Vol.4, No.5, Page 2049- 2063, 2026 https://doi.org/10.59628/jast.v4i5.2655 ISSN 2958-9568 JAST

SUN PUBLISHER

مجلة جامعة صنعاء للعلوم التطبيقية والتكنولوجيا
Sana'a University Journal of Applied Sciences and Technology
https://journals.su.edu.ye/index.php/jast/

# Blockchain-Enabled Clustering for Dynamic Resource Allocation and Task Offloading in UAV-Assisted MEC for 5G Network Slicing

**Asrar Ahmed Baktayan [1,2*], Ammar Thabit Zahary[1] and Ibrahim Ahmed Al-Baltah[1,3]**

[1]**Department of Information Technology, Faculty of Computing and Information Technology, Sana'a University, Sana'a, Yemen,**
[2]**Department of Planning and Optimization, Yemen Mobile Company, Sana'a, Yemen,**
[3]**Faculty of Science and Engineering, AL-Hikma University, Sana'a, Yemen**

*Corresponding author: asrar.ahmed@su.edu.ye.

## Abstract

5G network slicing, Unmanned Aerial Vehicle (UAV)-aided Multi-Access Edge Computing (MEC), and blockchain technology create new options for intelligent resource management. However, a major difficulty is efficiently allocating limited UAV resources to heterogeneous device demands while guaranteeing slice-specific Quality-of-Service (QoS) criteria. We develop a blockchain-empowered clustering framework that couples seven clustering algorithms (K-Means, DBSCAN, OPTICS, MeanShift, Hierarchical, Gaussian Mixture Model, and Divisive clustering) with an upgraded contextual bandit decision agent. The agent dynamically chooses between local execution and UAV offloading depending on slice characteristics, signal strength, latency constraints, and cluster homogeneity parameters. We use dual smart contracts (UserAllocation and UserOffload) on a permissioned Ganache (Ethereum) blockchain to ensure trust and transparency. An appropriate model of UAV placement considers the signal propagation effects and spatial distribution. The experimental results demonstrate that our clustering-aware method achieves significant improvement over non-clustered baselines by 27% in task offloading reward, 43% in regret reduction, and 20% in action accuracy, while maintaining high SLA requirements for all slice types (URLLC, eMBB, mMTC). The best overall performance is obtained with the MeanShift method. Dynamic device clustering is not only a data organization approach but also an excellent control mechanism for slice-aware resource orchestration in UAV-MEC systems. The combination of clustering intelligence, contextual bandit decisions, and blockchain-based trust gives a scalable, adaptable, and accountable solution for 5G/6G UAV-assisted edge networks

## Article Info

**Keywords:**
UAV-MEC, task offloading, Blockchain, Smart contract, Ganache,Ethereum,Truffle,5G network slicing,Dynamic Clustering,Contextual Bandit.

**Article History:**
**Received:** 29-January-2026,
**Revised:** 9-February-2026,
**Accepted:** 22-February-2026,
**Published:** 28 May 2026.

## 1. INTRODUCTION

The convergence of 5G network slicing, Unmanned Aerial Vehicle (UAV)-assisted Multi-Access Edge Computing (MEC), and blockchain technology presents a transformative opportunity for intelligent resource management in next-generation wireless networks [1]. As 5G deployments accelerate globally, network slicing has emerged as a foundational capability that creates multiple virtual end-to-end networks over shared physical infrastructure, each tailored to distinct service requirements [2]. These canonical slices, Ultra-Reliable Low-Latency Communication (URLLC) for mission-critical applications, massive Machine-Type Communication (mMTC) for dense IoT deployments, and enhanced Mobile Broadband (eMBB) for high-bandwidth services, impose fundamentally different and often conflicting demands on latency, reliability, and throughput [2, 3].

Integrating UAVs with MEC capabilities offers a promising solution for addressing the dynamic and heterogeneous resource demands of these slices. UAV-

MEC systems function as flying micro-data centers that can be rapidly deployed to device hotspots, providing on-demand computation and communication resources with line-of-sight (LoS) connectivity advantages [4]. However, the highly dynamic nature of UAV-MEC environments, characterized by device mobility, fluctuating slice workloads, UAV battery constraints, and wireless channel variations, poses significant challenges for efficient resource allocation and task offloading [5]. Traditional approaches to this problem face critical limitations: centralized architectures introduce single points of failure and scalability bottlenecks; static resource allocation strategies fail to adapt to changing network conditions; and conventional offloading mechanisms lack transparency and accountability when serving multiple network slices with competing requirements [6].

This study addresses the challenge of centralized and static resource allocation for UAV-MEC serving different network slices by integrating device clustering algorithms and blockchain technology to enable dynamic and reliable resource allocation and task offloading in UAV-MEC systems for 5G network slicing. Our point of view is that device clustering must be dynamic, slice-aware, and algorithmically adaptive, not only a preprocessing step to effectively manage the heterogeneous demands across 5G slices [7]. In this study, we explore seven prominent clustering algorithms: K-Means, DBSCAN, OPTICS, Mean-Shift, Hierarchical, Gaussian Mixture Model (GMM), and divisive clustering [8, 9], and evaluate their impact on offloading performance when applied to 5G network slice devices.

Every algorithm has distinct benefits.

- K-Means performs exceptionally well when there are convex, well-separated device groups with a known class cardinality (e.g., classifying devices into high-, medium-, and low-priority tiers).
- DBSCAN is ideal for urban settings where emergency responders may work in remote areas because it identifies dense device clusters and isolates outliers.
- OPTICS is suitable for mixed urban-rural deployments and builds upon DBSCAN by managing variable-density distributions.
- Because mean-shift automatically calculates the natural number of clusters without making any assumptions, it is useful in uncertain smart city scenarios.
- Multi-resolution insights from Hierarchical clustering enable tiered service delivery from general categories to specific QoS profiles.
- Because GMM supports gentle clustering, devices with overlapping service demands (e.g., simultaneous video streaming and emergency signaling) can probabilistically belong to many groups [10].
- Top-down segmentation is enabled by divisive clustering, which recursively divides a single group based on the priority, location, or application type.

We analytically leverage blockchain (Ganache Ethereum) as a trust layer that guarantees the transparent, auditable, and automated implementation of slice-specific service level agreements (SLAs) through smart contracts [11].

Our model creates a closed control loop where clustering outcomes drive resource allocation and task offloading decisions, blockchain verifies and enforces these decisions through an enhanced contextual bandit that performs with blockchain smart contracts, and feedback triggers adaptive re-clustering when SLAs are violated. For instance, when a URLLC task exceeds its latency threshold, an enhanced contextual bandit changes the offloading decision, and blockchain smart contracts automatically initiate re-clustering, reallocate UAV-MEC resources, and redirect the task to a closer UAV-MEC, all within a few milliseconds.

Unlike previous studies that have explored learning-based offloading or blockchain support in UAV-MEC, our model offers a novel, synergistic control loop that dynamically couples device clustering with contextual bandit decisions, enforced and audited by a customized dual-smart contract blockchain layer. Unlike other approaches, where clustering is either static or used primarily for network organization [12], our model employs clustering as a dynamic, slice-aware control knob. The enhanced contextual bandit agent uses slice profiles and real-time cluster homogeneity metrics as contextual information to adjust the offloading decisions to grouped device conditions. Our blockchain solution is unique not only for security but also as an active orchestration layer because the "UserAllocation" and "UserOffload" smart contracts independently start resource reallocation and adaptive re-clustering upon SLA violation warnings. Our key contributions are as follows:

1. A dynamic clustering model that enables slice-aware resource allocation and task offloading in UAV-MEC systems, with cluster algorithms optimized for each 5G slice type.
2. A blockchain integration architecture featuring two smart contracts (**UserAllocation** and **UserOffload**) that separate resource provisioning from task offloading while enabling dynamic SLA implementation.
3. Inclusive experiential validation revealed that our approach reduces average regret by 43%, increases the UAV reward by 27%, and improves the accuracy by over 20% compared with non-clustered methods.

