commitment to the smart contract for settlement. If the off-chain execution is successful, the contract releases the tokens. Otherwise, the contract arbitrates based on the encrypted proof in the commitment. This design decouples task execution from the chain-based settlement, ensuring that delay-sensitive tasks are not affected by the consensus delay.

Furthermore, to address the multi-agent credit assignment and non-stationarity challenges, we employ a policy gradient method, MAPPO, as shown in Fig. 2, which is a variant of PPO adapted for cooperative multi-agent settings. We employ the widely-used Centralized Training with Decentralized Execution (CTDE) paradigm [43]. This approach utilizes a centralized critic during the offline training phase to stabilize learning, while enabling the fully decentralized execution of the learned policies on edge devices.

- **Decentralized Actors:** Each agent i has an actor network, $\pi_{\theta_i}(a_i|o_i)$, parameterized by θ_i. The actor takes the agent's local observation o_i as input and outputs a policy, which is a probability distribution over its action space $\mathcal{A}_i$.
- **Centralized Critic:** During training, a single, centralized critic network, $V_\phi(s)$, parameterized by ϕ, is used. The critic takes the global state s as input and outputs an estimate of the expected cumulative reward from that state. By having access to the global state, the critic provides a stable and consistent learning signal for all actors, mitigating the non-stationarity that arises when multiple agents are learning simultaneously [43].

Moreover, We use MAPPO to optimize the actor networks by maximizing a clipped surrogate objective function. This

Algorithm 1: Centralized Training of BEST Agents

```
Initialization:
1. Initialize actor networks $\pi_{\theta_i}$ and critic network $V_\phi$
   for all agents $i = 1, ..., N$
2. Initialize replay buffer $\mathcal{B}$
for episode = 1 to $K_{episodes}$ do
    Episode Initialization:
    1. Reset simulation environment and get initial
       global state $s_0$
    for $t = 1$ to $T_{steps}$ do
        Decentralized Action Selection:
        for each agent $i = 1, ..., N$ do
            1. Get local observation $o_{t,i}$ from $s_t$
            2. Select action $a_{t,i} \sim \pi_{\theta_i}(a|o_{t,i})$
        Environment Interaction:
        1. Execute joint action $\mathbf{a}_t = (a_{t,1}, ..., a_{t,N})$ in
           the environment
        2. Observe next global state $s_{t+1}$, and global
           reward $r_t$
        3. Store transition $(s_t, \mathbf{o}_t, \mathbf{a}_t, r_t, s_{t+1})$ in buffer
           $\mathcal{B}$
        4. $s_t \leftarrow s_{t+1}$
    Network Update:
    for epoch = 1 to $E_{epochs}$ do
        1. Sample a batch of trajectories from $\mathcal{B}$
        2. Compute advantage estimates $\hat{A}_t$ using the
           critic $V_\phi$
        for each agent $i = 1, ..., N$ do
            3. Update actor network $\theta_i$ by maximizing
               $L^{\text{CLIP}}(\theta_i)$
        4. Update critic network $\phi$ by minimizing the
           value loss
    5. Clear buffer $\mathcal{B}$
```

Algorithm 2: Decentralized Execution on Edge Device

```
Initialization:
1. Load the trained actor network $\pi_{\theta_i}$
while true do
    Task Arrival Handling:
    1. Wait for a new task arrival or decision epoch
    Local State Construction:
    2. Construct local observation vector $o_i$ based on
       current task queue, device status, and marketplace
       data
    Decision Making:
    3. Input $o_i$ into the actor network $\pi_{\theta_i}$
    4. Sample action $a_i$ from the output probability
       distribution
    Action Execution:
    if $a_i == 0$ then
        5. Schedule task for local execution
    else
        6. Initiate marketplace transaction to offload
           task to peer $a_i$
```

is the hallmark of PPO and is key to its stability [44]. The objective function for each actor i is

$$L^{\text{CLIP}}(\theta_i) = \hat{\mathbb{E}}_t \left[\min \left(r_t(\theta_i)\hat{A}_t, \text{clip}(r_t(\theta_i), 1-\epsilon, 1+\epsilon)\hat{A}_t \right) \right],$$
$$r_t(\theta_i) = \frac{\pi_{\theta_i}(a_t|o_t)}{\pi_{\theta_{i,\text{old}}}(a_t|o_t)}, \tag{4}$$

where: $r_t(\theta_i)$ denotes the probability ratio between the new and old policies. $\hat{A}_t$ is an estimate of the advantage function at timestep t, calculated by the centralized critic. It quantifies whether an action taken was better or worse than the policy's average action in that state. ϵ represents a small hyperparameter (e.g., 0.2) that defines the clipping range. The `clip` function constrains the probability ratio, preventing the policy from changing too drastically in a single update step, which avoids catastrophic performance drops and ensures more stable learning [45].

Meanwhile, the above critic network is trained concurrently by minimizing the mean-squared error between its value estimates and the actual observed returns. In particular, the advantage function $\hat{A}_t$ quantifies whether an action was better or worse than the policy's average. We use Generalized Advantage Estimation (GAE) for a stable estimate [46]:

$$\hat{A}_t = \sum_{l=0}^{\infty} (\gamma\lambda)^l \delta_{t+l}, \tag{5}$$

where $\delta_t = r_t + \gamma V_\phi(s_{t+1}) - V_\phi(s_t)$ is the Temporal Difference (TD) error, and $\lambda \in$ is the GAE parameter. GAE calculates the cumulative temporal difference error based on the $\gamma\lambda$ weight, balancing immediate and future rewards, thereby enhancing the long-term optimization capability of the scheduling strategy.