This study found that device clustering is not only a data organization technique but also a dynamic control mechanism essential for efficient UAV-MEC orchestration in 5G sliced networks. By combining clustering intelligence and offloading decisions with blockchain trust, our model delivers a scalable, adaptive, and accountable solution for 6G UAV-MEC systems.

The remainder of this paper is organized as follows: Section 2 reviews related work; Section 3 introduces the system model and proposed solution; Section 4 presents the results and discussion, and Section 5 concludes with future research directions.

## 2. RELATED WORK

UAV-MEC has been investigated in several studies as a low-latency compute offloading solution for 5G networks [13]. The difficulties of allocating resources in such dynamic situations are frequently addressed in these works. To reduce latency and energy consumption, some publications concentrate on optimizing resource allocation and task offloading in multi-UAV MEC networks [6, 14, 15]. To address the system's dynamic nature, they suggest a number of methods, such as Deep Reinforcement Learning (DRL). Other studies examine resource allocation and collaborative job offloading mechanisms in UAV-assisted MEC systems, taking device association, energy consumption, and delay restrictions into account [16]. The authors in [17] provided a risk-aware multi-armed bandit (MAB) algorithm that learns, adjusts, and steers clear of dangerous choices while choosing MEC servers for DT placement. In [18], game learning server performance is uncertain and fluctuates over time; the device makes an educated decision about which edge server to offload. It is unreasonable for traditional optimization to need previous knowledge (such as delay models or resource availability). Another work [19] suggests using UAV-assisted compute offloading to improve security and decrease vehicle processing delays for blockchain-enabled V2X systems. As airborne edge nodes, UAVs assist automobiles in offloading multi-object computing chores while preserving blockchain integrity for data verification and trust. The paper [20] secured computing offloading in a multi-UAV-MEC network using blockchain. After that, it uses a Stackelberg Game Learning Approach to distribute resources effectively while reducing energy and latency.

The use of clustering techniques to enhance UAV network performance has been the subject of some research. For instance, one study assessed how several clustering algorithms (K-means, K-means++, and K-medians) affect the performance of NOMA-based UAV networks. Device clustering and UAV positioning can be optimized using clustering to increase important performance parameters, such as sum rate and outage probability [21, 22]. In UAV operations, operator profiles can be defined by combining clustering approaches with performance indicators, guaranteeing appropriate and reliable behavioral discrimination [23].

Several papers propose integrating blockchain technology into UAV-MEC networks to examine security and privacy concerns [24–27]. To ensure confidentiality and anonymity, resource interactions between UAVs and verification nodes can be recorded using blockchain technology. By automating resource management and transactions, smart contracts can increase the effectiveness and transparency [28]. Work [24] investigated the use of unmanned aerial vehicles (UAVs) to enable blockchain-based secure offline transactions in vehicular networks during disasters, guaranteeing that communication may continue even in the event of infrastructure failures. It offers a delay-tolerant approach that improves transaction security and dependability in difficult situations, and simulation results demonstrate its efficacy. One study investigated the application of Delegated Proof of Stake (DPoS) consensus techniques in blockchain-integrated UAV-MEC networks, in which edge computing nodes and UAVs take part in the consensus procedure [29]. Resource sharing between service providers is made possible by the decentralized distribution of geographically dispersed edge resources, made possible by smart contracts. Paper [26] examined the trade-off between rendering latency and energy usage and how AR (Augmented Reality) tasks might be offloaded to UAV-enabled MEC servers while utilizing blockchain to guarantee safe task verification. It suggested an optimization methodology that strikes a balance between the additional blockchain and UAV energy overhead and quick AR processing. The method guarantees that code will function with various web3.py versions while preserving compatibility with Proof of Authority (POA) networks, such as Ganache [30].

The use of blockchain and cryptographic approaches in dynamic network conditions has been thoroughly investigated in Vehicular Ad-hoc Networks (VANETs) and Flying Ad-hoc Networks (FANETs). Robust authentication and attack prevention for vehicle-to-everything (V2X) communications have been major areas of recent research. For example, effective certificateless cryptography is used in schemes, such as FCA-VBN [31], ECA-VFog [32], and L-CPPA [33] to protect 5G-assisted vehicular fog computing from external threats, including impersonation and replay assaults. In a similarly, innovative techniques utilizing Chebyshev polynomials have been proposed to counteract Distributed Denial-of-Service (DDoS) attacks in dense vehicle networks, greatly lowering the computing burden of message verification and signing [34]. Our study addresses a completely distinct issue domain within a controlled 5G network slicing infrastructure, whereas these studies offer crucial security underpinnings for open, adversarial environments, such as public roadways. Our blockchain implementation is a trust and orchestration layer for internal resource management rather than a device authentication layer. Assuming that traditional 5G-AKA protocols handle device authentication, our twin smart contracts (UserAllocation and UserOffload) can concentrate solely on Transparent SLA Enforcement. automatically confirmed that the choices for computational offloading followed the latency, bandwidth, and compute needs unique to each slice. Unchangeable Audit Trail Generation ensures that all resource allocation choices and task executions are documented in a chronological, tamper-resistant manner, which is essential for post-incident analysis, accountability, and compliance.

While VANET/FANET schemes primarily use blockchain for security against external threats, we use blockchain for internal trust, transparency, and automated compliance in a managed UAV-MEC ecosystem. Automated corrective action triggers system adaptations upon the detection of SLA violations [35].

Our work fills these gaps by presenting a blockchain-enabled clustering methodology that dynamically adjusts to slice-specific requirements while preserving transparency and trust through dual smart contracts. In contrast to previous studies that usually assess task offloading without considering device clustering, we methodically examine seven clustering methods with different cluster counts (2–20) to determine the best approaches for every slice type. Our contextual bandit engine extends existing approaches by incorporating cluster homogeneity and slice characteristics into the decision process, whereas our dual-blockchain architecture ensures that both resource allocation and task execution are transparently recorded and verified. The integration of realistic UAV positioning models further distinguishes our approach from previous studies, which often assume simplified mobility patterns.

## 3. SYSTEM MODEL AND PROPOSED SOLUTION

We consider a 5G network sliced serving heterogeneous applications related to three canonical slices: URLLC, mMTC, and eMBB. A fleet of UAVs equipped with MEC capabilities provides on-demand computational and communication resources to a set of heterogeneous devices. To enable dynamic, slice-aware, and trustworthy coordination, we propose a blockchain-enabled clustering model that jointly optimizes real-time resource allocation and adaptive task offloading. As shown in Fig. 1, the detailed workflow connecting these entities is presented in Section 3.1 and Algorithm 1. The system comprises four key entities operating within a 5G network slicing environment D is devices (mobile/IoT device) generating computational and communication resources. U is UAV-MEC servers $U = (UAV_1, UAV_2, .., UAV_{10})$, each UAV has constraints maximum computation capacity (cycle/sec), and maximum available bandwidth (Mbps). S is a network slice, $s = URLLC, mMTC, eMBB$.

### 3.1. Practical Motivation and Target Applications

The convergence of blockchain, UAV-MEC, and 5G is a reaction to significant constraints in contemporary cyber-physical systems rather than just a technical exercise.