The critic is trained by minimizing the value loss, typically a mean-squared error. The overall loss function combines the policy loss, value loss, and an entropy bonus S to encourage exploration:

$$L(\theta, \phi) = L^{\text{CLIP}}(\theta) - c_1 L^{\text{VF}}(\phi) + c_2 S[\pi_\theta](s_t), \tag{6}$$

where $L^{\text{VF}}(\phi) = (V_\phi(s_t) - Y_t)^2$ is the value function loss, Y_t is the target value, and c_1, c_2 are coefficients.

The overall process is divided into an offline training phase and an online execution phase, which can be shown in Alg. 1 and Alg. 2, respectively.

C. Complexity Analysis

The computational complexity of the proposed DRL-based framework can be analyzed into two distinct phases: the offline training phase, which incurs a one-time, pre-deployment cost, and the online execution phase, which is crucial for achieving real-time performance on edge devices.

1) Offline Training Complexity: The offline training phase (Algorithm 1) is significantly more computationally intensive, as is typical for deep reinforcement learning. The total complexity is a function of the number of training episodes K_{episodes}, the number of steps per episode T_{steps}, the number of agents N, and the number of update epochs per episode E_{epochs}.

Let C_{actor} and C_{critic} denote the complexity of a forward pass through the actor and critic networks, respectively. The complexity of a backward pass is of the same order.

1) **Data Collection Phase:** In each of the T_{steps}, all N agents perform an action selection. This involves N parallel forward passes through their actor networks. The complexity of this phase is $O(T_{steps}NC_{actor})$.
2) **Network Update Phase:** For each of the E_{epochs}, the algorithm processes the collected data (T_{steps} transitions). This involves:
   - Calculating advantages, which requires one forward pass through the critic network for each timestep: $\mathcal{O}(T_{\text{steps}} \cdot C_{\text{critic}})$.
   - Updating each of the N actor networks: $\mathcal{O}(N \cdot C_{\text{actor}})$.
   - Updating the centralized critic network: $\mathcal{O}(C_{\text{critic}})$.

   The total complexity for the update phase is thus

$$O(E_{\text{epochs}} \cdot (T_{\text{steps}} \cdot N \cdot C_{\text{critic}} + N \cdot C_{\text{actor}} + C_{\text{critic}})). \quad (7)$$

2) Online Execution Complexity: During the online execution phase (Algorithm 2), each edge device makes scheduling decisions independently. The computational complexity for a single decision is determined by a forward pass through the agent's actor network.

Let the actor network be a Multi-Layer Perceptron (MLP) with L layers, and let n_k be the number of neurons in the k-th layer. The complexity of a forward pass is dominated by matrix-vector multiplications and is given by

$$\mathcal{O}\left(\sum_{k=1}^{L-1} n_k \cdot n_{k+1}\right). \quad (8)$$

Given that the network architecture is fixed, the values of n_k are constants. The size of the input and output are also relatively small and scales linearly with the number of devices N in the worst case. Therefore, the per-decision complexity is very low and can be considered approximately constant for a given network size. This ensures that the decision-making process is lightweight and does not introduce significant overhead, making it highly suitable for real-time deployment on resource-constrained edge devices.

V. Security Analysis

The integration of decentralized blockchain with economic incentives for resource sharing necessitates a rigorous analysis of the system's security posture. The BEST framework, by creating a market for tokenized RWAs such as computation and energy, introduces unique attack vectors alongside its novel capabilities. This section provides a security analysis of the BEST framework against several potential threats, detailing the inherent and designed defense mechanisms.

A. Resilience Against Sybil Attacks

A Sybil attack, where an adversary creates a large number of pseudonymous identities to gain disproportionate influence [47]. For instance, the attacker creates 100 fake edge devices, generates a large number of CTK/PTK without any real resource support, and manipulates the market price. It is a common threat in decentralized networks. In our BEST, an attacker might attempt to create numerous fake edge devices to manipulate the resource market, disrupt the DRL agents' learning process, or corner the resource supply.

The primary defense against this attack is the framework's foundational PoR mechanism, which is a core component of the RWA tokenization process. Unlike traditional systems, where creating an identity is free, in BEST, an identity is only economically viable if it can mint tradable assets (CTK and PTK). To do so, a device must verifiably prove to a decentralized oracle network that it possesses real, available computational and energy resources. This verification is anchored in hardware, ideally through a TEE or secure element, making the underlying resources difficult to forge [48]. Consequently, launching a large-scale Sybil attack becomes economically infeasible, as the cost of acquiring and maintaining the physical hardware to back each identity would be prohibitively expensive. This principle of tying digital identity to verifiably scarce off-chain assets is a cornerstone of securing RWA ecosystems.

B. Mitigation of Market Manipulation

Given that BEST operates a marketplace, it could be a target for market manipulation strategies such as wash trading, namely, colluding entities trading amongst themselves to create artificial volume and price signals, or pump-and-dump schemes. For example, two malicious nodes frequently engage in CTK transactions with each other, creating a false transaction volume and misleading the resource pricing judgments of other intelligent agents.

The framework has several layers of defense against such activities:

- **On-Chain Transparency:** All transactions are recorded on an immutable and publicly auditable blockchain ledger. This transparency makes it difficult to hide manipulative patterns like wash trading from on-chain analysis tools and vigilant participants [20].
- **Intelligent Agents:** The DRL agents are not naive traders. Their objective is not to maximize profit, but to minimize a cost function of latency and energy while ensuring task success. An agent's learned policy would not purchase tokens at an artificially inflated price if its internal calculation predicts that doing so would lead to a deadline violation or excessive energy cost. The agents' multi-objective, utility-driven nature makes them inherently resilient to price manipulation [49].
- **Immutable Market Logic:** The rules of the marketplace, e.g., order matching, settlement, are encoded in smart contracts. These rules are immutable and execute automatically, preventing any single entity from manipulating the market mechanics in its favor [50].