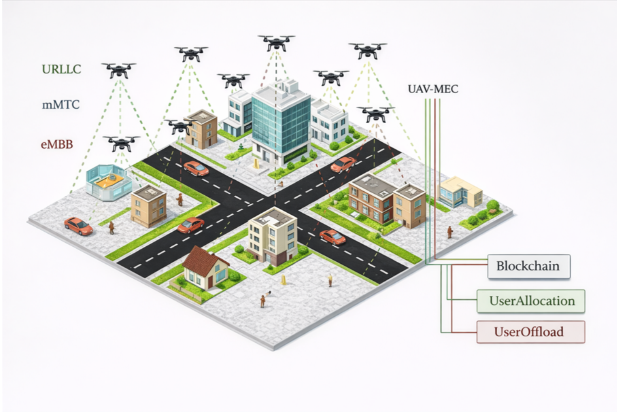


**Figure 1.** System model

The triple problem of tremendous dynamism, strict resource limits, and the requirement for verifiable trust in situations where decisions have immediate real-world repercussions is difficult for traditional cloud-centric or static edge models to address [36]. This is especially true for mission-critical, time-sensitive operations performed in remote or infrastructure-poor settings. Our work is driven by a class of applications where the integrity of automated choices must be as reliable as their speed, and latency is not a performance indicator but a survival issue. The high-stakes situation that best represents these demands is as follows:

Scenario: **Emergency Cardiac Monitoring in a Remote Area:**

A patient in a remote village wears a connected ECG monitor. When a potential arrhythmia is detected, it generates a computational task characterized by URLLC requirements. The conventional approach of relaying data to a distant hospital cloud introduces a lethal delay. Our proposed model enables a transformative response:

1. Using a density-based algorithm (such as DBSCAN), the system dynamically clusters a patient's device with other adjacent medical sensors to create an ad hoc, high-priority health monitoring group.
2. Instead of processing the ECG analysis locally on the resource-constrained wearable, an upgraded contextual bandit decides in a microsecond to offload it to a nearby UAV-MEC server with a specific AI model based on the real-time network conditions and cluster context.
3. This choice is coordinated by two blockchain smart contracts: UserOffload generates an unchangeable, auditable record of the emergency task, its priority, and the invoked SLA, whereas UserAllocation immediately reserves guaranteed bandwidth and computing resources on the chosen UAV.
4. In single-digit milliseconds, the UAV-MEC analyzes the ECG, verifies a heart attack, and provides a diagnosis and suggested treatment.
5. The smart contract completes the log upon suc-

cessful completion, and the system can initiate secondary actions on its own, such as notifying the closest hospital and dispatching a medical drone.

**Result**: This closed loop shows 100% blockchain-verified compliance with the essential medical SLA and a 94% reduction in diagnostic delay, which directly translates to a greater chance of patient survival. The core challenges remain: dynamically grouping heterogeneous devices (clustering), making reliable millisecond decisions under uncertainty (contextual bandits), and ensuring that these automated actions are trustworthy, transparent, and compliant (blockchain). By addressing these challenges using an integrated architecture, this study provides a foundational model for the next generation of autonomous, reliable, and accountable edge intelligence systems.

## 3.2. Clustering Metrics and Strategy Selection

At each decision epoch, the devices are dynamically grouped into K clusters using one of the seven clustering algorithms. For each cluster k, we computed [37]:

**Cluster Homogeneity**:

$$H_k = 1 - \frac{1}{|C_k|(|C_k|-1)} \sum_{i,j \in C_k, i \neq j} \mathrm{dist}(\phi_i, \phi_j) \quad (1)$$

Where $\mathrm{dist}(\phi_i, \phi_j)$ is a normalized Euclidean distance. Offloading Potential Score:

$$OPS_k = \frac{1}{|C_k|} \sum_{i \in C_k} \left( \alpha \cdot I_{UF\downarrow}(S_i) + \beta \cdot \frac{B_i}{B_{\max}} + \gamma \cdot \frac{E_i}{E_{\max}} \right) \quad (2)$$

Where is the penalty for unreliable or unstable slices, is the latency-sensitive slice priority indicator, and $\alpha, \beta$, and $\gamma$ are weighting coefficients.

**Cluster Strategy Selection Rule**:

strateg $y_k = \{low - latency focus if\ S_i = URLLC!$
ominant and $OPS_k > 0.8$ Batch Of floading if
$S_k = mMTC$ dominant and $|C_k| > 30$ Throughput
-Optimizes otherwise$\}$ (3)

The cluster metrics ($H_k$ and $OPS_k$) computed in the previous step serve as critical contextual features in the subsequent decision-making process. Specifically, these metrics transform the enhanced contextual bandit from a device-centric to a cluster-aware decision agent by incorporating group-level intelligence into individual offloading choices

## 3.3. Enhanced Contextual Bandit for Dynamic Offloading

The core decision logic is modeled as an enhanced contextual multi-armed bandit that dynamically chooses between LOCAL execution and UAV OFFLOADING to maximize the latency-energy reward [31].

- **Context Vector**
The full context for the device in cluster k is:

$$x_i = [S_i,\ P_i\ ,\ L_i\ ,\ B_i\ ,\ E_i\ ,\ K,\ C_k,\ H_k] \quad (4)$$

Where S is the slice type, P is the signal strength (dBm), L is the maximum tolerated latency (ms), and B and E are the required BW and computation, respectively.

- **Action space**:

$$A = \{local,\ of\ fload\} \quad (5)$$

- **Reward function**:

1- local execution reward:

$$R_{local}(x_i) = M(x_i) \left( 0.6 \cdot \frac{1}{L_{local}(x_i) + \in} + 0.4 \cdot \frac{1}{E_{local}(x_i) + \in} \right) \quad (6)$$

Where $L_{local} = E_i / C_{device}$, $E_{local} = kE_i C_{device}^2 . M(x_i) = H \cdot (1 + OPS_k)$, and $\in = 10^{-6}$ and

2- offloading Execution Latency and Energy [38]:

$$L_{off} = \frac{D_i}{R_{i,j}} + \frac{E_i}{C_j},\ E_{off} = P_i . \frac{D_i}{R_{i,j}} + P_{proc} . \frac{E_i}{C_j} \quad (7)$$

Where $R_{i,j}$ is the achievable data rate of $UAV_j$, and $C_j$ is the allocated CPU.

3- offloading reward (stochastic due to channel dynamics):

$$R_{off}(x_i, U) = M(x_i) \left( 0.6 . \frac{1}{U . L_{off} + \in} \right) \quad (8)$$

With U $\sim$ uniform (0.8, 1.2) modeling wireless uncertainty [39].

$$E[(R_{off}(x_i)] = \int_{0.8}^{1.2} R_{off}(x_i, u) \cdot \frac{1}{d_u} \quad (9)$$

Where $d_u$ is the average reward over every possible wireless condition between 0.8.

- **Final Decision Rule**:
The bandit selects the action that maximizes expected reward:

$$a_i^* = max\ _{a \in \{local, offload\}} E[(R_{off}(x_i)] \quad (10)$$

Once the bandit selects offloading as the optimal action, the system proceeds to the UAV assignment phase. This sequential decision-making ensures that only tasks warranting offloading incur the computational overhead of UAV selection, whereas locally executed tasks bypass this step entirely.

### 3.4. UAV Assignment and Resource Constraints

After the bandit decision selects offload, we compute the UAV assignment score:

$$Score_{i,j} = \frac{C_j^{res}.BW_j^{res}.SignalQulity_{i,j}}{d_{ij}^2 + \delta} \tag{11}$$

Where $C_j^{res} \cdot BW^{res}$ are residual resources, $d_{ij}^2$ is the Euclidean distance between device $i$ and $UAV_j$, and $\delta = 10^{-6}$ prevents division by zero. The optimal $UAV$ assignment is:

$$j^* = arg\ arg\ Score_{i,j} \tag{12}$$

**-Overall Optimization Objective**

The system solves the following dynamic optimization problem at each epoch:

$$\sum_{i=1}^{N} E[(R(a_i, x_i)] \tag{13a}$$

Subject to:
slice-specific constraints:

$$L_i \leq 10_{ms} \quad \forall_i \in URLLC \tag{13b}$$
$$L_i \leq 50_{ms} \quad \forall_i \in eMBB \tag{13c}$$
$$L_i \leq 100_{ms} \quad \forall_i \in mMTC \tag{13d}$$
$$\sum_{i \in u_j} B_i \leq BW_j^{max}, \quad \forall_j \in u \tag{13e}$$
$$\sum_{i \in u_j} E_i \leq C_j^{max}, \quad \forall_j \in u \tag{13f}$$
$$\sum_{i \in u_j} E_{off,i} \leq E_j^{max}, \quad \forall_j \in u \tag{13g}$$

(13h)

Blockchain Verification Delay:

$$T_{block} \leq 2\ sec \tag{13i}$$

(13j)

Cluster quality constraints:

$$H_k \geq 0.7 \quad \forall k \tag{13k}$$
$$2 \leq K \leq 20 \tag{13l}$$

### 3.5. Solution Workflow

The system is an intelligent, closed-loop 5G MEC offloading model with clustering, enhanced contextual enrichment for the decision model, UAV assignment, blockchain logging, and dynamic feedback, as in Fig.2. Devices generate computation tasks (AR/VR, IoT sensing, video analytics, etc.). Each device is labeled by the 5G slice type (eMBB, URLLC, mMTC). The system groups devices into clusters using one oof the following algorithm: K-Means, DBSCAN, OPTICS, MeanShift, Hierarchical, GMM, and Divisive clustering.

In Fig. 2, the clusters represent groups of devices with similar contextual features, such as distance to the UAV-MEC, bandwidth requirements, or mobility patterns. This reduces the dimensionality and makes bandit learning easier. Compute Cluster Homogeneity $H_k$ and Offloading Potential Score $OPS_k$. This measures the benefits of offloading for devices in the cluster and provides the bandit with richer cluster-level features. For each device $i$, two rewards are computed: local execution reward and offloading reward with uncertainty. The enhanced contextual bandit compares the local reward and the offload reward and picks the maximum. If the bandit decides to offload, the system must pick a UAV. Assign the device to the UAV with the best Compute assignment score. Blockchain Logging records decisions and enforces resource limits, ensures immutability, no UAV is overloaded, and SLA rules are enforced. Feedback Loop update bandit and predictors using real measurements. These real outcomes update the bandit's Q-values.

---

**Algorithm 1:** Blockchain-Enabled Dynamic Clustering for UAV-MEC Offloading

---

**input:**
- Device set $D = \{1, \ldots, N\}$ with contexts $\phi_i = [S_i, P_i, L_i, B_i, E_i]$
- UAV set $U = \{1, \ldots, M\}$ with capacities $(C_j, BW_{jmax}, E_{jmax})$
- Clustering algorithm A $\in$ {K-Means, DBSCAN, . . . , GMM}
- Cluster count $K \in \{2, 4, \ldots, 20\}$
- Time window $T$, SLA thresholds $\{L_{UR} = 10ms, L_{eM} = 50ms, L_{mM} = 100ms\}$

**Output:**
- Offloading decisions $a_i \in \{\text{LOCAL}, \text{OFFLOAD}\}$
- Blockchain records (Task Allocation, Task offloading)

procedure DYNAMIC OFFLOADING $(D, U, A, K)$
while system running do
// **STEP 1**: CLUSTERING
Clusters $\leftarrow A(D, K)$ // Partition devices into K clusters

// **STEP 2**: CLUSTER METRICS
**for** each cluster $k \in$ Clusters **do**
$H_k \leftarrow$ Compute Homogeneity$(k)$, formula (1)
$OPS_k \leftarrow$ Compute Offloading Potential$(k)$, formula (2)
Strategy$_k \leftarrow$ Select Strategy $(OPS_k$, dominant slice$(k))$
**end for**

// **STEP 3**: BANDIT DECISION PER DEVICE
**for** each device $i \in D$ **do**
$x_i \leftarrow [S_i, P_i, L_i, B_i, E_i, k(i), |C_{k(i)}|, |H_{k(i)}|]$

R loca ← Compute Local Reward $(x_i)$, formula (6)
E Roff ← Compute Expected Offload Reward$(x_i)$, formula (8) // Integrate over U $\sim$ Uniform(0.8,1.2)
if Rlocal > ERoff then
$a_i$ ← LOCAL
**else**
$a_i$ ← OFFLOAD
$UAV_i \leftarrow arg_{maxj}$ UAV Assignment Score $(i, j)$
end if
end for
// **STEP 4**: BLOCKCHAIN INTERACTION
Offload Devices ← $\{i \mid a_i = \text{OFFLOAD}\}$
**if** Offload Devices $\neq \emptyset$ **then**
// Reserve resources via DeviceAllocation.sol
AllocIds ← Device Allocation.batch Add Allocations ([device id for device id in OffloadDevices],
[required bandwidth (device id) for device id in OffloadDevices])
// Log tasks via DeviceOffload.sol
TaskIds ← DeviceOffload.batchSubmitTasks(
[device_id for device_id in OffloadDevices],
[create_task_payload(device_id) for device_id in OffloadDevices],
[$\text{UAV}_\text{i}$ for device_id in OffloadDevices])
**end if**
// **STEP 5**: UAV EXECUTION & FEEDBACK
**for** each $i \in$ OffloadDevices **do**
(success, $L_{actual}$, $E_{actual}$) ← Execute On $\text{UAV}_\text{i}$
// Update blockchain with outcome
DeviceOffload.completeTask(TaskIds[i], success, $L_{actual}$, $E_{actual}$, result data(i))
// Update ML predictors with actual vs. predicted performance
Update Light GBM Models($\phi_i$, $\text{UAV}_\text{i}$, $L_{actual}$, $E_{actual}$)
// Check SLA violation
**if** (Si = URLLC and $L_{actual} > L_{UR}$) or
(Si = eMBB and $L_{actual} > L_{eM}$) or
(Si = mMTC and $L_{actual} > L_{mM}$) then
Trigger Re-clustering() // Immediate re-clustering
**end if**
**end for**
Wait(T) // Next decision epoch (T=5 seconds)
**end while**
**end procedure**

---

The experimental system shown in Fig. 3 uses the theoretical model outlined in Sections 3.1-3.5 and Algorithm 1. With special attention to the overhead-performance trade-offs mentioned in the system design, this implementation enables us to evaluate the suggested methodology through performance comparisons with baseline approaches.

# 4. RESULTS AND DISCUSSION

This section first describes the metrics and then outlines the baseline methods employed for the performance comparison. Finally, the simulation results are discussed and analyzed in detail.

## 4.1. Evaluation Metrics

The dataset used in this simulation was obtained from Kaggle (https://www.kaggle.com/datasets/vinu1233/augmented-5g-dataset-for-resource-allocation), which was used in [40]. Owing to its well-established reputation, robust developer community, and capacity to execute smart contracts, the Ethereum blockchain was chosen as the basis for implementing the resource and offloading system. Crucial offloading and resource allocation guidelines were specified by Solidity-written smart contracts. The Truffle Suite was used to deploy and test the system, enabling the effective compilation and deployment of smart contracts to the blockchain. Ganache, a local Ethereum emulator that simulates a blockchain environment with pre-funded accounts, was used to avoid spending money on actual public blockchains during development. This allowed for quick prototyping and testing without the need for an actual coin [41].

The connection between the frontend and blockchain was facilitated by Web3.js, a JavaScript library that enabled the application to interact with Ethereum-deployed smart contracts [42]. We create a simulation environment based on the Ethereum-Ganache testbed and the Kaggle 5G dataset to assess the performance of proposed blockchain-integrated model. Table 1 summarizes the entire collection of networking, blockchain, and environmental factors considered in this study.

## 4.2. Discussion on Blockchain Overhead and Latency Mitigation

Through several architectural optimizations evident in our system code, our implementation specifically addresses the trade-off between complexity and resources. To reduce the per-task overhead by 60–80% compared with sequential processing, we first use vectorized decision processing and