C. Defense Against Malicious Service Providers

A critical operational risk is the defaulting node problem. It is a resource provider that accepts a task and payment but fails to execute it, either maliciously or due to an unexpected failure. For instance, a resource supplier may deliberately halt the execution after receiving a task, or return incorrect results to defraud the payment of the cryptocurrency.

BEST mitigates this risk primarily through smart contract-based escrow. When a task is offloaded, the requester's payment is not transferred directly to the provider but is locked in an escrow smart contract. The funds are only released to the provider upon the successful completion of the task, which is cryptographically verified by the requester. If the provider defaults, the requester can issue a timeout or failure claim, allowing the smart contract to slash the provider's collateral and refund the payment. This mechanism removes the need for trust between parties and ensures that providers are only paid for work successfully delivered [50]. Furthermore, the immutable history of transactions on the blockchain naturally enables the creation of a decentralized reputation system, allowing agents to learn to avoid providers with a history of defaults.

D. Robustness to Foundational Blockchain and Oracle Attacks

The security of the BEST framework is ultimately dependent on the security of its underlying infrastructure: the blockchain network and the oracle system.

- **Blockchain Network Integrity:** An attack on the consensus mechanism of the underlying blockchain could allow for transaction reversal or censorship. The defense here is extrinsic to BEST and relies on the selection of a robust and highly decentralized blockchain. For such networks, the economic cost of acquiring enough hashing power or stake to launch a successful 51% attack is designed to be astronomical, providing a strong economic security guarantee [1].
- **Oracle Manipulation:** The oracle network, which bridges the off-chain world (device resources) and the on-chain world (tokens), is a critical security component. A compromised oracle could falsely report resource availability, allowing a malicious node to mint unbacked tokens. BEST's security model specifies the use of a decentralized oracle network. In such a system, data is not provided by a single source but is aggregated from a multitude of independent, cryptographically-secured nodes. An attacker would need to compromise a significant quorum of these nodes to corrupt the data feed, a task that is designed to be exceptionally difficult and expensive. This decentralized, multi-layered verification process is the industry standard for securing high-value RWA tokenization platforms[4].

In summary, BEST employs a defense-in-depth security strategy. It leverages the immutability, transparency properties of blockchain, consensus, smart contracts, and DRL agents to create a framework that is resilient to a wide range of attacks common in decentralized, economically-incentivized systems.

VI. Performance Evaluation

This section presents a comprehensive empirical evaluation of the BEST framework. We design a discrete-event simulator to model the proposed edge computing environment. We conduct extensive experiments to assess the performance of our DRL-based scheduler against several baseline algorithms across key metrics.

A. Simulation Environment and Parameters

The simulation environment was developed in Python, using the SimPy library for event-driven process management and PyTorch for implementing the MADRL agents. The environment simulates a network of N EDs, a stochastic task arrival process, a simplified wireless network model, and the blockchain-enabled resource marketplace. To ensure the reproducibility of our results, the key simulation parameters are detailed in Table II. Among them, each device e_i independently generates tasks, and the arrival intervals follow an exponential distribution. The task parameters (W_j, D_j, L_j^{max}) are sampled from a uniform distribution and locally generated. The simulation runs on a server that contains a 96-core Intel(R) Xeon(R) Gold 5220R CPU @ 2.20 GHz with 1 TB of memory.

TABLE II: Simulation Parameters

Parameter	Value(s)
Network Parameters	
Number of Devices (N)	5, 10, 15, 20, 25
Wireless Bandwidth	20 Mbps
Blockchain Latency	Uniformly distributed in [1, 2] s
Commun. Reserved Time	2 s
Δt	5 s
Device Parameters	
Device CPU Capacity	Uniformly distributed in [2, 3] GFLOPS
Device Battery Capacity	Uniformly distributed in [4, 5] kJ
CPU Power (Active)	Uniformly distributed in [5, 8] W
CPU Power (Idle)	Uniformly distributed in [0.5, 0.8] W
Exchange Rate η	24,000
Task Parameters	
Task Arrival Rate (λ)	0.05, 0.1,..., 0.25
Task Compute Req.	Uniformly distributed in [3, 8] GFLOPs
Task Data Size	Uniformly distributed in [2, 3] MB
Task Deadline	15 s
DRL Agent Parameters	
Actor/Critic Network	3-layer MLP (256-256-128)
PPO Learning Rate	1e-4
PPO Discount (γ)	0.99
PPO Clip Range (ϵ)	0.2
Objective Weights (w_l, w_e)	(0.5, 0.5)