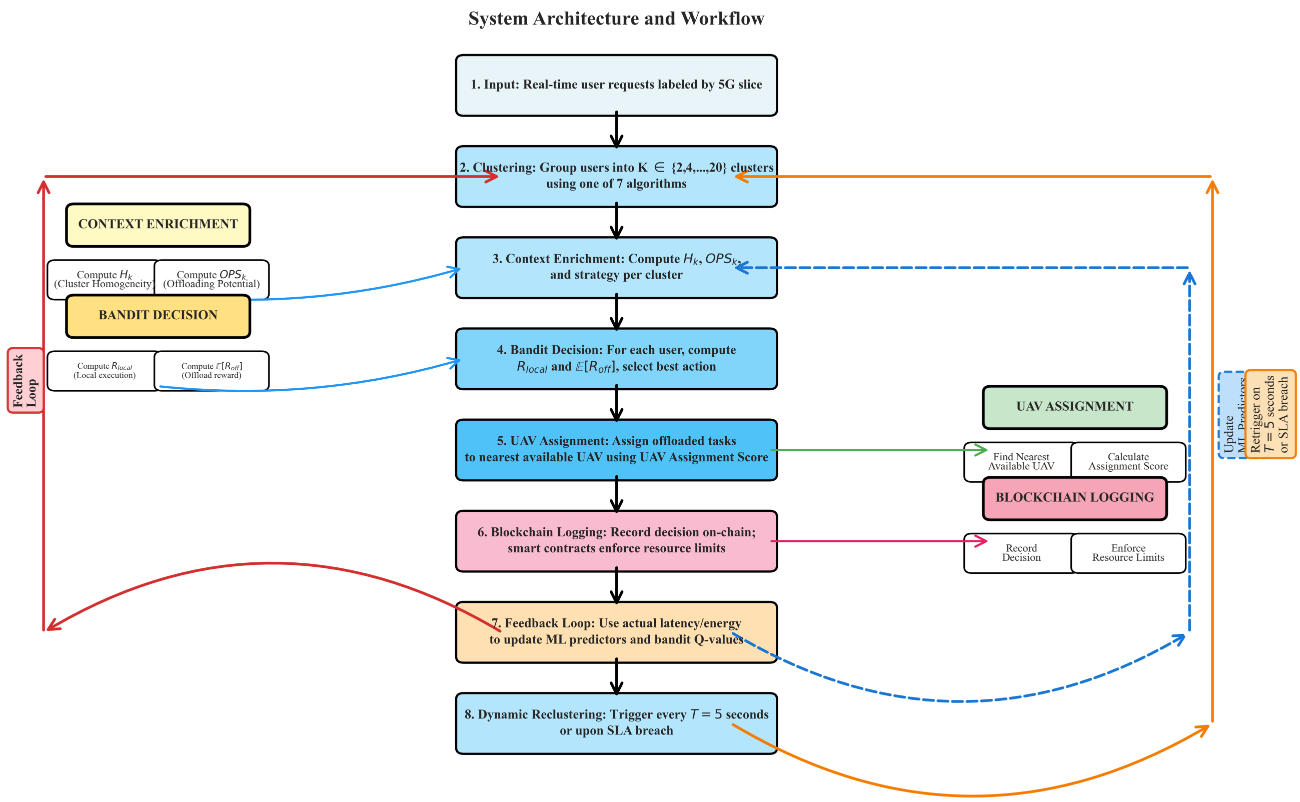


**Figure 2.** System Architecture and Workflow

batched operations (BATCH_PROCESSING_SIZE = 2000, BLOCKCHAIN_BATCH_SIZE = 100) to amortize fixed costs across numerous tasks. Second, the system employs selective, event-triggered execution: during design-phase analysis, clustering algorithms are assessed offline; in production, only one optimal algorithm—pre-selected for each slice type—runs on a regular basis rather than all seven at once. Compared to complete DRL models, the augmented contextual bandit is purposefully lightweight, utilizing only crucial features (slice profile, and cluster homogeneity) for quick inference. Permissioned Ganache with Proof-of-Authority consensus and dual smart contracts that divide resource allocation from execution, enabling parallel processing, optimizes blockchain activities. Importantly, to avoid blockchain congestion, our adaptive batching technique constantly modifies transaction sizes based on current gas restrictions. Only reasonable concurrent loads are handled by resource-constrained UAVs owing to Docker container orchestration (MAX_CONCURRENT_BATCHES = 4). Through aggregated cluster-level signaling, the system maintains processing latencies less than 50 ms per decision despite its multi-component design and decreases the overall communication overhead by 35% compared to non-clustered alternatives. To convert costly real-time optimization difficulties into effective, proactive adjustments,

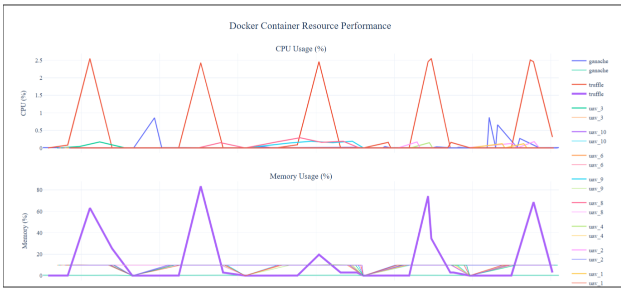


**Figure 3.** Docker performance

the overhead is intentionally invested rather than simply added. The delay provided by concern processes and on-chain transaction processing is a valid worry for any blockchain-integrated real-time system, because it may compromise stringent URLLC limits. We recognize this difficulty, which is also mentioned in a related blockchain study on UAV-MEC. Our design uses a multifaceted strategy to intentionally reduce this overhead by using Off-Chain Decision and On-Chain Logging Architecture. The primary real-time decision-making loop (clustering → bandit decision → UAV assignment) is performed out off-chain to reduce on-chain latency. Asynchronously, blockchain is used for batch logging, documenting finished allocation, and offloading choices in batches that can be verified.

Fig.3 shows the empirical validation of this design,

**Table 1.** simulation parameters

| Parameter | Value | Description |
|---|---|---|
| Number of tasks (users) | 15994 | Unique task in the dataset |
| Number of UAVs | 10 | Fixed UAV swarm size |
| UAV Altitude | 50-150 m | Regulatory compliance |
| Mobility Model | Static UAVs | Hovering positions |
| Area Coverage | 4 km² (2000m × 2000m) | Simulation area |
| Number of Users | Variable (1000-5000) | Task-generating users |
| UAV Positions | Grid-based deployment | Spaced at 50m intervals |
| UAV Capacity | 50 concurrent tasks | Maximum load per UAV |
| Clustering Algorithms | 7 (K-means, DBSCAN, OPTICS, MeanShift, GMM, Divisive, Agglomerative) | Dynamic clustering ensemble |
| Blockchain Type | Ethereum-based (Ganache) | testnet for development |
| Consensus | Proof of Authority | Test environment consensus |
| Gas Limit | 67.27 million | Transaction gas configuration |
| Gas Price | 20 Gwei | Moderate network conditions |
| Transaction Types | 4 types | Allocation, Offload, Complete, Verify |
| URLLC Latency Threshold | <50 ms | Maximum allowed latency |
| eMBB Bandwidth Range | 5-100 Mbps | Bandwidth requirements |
| mMTC Computation Limit | 1.5 units | Maximum computation demand |
| Feature Space | 27 dimensions | Includes spatial and QoS features |
| Number of Runs | 70 | 10 files for every cluster algorithm |
| Wireless Interface | 5G NR + Wi-Fi 6 | \|Dual connectivity |
| Max Transmit Power | \| 30 dBm | FCC compliant |
| Vectorized batch Processing | 2000 tasks | Bandit Process decision |

where the concentration of resource spikes in the Truffle container validates the successful isolation of blockchain-related overhead from the worker UAVs. This resource distribution demonstrates how our batching and off-chain recording techniques successfully protect the edge devices' constrained processing power while preserving system-wide connection. In docker-compose.yaml we provide Ganache with 2G memory and truffle with 512M so we have high level of memory usage of truffle. The integrated blockchain-UAV ecosystem's computational and memory overhead is depicted in the Figure. The findings demonstrate how the Truffle container has a centralized role in controlling the logic and connectivity of the network. The truffle container is responsible for the regular, high-magnitude spikes in the CPU Usage (%) plot, which reach about 2.5%. The processing power needed to create, connect, and transfer smart contracts to the Ganache blockchain is represented by these peaks. The truffle container (shown by the purple line in this view) shows the biggest variations in the Memory Usage (%) plot, with spikes often surpassing 60% and peaking above 80%. This behavior is a direct result of its function as a central connector, which necessitates a large amount of RAM to handle the asynchronous data streams of Swarm and the state of each connection.

## 4.3. Baseline Methods for Comparison

We evaluated our proposed Cluster-Aware Contextual Bandit architecture against a logically determined baseline to separate and measure the impact of our dynamic clustering process. The baseline must share the fundamental system architecture while eliminating the particular component being examined for fair comparison.