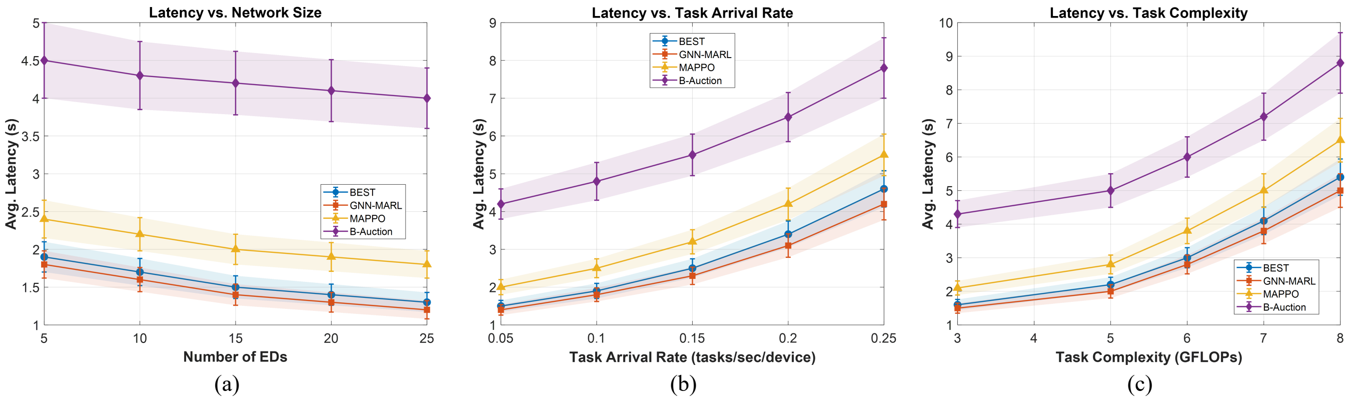
B. Comparison Schemes and Metrics

To effectively evaluate the performance of BEST, we compare it against three baseline algorithms representing different levels of scheduling sophistication:

- **MAPPO [44]**: It is a cooperative MARL baseline. Each edge device acts as a PPO agent, learning a collaborative scheduling policy. It utilizes the CTDE paradigm to overcome environmental non-stationarity, where a centralized critic guides the training of decentralized actors. This approach relies purely on learned implicit coordination to manage resources.

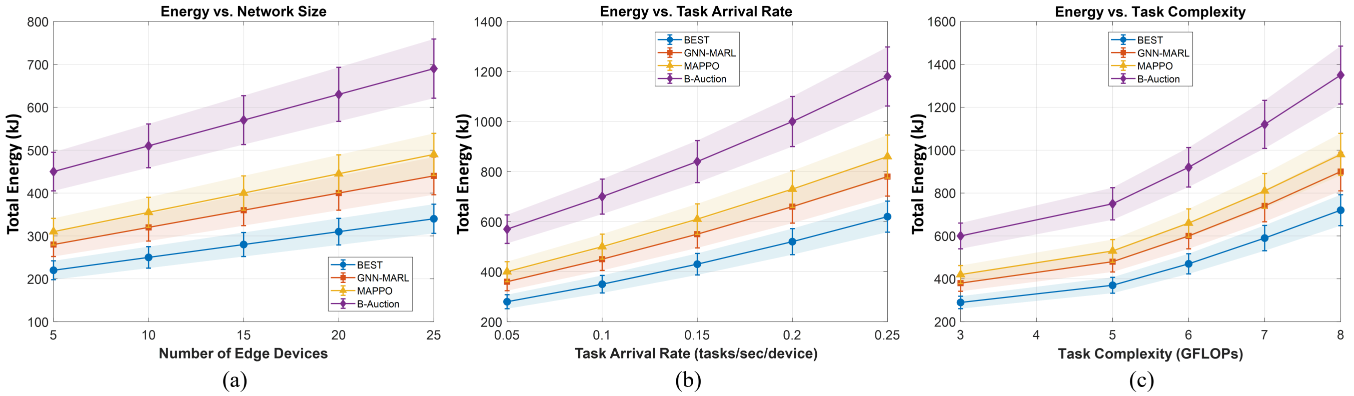
[4] https://chain.link/education-hub/real-world-assets-rwas-explained

This article has been accepted for publication in IEEE Transactions on Artificial Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAI.2026.3669544



(a) (b) (c)

Fig. 3: Average task completion latency.



(a) (b) (c)

Fig. 4: Total system energy consumption.

- **Graph Neural Network-Enhanced MARL (GNN-MARL) [51]**: It enhances the MAPPO architecture by incorporating a GNN to process the network's state. Each agent builds a local graph of its neighbors and tasks, allowing the GNN to create a rich, topology-aware state representation. This enables more sophisticated, context-aware decision-making.
- **Blockchain-based Multi-dimensional Auction (B-Auction) [52]**: Instead of learned cooperation, it implements a decentralized marketplace using a smart contract-based auction. When a task needs offloading, an auction is initiated. Other devices submit multi-dimensional bids that include not only price but also fairness metrics like their current CPU utilization. A second-price auction rule determines the winner, ensuring truthful bidding.

Additionally, we evaluate the performance of all algorithms based on four critical, system-wide metrics:

- **Average Task Completion Latency (s)**: This measures the average end-to-end time from a task's arrival at any device in the network to the moment its result is available. This is a primary measure of system responsiveness and quality of service.
- **Total System Energy Consumption (mJ):** This is the sum of all energy consumed by all devices in the network over the entire simulation duration. It includes energy for both computation (active and idle states) and data communication (transmitting and receiving task data). This is our primary measure of sustainability.
- **Task Success Rate (%):** This metric represents the percentage of all tasks generated in the system that are completed before their specified deadline. It is a key indicator of system reliability and effectiveness.
- **Computational Resource Utilization Rate (%):** This measures the efficiency of resource usage across the network. It is the percentage of the total available computational capacity (in GFLOPs-seconds) that is actively used for processing tasks over the simulation period.