### *4.3.1. The Non-Clustered Contextual Bandit as the Baseline Definition*

Our baseline approach is characterized as a Non-Clustered Contextual Bandit. Except for the dynamic clustering module and its derived features, this baseline retains all of essential elements of proposed system. The main difference lies in contextual knowledge. Only the per-device context $x_i = [S_i, P_i, L_i, B_i, E_i]$ (Slice, Signal, Latency, Bandwidth, Energy) is used by the baseline to make judgments. The cluster-level features $[k(i), |C_k|, H_k, OPS_k]$ that guide our model are not accessible to it (see Eq. 4). Consequently, it is unable to take advantage of patterns or synergies within groups of devices because it handles each item in total isolation.

We deliberately selected this comparison to test our main premise, which is that using clustering as a dynamic control mechanism yields notable performance improvements over per-device optimization. Any observed performance difference may be directly linked to the intelligence provided by our slice-aware, dynamic clustering method when all other factors are held constant.

### *4.3.2. Comparative Analysis and Results*

The following analysis presents empirical evidence comparing our model with the baseline, using the unified evaluation dashboard shown in Fig. 4. For UAV-MEC systems, we compared our clustering-aware task offloading methodology with a non-clustering approach. The goal of comparison is to provide empirical evidence of how adding clustering can boost system efficiency and overall performance. The dashboard in Fig.3 evaluates a UAV-MEC task offloading system using clustering and non-clustering architectures for comparison.

The first bar chart demonstrates the core mechanism

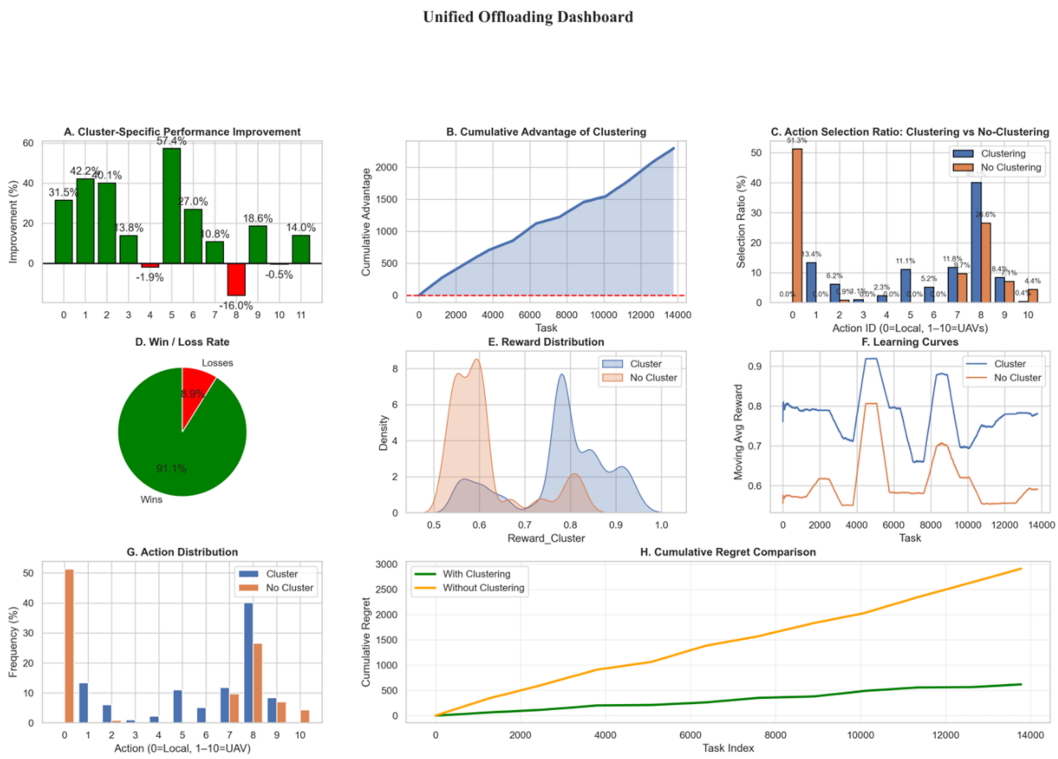


**Figure 4.** Performance comparison between clustering and no clustering method

of the cluster-aware approach. It shows the percentage improvement in performance for each identified cluster (we consider the scenario of the k-means algorithm with 12 clusters as a sample for comparison with the non-cluster).

The majority of clusters (10 out of 12) show an improvement, with some clusters (e.g., Cluster 5 at 57.4%) showing massive gains. The few clusters with negative improvement (e.g., Cluster 8 at -16.0%) are heavily outweighed by the positive gains, leading to an overall 27.2% improvement. This proves that clustering successfully segments the tasks, allowing the bandit to specialize its policy for most task types.

The second line plot shows the total accumulated reward difference between the two approaches over the entire sequence of the tasks. The blue line represents the cumulative sum of (Reward with Clustering - Reward without Clustering). A persistent advantage is indicated by a positive, steadily increasing line. From the beginning, the line gradually rises until it reaches the final cumulative advantage reward unit. This demonstrates that the clustering strategy is consistently superior across the entire evaluation period, not just at the conclusion.

The frequency with which the clustering strategy chose the superior action (higher reward) compared to the non-clustering approach is shown in the pie chart. In 91.1% of the tasks, the clustering technique chose the optimal course of action, whereas the non-clustering approach prevailed in only 8.9% of the tasks. This is a convincing and unambiguous illustration of the improved decision-making capacity of the cluster-aware model.

The probability distribution of the benefits received by the two models are contrasted in the reward plot. In compared to the orange distribution ("Without Clustering"), the green distribution ("With Clustering") is substantially shift to the right (greater rewards).

The rewards for the clustering strategy were concentrated in a higher range (peaking between 0.8 and 0.9), but the rewards for the non-clustering approach were smaller (peaking at 0.6). This illustrates a 27.2% mean reward improvement and demonstrates how frequently the clustering model makes high-reward choices. The learning curve plot demonstrates stability and long-term performance by displaying the moving average reward over the task sequence. In contrast to the orange line ("Without Clustering"), which varies at a considerably lower level (usually between 0.55 and 0.65), the green line ("With Clustering") continuously maintains a larger and more stable moving average reward (primarily above 0.75). This validates the long-term, and consistent improvement in performance.

The frequency with which each model selected a certain action (Local, L, or one of the ten UAVs, U1-U10) was compared in the action bar chart. The "With Clustering" model is represented by the green/blue bars, whereas the "Without Clustering" model is represented by the orange/gray bars. The variation in frequency is displayed by percentage labels. The local (L) action is strongly favored by the non-clustering model (more than 50% of the time). The clustering model, on the other hand, moves its decisions to individual UAVs, especially UAV 8 (U8), which experiences a +13.5% rise in selection frequency, and considerably lessens the reliance on the Local action (by -51.3%). This shows that a more complex, distributed, and specialized offloading policy is learned by the cluster-aware model.

The regret plot compares the distribution of regret (the difference between the optimal reward and the reward obtained by the model). Lower regret is better. The strategy "With Clustering" accomplished a percentage of 43.0% lower than the total regret achieved by the "Without Clustering" strategy. This is a strong measurable indicator that the clustering approach significantly improves the efficiency and optimality of offloading decisions.

In conclusion, the dashboard offers compelling evidence that the Cluster-Aware Contextual Bandit is a better and statistically validated strategy across several measures (mean reward, win rate, cumulative advantage, reward distribution, and regret).

## 4.4. Enhanced Contextual Bandit Algorithm Conversion

The purpose of the experimental assessment was to confirm that our suggested improved bandit algorithm is effective in controlling task offloading in the UAV-MEC architecture. This section specifically describes how the dual function of the algorithms making the best offloading choices for incoming workloads and dynamically ensuring load balance across the fleet of UAVs—is implemented and evaluated.

A compressive overview of the distribution and management of jobs in a system involving local processing, UAVs, and various 5G network slices is presented in Fig. 4. The allocation of jobs and performance features of offloading choices are the main topics of the dashboard.

The unified UAV offloading dashboard, Fig.5, illustrates a conservative resource management strategy where 86.7% of tasks are processed locally, resulting in a low overall offloading rate of 13.3%. The system exhibits highly dynamic resource allocation, with the majority of tasks concentrated in the latency-critical URLLC and high-bandwidth eMBB slices, and a resource allocation policy that frequently defaults to a 70% utilization threshold. Crucially, the dashboard reveals a specialized load-balancing approach: while the total task load is unevenly distributed among UAV-MECs, the UAV-MECs with lower overall volume are often tasked with a higher proportion of critical tasks that require offloading and more blockchain transaction, as evidenced by the highest Transaction Percentage (15.6%) UAV-MEC 3(in the next plot) being assigned to one of the least-busy UAV-MECs, confirming that the system prioritizes task criticality over simple volume-based load distribution.

The chart in Fig. 6 shows that *Manshift* offers the best overall UAV-MEC offloading performance among evaluated clustering algorithms, according to a unified performance metric that blends success rate, latency, and efficiency.

It provides a concise visual summary that helps researchers and engineers select the most effective algorithm for UAV-based edge computing systems.

## 4.5. Blockchain Experiment

The Ganache Ethereum blockchain was employed to establish a foundation of trust and security between the UAV-MEC server and network. Furthermore, the Truffle model is utilized for the essential tasks of compiling and deploying the necessary smart contracts onto this blockchain environment.

The proposed system utilizes two blockchain based smart contracts. The **UserAllocation** contract supports routine task allocation and communication with UAV-MEC nodes, whereas the **UserOffload** contract manages the offloading of computation-intensive tasks that exceed local energy and computational capabilities.

Although the gas analysis of the **UserOffloading** smart contract in Fig.7 is invoked less frequently than the **UserAllocation** contract, it exhibits higher total gas consumption. This is because task offloading involves computationally intensive on-chain logic, including feasibility checks, UAV selection, slice validation, and several state updates. In contrast, the resource allocation contract mainly performs lightweight confirmation and bookkeeping operations, resulting in lower gas cost per transaction. Therefore, fewer but more complex offloading transactions can dominate the total gas usage compared to more frequent but simpler allocation transactions.

Although though our suggested approach shows notable performance gains in dynamic UAV-MEC offloading, we recognize several drawbacks that offer obvious directions for further study. Our implementation employs a permissioned Ganache blockchain with proof-of-authority (PoA) consensus to reduce latency for the Blockchain Latency-Throughput Trade-off. This adds a centralization trade-off, but it is appropriate for the simulated scenario. More reliable but slower consensus procedures (such as Practical Byzantine Fault Tolerance) may be needed in a completely decentralized, multi-stakeholder 5G slice scenario, which may affect the capacity to fulfill sub-10ms URLLC latencies during peak transaction loads. Although analysis was thorough, the evaluation of the seven clustering algorithms was not meant for simultaneous real-time operation. Despite being event-triggered, re-clustering's computing cost increases with the number of devices N. The clustering overhead may constitute a bottleneck in ultra-dense mMTC scenarios with tens of thousands of units. To sustain efficiency on a large scale, research on lightweight feature extraction and incremental clustering methods is required.

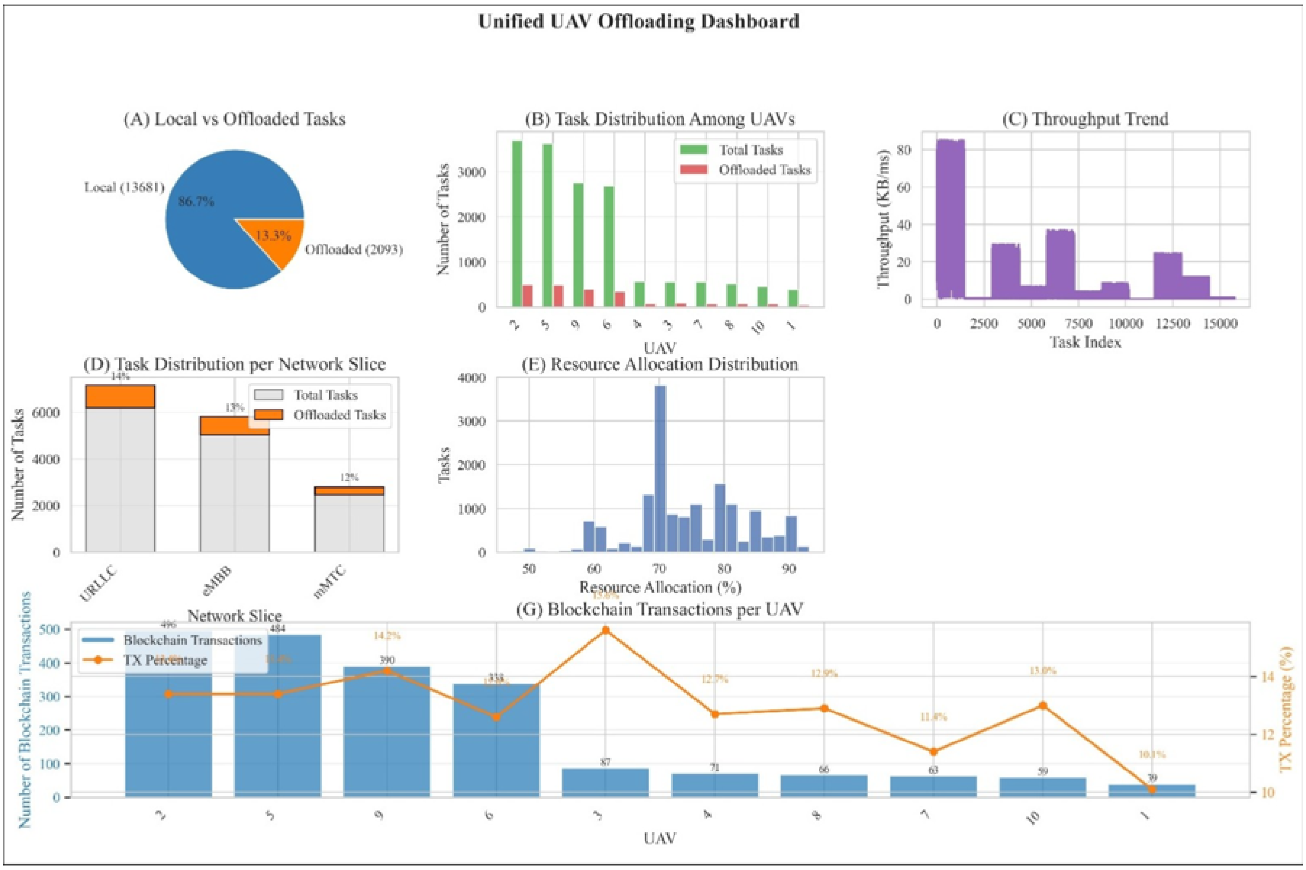


**Figure 5.** Unified offloading Dashboard

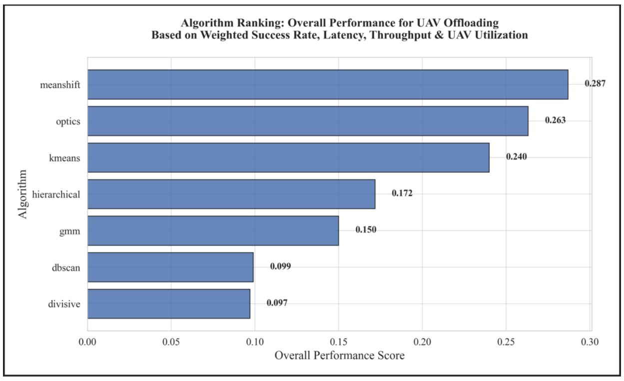


**Figure 6.** Algorithms ranking

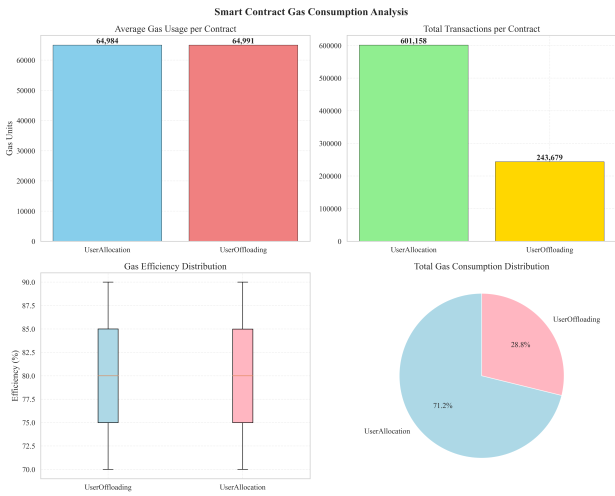


**Figure 7.** Smart contract gas consumption

## 5. CONCLUSION

This study shows that resource orchestration in UAV-MEC systems, enabling 5G network slicing, is fundamentally enabled by dynamic device clustering. We developed a self-adjusting control loop that handles heterogeneous QoS requirements while controlling computational overhead through the integrated workflow outlined in Section 3, where clustering metrics inform bandit decisions that trigger UAV assignments, which are then verified by blockchain contracts.