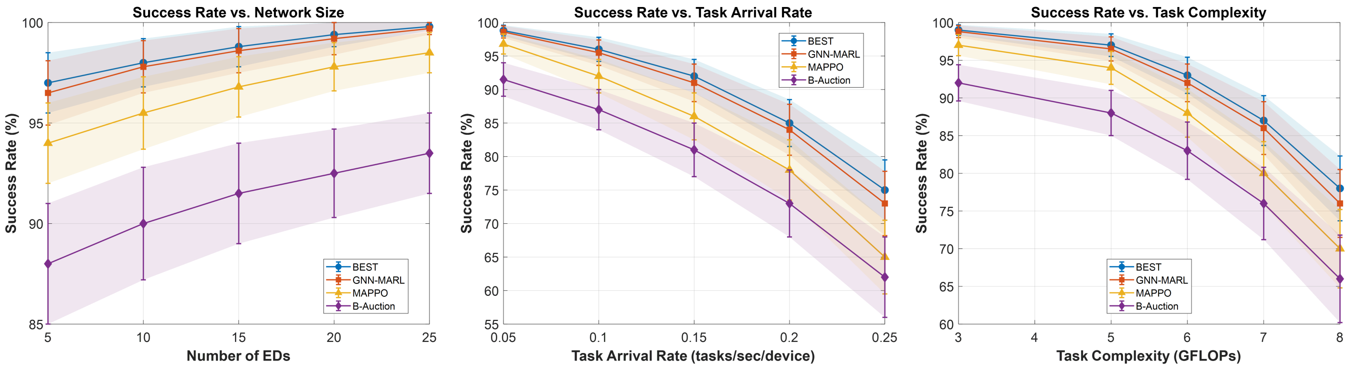
C. *Performance Results*

We conduct a comprehensive set of experiments by varying network size, system load, and workload complexity. The results are analyzed below, categorized by performance metric. Each data point represents the average of 10 independent simulation runs, with shaded areas indicating the standard deviation.

1) Average Task Completion Latency: In terms of latency, GNN-MARL consistently demonstrates the best performance, closely followed by BEST. As shown in Fig. 3 (a), the latency for all learning-based methods (GNN-MARL, BEST, MAPPO) generally improves with more devices as this provides more offloading options. Conversely, latency increases with higher system load and task complexity, as shown in Fig. 3 (b), (c).

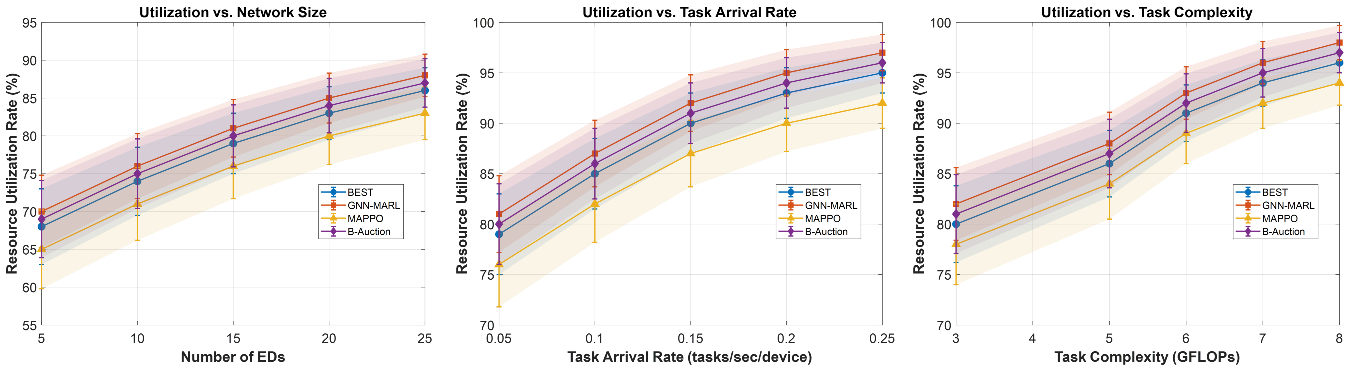
GNN-MARL excels because its primary strength lies in understanding network topology. The GNN enables agents

This article has been accepted for publication in IEEE Transactions on Artificial Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAI.2026.3669544



(a) (b) (c)

Fig. 5: Task success rate.



(a) (b) (c)

Fig. 6: Computational resource utilization rate.

to make highly informed decisions about the fastest path for a task, implicitly accounting for both computation and communication delays through its learned graph representations. BEST remains highly competitive. For instance, under the highest system load (task arrival rat $= 0.25$), BEST reduces average latency by 16.4% compared to MAPPO and a significant 41.0% compared to the B-Auction baseline. This demonstrates the high efficiency of its market mechanism, which rapidly matches tasks with powerful providers, suffering only a minor overhead compared to the pure, topology-aware GNN-MARL. B-Auction shows the highest latency, which is an expected trade-off for its decentralized, trustless nature. The overhead of initiating on-chain auctions and waiting for bid settlement inherently adds a delay that the learning-based, off-chain decision methods can avoid.

2) Total System Energy Consumption: BEST is the clear winner in minimizing energy consumption, showcasing the effectiveness of its sustainability-focused design. For all methods, total energy consumption naturally rises with more devices, higher load, and greater task complexity, as more work is being done across the system, as shown in Fig. 4.

BEST's superiority stems from its core design, which makes energy an explicit, tradable commodity via PTK. This provides a decisive advantage. When handling the most complex workloads (25 GFLOPs), BEST reduces total system energy consumption by 20.0% compared to the latency-focused GNN-MARL, 26.5% compared to the standard MAPPO, and a substantial 46.7% compared to the energy-intensive B-Auction method. This quantitative gap highlights that by forcing the DRL agent to economically account for energy, BEST learns to make more sustainable trade-offs than methods that treat energy merely as a component in a mixed reward function. B-Auction is the most energy-intensive due to the computational overhead of auction participation and the energy consumed by nodes interacting with the blockchain for every allocation decision.

3) Task Success Rate: BEST and GNN-MARL both achieve the highest and most robust task success rates. Success rates improve with more devices, as shown in Fig. 5 (a), due to increased resource availability. They degrade under heavy load and with more complex tasks as deadlines become harder to meet, as shown in Fig. 5 (b) and (c).

Under the most challenging high-load conditions (task arrival rate$= 0.25$), BEST maintains a 75% success rate. This represents a significant relative improvement in reliability of 15.4% over MAPPO and 21.0% over B-Auction. Its high success rate is driven by a market mechanism that is highly effective at finding and securing available resources before deadlines expire. GNN-MARL is equally strong because its superior understanding of the global network state allows it to accurately predict task completion times and avoid offloading to nodes that are likely to fail. MAPPO is slightly less reliable due to its less-informed view of the global state. B-Auction has the lowest success rate because its slower, auction-based

This article has been accepted for publication in IEEE Transactions on Artificial Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAI.2026.3669544

matching process can cause tasks to miss their deadlines, especially in high-traffic scenarios.

4) Computational Resource Utilization Rate: GNN-MARL and BEST demonstrate the most effective resource utilization, indicating superior load balancing. Utilization increases for all methods with higher system load and task complexity, as the demand for resources grows, as shown in Fig. 6.

GNN-MARL excels at utilization because its core strength is network-wide optimization. It can see the entire resource landscape through the graph and distribute tasks with exceptional efficiency to minimize idle time across all nodes. BEST proves to be an excellent load balancer as well, achieving a highly competitive utilization rate of 96% under the same conditions. This is a 2.1% relative improvement over the standard MAPPO baseline and nearly matches the performance of the market-clearing B-Auction. This indicates that BEST's market-driven approach is highly effective at minimizing idle resources by naturally routing tasks from overloaded devices to those with spare capacity. While B-Auction also shows high utilization, it comes at the cost of higher latency and energy consumption for the allocation process itself.

VII. Conclusion

The increasing deployment of LAMs at the network edge presents a critical sustainability challenge that cannot be solved by model-level optimizations alone. In this paper, we introduced BEST, a novel framework that addresses this challenge at the system level by creating a collaborative, market-driven ecosystem for edge resources. The core contributions of our work are the pioneering application of RWA tokenization to abstract and commoditize ephemeral compute and power resources, and the development of a cooperative multi-agent DRL scheduler to intelligently manage LAM workloads within this decentralized marketplace.

Our extensive simulation results, benchmarked against advanced alternatives, compellingly demonstrate the effectiveness and unique advantages of the BEST framework, especially superior sustainability. It drastically reduces total energy consumption, achieving up to a 46% reduction compared to market-based auction mechanisms and over 26% compared to other advanced MARL schedulers that lack this explicit economic incentive. Also, BEST achieves highly competitive latency and maintains an equally robust task success rate, proving it does not sacrifice performance for sustainability. These findings confirm that by transforming a collection of isolated, constrained devices into a cooperative, resource-sharing collective, we can unlock the full potential of edge LAM in a scalable and sustainable manner.

References

[1] H. Luo, J. Luo, and A. V. Vasilakos, "Bc4llm: A perspective of trusted artificial intelligence when blockchain meets large language models," *Neurocomputing*, vol. 599, p. 128089, 2024.

[2] S. R. Dubey and S. K. Singh, "Transformer-based generative adversarial networks in computer vision: A comprehensive survey," *IEEE Transactions on Artificial Intelligence*, vol. 5, no. 10, pp. 4851–4867, 2024.

[3] R. Zhang, H. Du, Y. Liu, D. Niyato, J. Kang, Z. Xiong, A. Jamalipour, and D. I. Kim, "Generative ai agents with large language model for satellite networks via a mixture of experts transmission," *IEEE Journal on Selected Areas in Communications*, vol. 42, no. 12, pp. 3581–3596, 2024.

[4] Z. Wang, Y. Shi, and K. B. Letaief, "Edge large ai models: Collaborative deployment and iot applications," *IEEE Internet of Things Magazine*, 2025.

[5] H. Luo, Y. Liu, R. Zhang, J. Wang, G. Sun, D. Niyato, H. Yu, Z. Xiong, X. Wang, and X. Shen, "Toward edge general intelligence with multiple-large language model (multi-llm): Architecture, trust, and orchestration," *IEEE Transactions on Cognitive Communications and Networking*, 2025.

[6] J. Wang, H. Du, D. Niyato, J. Kang, Z. Xiong, D. I. Kim, and K. B. Letaief, "Toward scalable generative ai via mixture of experts in mobile edge networks," *IEEE Wireless Communications*, vol. 32, no. 1, pp. 142–149, 2025.

[7] Y. Mao, X. Yu, K. Huang, Y.-J. A. Zhang, and J. Zhang, "Green edge ai: A contemporary survey," *Proceedings of the IEEE*, vol. 112, no. 7, pp. 880–911, 2024.

[8] N. Jegham, M. Abdelatti, L. Elmoubarki, and A. Hendawi, "How hungry is ai? benchmarking energy, water, and carbon footprint of llm inference," *arXiv preprint arXiv:2505.09598*, 2025.

[9] D. Patterson, J. Gonzalez, Q. Le, C. Liang, L.-M. Munguia, D. Rothchild, D. So, M. Texier, and J. Dean, "Carbon emissions and large neural network training," *arXiv preprint arXiv:2104.10350*, 2021.

[10] G. Varoquaux, S. Luccioni, and M. Whittaker, "Hype, sustainability, and the price of the bigger-is-better paradigm in ai," in *Proceedings of the 2025 ACM Conference on Fairness, Accountability, and Transparency*, 2025, pp. 61–75.

[11] H. Jiang, X. Dai, Z. Xiao, and A. Iyengar, "Joint task offloading and resource allocation for energy-constrained mobile edge computing," *IEEE Transactions on Mobile Computing*, vol. 22, no. 7, pp. 4000–4015, 2022.

[12] W. Feng, R. Xiao, Z. Li, H. Yu, G. Sun, L. Luo, M. Guizani, and Q. Ho, "Learning in chaos: Efficient autoscaling and self-healing for distributed training at the edge," *arXiv preprint arXiv:2505.12815*, 2025.

[13] H. Cheng, M. Zhang, and J. Q. Shi, "A survey on deep neural network pruning: Taxonomy, comparison, analysis, and recommendations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 12, pp. 10 558–10 578, 2024.

[14] K. Egashira, M. Vero, R. Staab, J. He, and M. Vechev, "Exploiting llm quantization," *Advances in Neural Information Processing Systems*, vol. 37, pp. 41 709–41 732, 2024.

[15] L. Che, J. Wang, X. Liu, and F. Ma, "Leveraging foundation models for multi-modal federated learning with incomplete modality," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2024, pp. 401–417.

[16] D. Wright, C. Igel, G. Samuel, and R. Selvan, "Efficiency is not enough: A critical perspective of environmentally sustainable ai," *Communications of the ACM*, vol. 68, no. 7, pp. 62–69, 2025.

[17] H. Luo, G. Sun, Y. Liu, D. Zhao, D. Niyato, H. Yu, and S. Dustdar, "A weighted byzantine fault tolerance consensus driven trusted multiple large language models network," *IEEE Transactions on Cognitive Communications and Networking*, 2025.

[18] M. Fahim, S. A. Kazmi, V. Sharma, H. Shin, and T. Q. Duong, "Edge intelligence: A deep distilled model for wearables to enable proactive eldercare," *IEEE Transactions on Artificial Intelligence*, vol. 6, no. 7, pp. 1736–1745, 2025.

[19] H. Luo, G. Sun, J. Wang, H. Yu, D. Niyato, S. Dustdar, and Z. Han, "Wireless blockchain meets 6g: The future trustworthy and ubiquitous connectivity," *IEEE Communications Surveys and Tutorials*, 2025.

[20] Y. Liu, H. Du, D. Niyato, J. Kang, Z. Xiong, A. Jamalipour, and X. Shen, "Prosecutor: Protecting mobile aigc services on two-layer blockchain via reputation and contract theoretic approaches," *IEEE Transactions on Mobile Computing*, vol. 23, no. 12, pp. 10 966–10 983, 2024.

[21] H. Luo, R. Zhang, Y. Liu, G. Sun, H. Yu, and Z. Han, "Real world assets on-chain assistance low-altitude computility networks: Architecture, methodology, and challenges," *IEEE Internet of Things Magazine*, 2026.

[22] S. Chen, M. Jiang, and X. Luo, "Exploring the security issues of real world assets (rwa)," in *Proceedings of the Workshop on Decentralized Finance and Security*, 2024, pp. 31–40.

[23] T. Meuser, L. Lovén, M. Bhuyan, S. G. Patil, S. Dustdar, A. Aral, S. Bayhan, C. Becker, E. De Lara, A. Y. Ding *et al.*, "Revisiting edge ai: Opportunities and challenges," *IEEE Internet Computing*, vol. 28, no. 4, pp. 49–59, 2024.

[24] Y. Sun, Y. Liu, S. Guo, X. Qiu, J. Chen, J. Hao, and D. Niyato, "Edge large ai model agent-empowered cognitive multimodal semantic communication," *IEEE Transactions on Mobile Computing*, 2025.

[25] K. Xu, L. Chen, and S. Wang, "Towards robust nonlinear subspace clustering: A kernel learning approach," *IEEE Transactions on Artificial Intelligence*, 2025.

[26] P. Arjun, S. Suryanarayan, R. Viswamanav, S. Abhishek, and T. Anjali, "Unveiling underwater structures: Mobilenet vs. efficientnet in sonar image detection," *Procedia Computer Science*, vol. 233, pp. 518–527, 2024.

[27] L. Cruz, X. Franch, and S. Martínez-Fernández, "Innovating for tomorrow: The convergence of software engineering and green ai," *ACM Transactions on Software Engineering and Methodology*, vol. 34, no. 5, pp. 1–13, 2025.

[28] M. Armoni, "Tensor processing units (tpu): A technical analysis and their impact on artificial intelligence," *Tech4Future Information Technology Report*, 2023.

[29] H. Huang, W. Lin, J. Lin, and K. Li, "Power management optimization for data centers: A power supply perspective," *IEEE Transactions on Sustainable Computing*, 2025.

[30] H. Luo, K. Yang, Q. Huang, and S. Dustdar, "A novel hierarchical co-optimization framework for coordinated task scheduling and power dispatch in computing power networks," *arXiv preprint arXiv:2508.04015*, 2025.

[31] L. Lin, J. Wu, Z. Zhou, J. Zhao, P. Li, and J. Xiong, "Computing power networking meets blockchain: A reputation-enhanced trading framework for decentralized iot cloud services," *IEEE Internet of Things Journal*, vol. 11, no. 10, pp. 17 082–17 096, 2024.

[32] N. Weerasinghe, P. Porambage, A. Braeken, M. Liyanage, and M. Ylianttila, "Tokennet: A novel tokenized resource marketplace for 6 g network slicing," *IEEE Transactions on Network Science and Engineering*, 2025.

[33] M. Dai, G. Sun, H. Yu, and D. Niyato, "Maximize the long-term average revenue of network slice provider via admission control among heterogeneous slices," *IEEE/ACM Transactions on Networking*, vol. 32, no. 1, pp. 745–760, 2023.

[34] Y. Liu, K. Wang, Y. Lin, and W. Xu, "Lightchain: a lightweight blockchain system for industrial internet of things," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 6, pp. 3571–3581, 2019.

[35] B. Qiu, Y. Wang, H. Xiao, and Z. Zhang, "Deep reinforcement learning-based adaptive computation offloading and power allocation in vehicular edge computing networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, no. 10, pp. 13 339–13 349, 2024.

[36] D. Luo, Q. Cai, G. Sun, H. Yu, and D. Niyato, "Split-chain-based efficient blockchain-assisted cross-domain authentication for iot," *IEEE Transactions on Network and Service Management*, vol. 21, no. 3, pp. 3209–3223, 2024.

[37] H. Du, R. Zhang, D. Niyato, J. Kang, Z. Xiong, S. Cui, X. Shen, and D. I. Kim, "Reinforcement learning with llms interaction for distributed diffusion model services," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

[38] D. Hou, W. Ma, W. Zhang, Y. Li, Y. Du, and Y. Hao, "An on-chain trading model of real world asset backed digital assets," *IET Blockchain*, vol. 4, no. 4, pp. 315–323, 2024.

[39] Z. Guo, H. Pan, A. He, Y. Dai, X. Huang, X. Si, C. Yuen, and Y. Zhang, "Trusted execution environments for blockchain: Towards robust, private, and scalable distributed ledgers," *IEEE Internet of Things Journal*, 2025.

[40] T. Conley, N. Diaz, D. Espada, A. Kuruvilla, S. Mayne, and X. Fu, "Izpr: Instant zero knowledge proof of reserve," in *International Conference on Financial Cryptography and Data Security*. Springer, 2024, pp. 225–239.

[41] X. He, C. You, and T. Q. Quek, "Age-based scheduling for mobile edge computing: A deep reinforcement learning approach," *IEEE Transactions on Mobile Computing*, vol. 23, no. 10, pp. 9881–9897, 2024.

[42] J. Pey, S. B. P. Samarakoon, M. V. J. Muthugala, and M. R. Elara, "A decentralized partially observable markov decision process for complete coverage onboard multiple shape changing reconfigurable robots," *Expert Systems with Applications*, vol. 271, p. 126565, 2025.

[43] A. Suzuki, M. Kobayashi, and E. Oki, "Multi-agent deep reinforcement learning for cooperative computing offloading and route optimization in multi cloud-edge networks," *IEEE Transactions on Network and Service Management*, vol. 20, no. 4, pp. 4416–4434, 2023.

[44] X. Ning, M. Zeng, M. Hua, and Z. Fei, "Multiple reconfigurable intelligent surfaces aided vehicular edge computing networks: A mappo-based approach," *IEEE Transactions on Vehicular Technology*, vol. 73, no. 11, pp. 17 496–17 509, 2024.

[45] M. Elsayed, Q. Lan, C. Lyle, and A. R. Mahmood, "Weight clipping for deep continual and reinforcement learning," *arXiv preprint arXiv:2407.01704*, 2024.

[46] Y. Chen, F. Zhang, and Z. Liu, "Adaptive bias-variance trade-off in advantage estimator for actor–critic algorithms," *Neural Networks*, vol. 169, pp. 764–777, 2024.

[47] H. B. Tulay and C. E. Koksal, "Sybil attack detection based on signal clustering in vehicular networks," *IEEE Transactions on Machine Learning in Communications and Networking*, vol. 2, pp. 753–765, 2024.

[48] Y. Xie, Q. Wang, S. Li, R. Xiao, C. Zhang, and L. Wei, "Secure and efficient decentralized bitcoin mixing scheme using trusted execution environment," in *ICC 2022-IEEE International Conference on Communications*. IEEE, 2022, pp. 4390–4395.

[49] N. Pippas, E. A. Ludvig, and C. Turkay, "The evolution of reinforcement learning in quantitative finance: A survey," *ACM Computing Surveys*, vol. 57, no. 11, pp. 1–51, 2025.

[50] Y. Lin, Z. Gao, H. Du, D. Niyato, J. Kang, Z. Xiong, and Z. Zheng, "Blockchain-based efficient and trustworthy aigc services in metaverse," *IEEE Transactions on Services Computing*, vol. 17, no. 5, pp. 2067–2079, 2024.

[51] G. Bernárdez, J. Suárez-Varela, A. López, X. Shi, S. Xiao, X. Cheng, P. Barlet-Ros, and A. Cabellos-Aparicio, "Magnneto: A graph neural network-based multi-agent system for traffic engineering," *IEEE Transactions on Cognitive Communications and Networking*, vol. 9, no. 2, pp. 494–506, 2023.

[52] E. Wu and Z. Peng, "Research progress on incentive mechanisms in mobile crowdsensing," *IEEE Internet of Things Journal*, vol. 11, no. 14, pp. 24 621–24 633, 2024.