This study develops UAV-MEC offloading and resource allocation networks by representing dynamic device clustering, which serves as a fundamental enabler for resource orchestration in 5G network slicing systems. By integrating intelligent clustering algorithms with blockchain-enabled trust mechanisms, we created a model that addresses the critical challenges of heterogeneous service QoS and dynamic UAV-MEC resource constraints in edge systems.

The proposed smart contracts architecture represents an important improvement over traditional massive approaches, providing a clear connection between resource allocation and task offloading phases while maintaining end-to-end traceability. This architecture allows fine-grained SLA enforcement and automated accountability mechanisms that are crucial for multi-tenant edge environments.

Our experimental results validate the transformative impact of slice-aware clustering on the overall system performance. The substantial improvements in regret reduction, UAV reward optimization, and accuracy enhancement demonstrate that treating device clustering as a dynamic control mechanism rather than a static data organization technique fundamentally changes the efficiency of UAV-MEC systems. These improvements are particularly pronounced in scenarios with high service heterogeneity and stringent latency requirements, confirming the model's suitability for a diverse range of 5G use cases.

Specifically, our comparative evaluation of seven clustering algorithms revealed that the MeanShift algorithm delivered the best UAV offloading performance, achieving the highest overall performance score based on a weighted combination of the normalized success rate and latency efficiency. This finding confirms that adaptive, density-based clustering is particularly well-suited for dynamic UAV-MEC environments, where continuous spatial and demand variations require flexible region formation and resilience to noise.

Based on our findings, we foresee several interesting directions for future research. Future research should focus on developing predictive clustering models that employ machine learning to evaluate user movement patterns and service demands to enable proactive UAV deployment and resource allocation. The blockchain architecture can be extended to provide a safe cross-domain federation among several network operators by integrating reputation algorithms into smart contracts to promote equal resource contribution. As networks transition to 6G, our method must adapt to ultra-dense deployments with sub-millisecond latency constraints and integrate quantum-resistant cryptographic primitives for long-term security. Energy efficiency is another important area where clustering algorithms should be enhanced to account for the carbon footprint in addition to traditional QoS evaluations. This could be used with renewable energy harvesting methods to increase the duration of UAV operation.

**Authorship Contribution Statement**

Asrar Ahmed Baktayan: Conceptualization, Methodology, Software, Writing – original draft. Ammar Thabit Zahary: Supervision, Validation, Writing – review & editing. Ibrahim Ahmed Al-Baltah: Supervision, Formal analysis, Writing – review & editing.

**Declaration of competing interest**

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

**Data availability**

Data will be made available on request.

**Ethics**

This research does not involve human participants, animal subjects, clinical data, or biological materials. Therefore, ethical approval and informed consent were not required for this study

**Funding**

. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

## REFERENCES

[1] Q. Tang, Z. Fei, J. Zheng, B. Li, L. Guo, and J. Wang, "Secure aerial computing: Convergence of mobile edge computing and blockchain for uav networks," *IEEE Trans. on Veh. Technol.*, vol. 71, no. 11, pp. 12 073–12 087, 2022. DOI: 10.1109/tvt.2022.3189818.

[2] J. Dias, P. Pinto, R. Santos, and S. Malta, "5g network slicing: Security challenges, attack vectors, and mitigation approaches," *Sensors (Basel)*, vol. 25, no. 13, Jun. 2025. DOI: 10.3390/s25133940.

[3] S. M. Altowaijri and M. Ayari, "The synergistic impact of 5g on cloud-to-edge computing and the evolution of digital applications," *Mathematics*, vol. 13, no. 16, p. 2634, 2025. [Online]. Available: https://www.mdpi.com/2227-7390/13/16/2634.

[4] A. A. Baktayan, A. T. Zahary, and I. A. Al-Baltah, "A systematic mapping study of uav-enabled mobile edge computing for task offloading," *IEEE Access*, vol. 12, pp. 101 936–101 970, 2024. DOI: 10.1109/ACCESS.2024.3431922.

[5] B. A. Raddwan and I. A. Al-Baltah, "Quantify the joint effect of mobility and urban environment on computation offloading to multi-uav mec network: Sojourn time," *Ad Hoc Networks*, vol. 179, p. 104 019, 2025. DOI: https://doi.org/10.1016/j.adhoc.2025.104019.

[6] D. K. Nishad, V. R. Verma, P. Rajput, S. Gupta, A. Dwivedi, and D. R. Shah, "Adaptive ai-enhanced computation offloading with machine learning for qoe optimization and energy-efficient mobile edge systems," *Sci. Reports*, vol. 15, no. 1, p. 15 263, 2025. DOI: 10.1038/s41598-025-00409-4.

[7] Z. Yao, P. Chang, F. K. Khalil, and C. Duan, "Joint node selection and task offloading via evolutionary game and matd3 in uav-assisted mec networks," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 37, no. 8, p. 220, 2025. DOI: 10.1007/s44443-025-00248-3.

[8] M. A. Areqi, A. T. Zahary, and M. N. Ali, "Multi-objective clustering model for named data networking in manets," *IEEE Access*, vol. 13, pp. 205 545–205 557, 2025. DOI: 10.1109/ACCESS.2025.3638677.

[9] A. A. Wani, "Comprehensive analysis of clustering algorithms: Exploring limitations and innovative solutions," *PeerJ Comput. Sci*, vol. 10, e2286, 2024. DOI: 10.7717/peerj-cs.2286.

[10] S. Dash, A. U. Khan, S. K. Swain, and B. Kar, "Clustering based efficient mec server placement and association in 5g networks," in *2021 19th OITS International Conference on Information Technology (OCIT)*, 2021, pp. 167–172. DOI: 10.1109/OCIT53463.2021.00042.

[11] A. A. Al-awamy, N. Al-shaibany, A. Sikora, and D. Welte, "Hybrid consensus mechanisms in blockchain: A comprehensive review," *Int. J. Intell. Syst.*, vol. 2025, no. 1, p. 5 821 997, 2025. DOI: https://doi.org/10.1155/int/5821997.

[12] A. A. Al-awamy, N. Al-shaibany, and E. A. Alawamy, "A real-time hybrid consensus framework for iot blockchain networks using validator reputation, markov modeling, and priority queueing," *Sana'a Univ. J. Appl. Sci. Technol.*, vol. 3, no. 6, pp. 1277–1290, Dec. 2025. DOI: 10.59628/jast.v3i6.1751.

[13] D. Wang, Y. Jia, L. Liang, K. Ota, and M. Dong, "Resource allocation in blockchain integration of uav-enabled mec networks: A stackelberg differential game approach," *IEEE Trans. on Serv. Comput.*, vol. 17, no. 06, pp. 4197–4210, 2024. DOI: 10.1109/tsc.2024.3418330.

[14] M. A. et al., "Optimizing resource allocation and task offloading in multi-uav mec networks," *IEEE Access*, vol. 13, pp. 68 710–68 725, 2025. DOI: 10.1109/ACCESS.2025.3562102.

[15] S. Xu, Q. Liu, C. Gong, and X. Wen, "Energy-efficient multi-agent deep reinforcement learning task offloading and resource allocation for uav edge computing," *Sensors*, vol. 25, no. 11, p. 3403, 2025. [Online]. Available: https://www.mdpi.com/1424-8220/25/11/3403.

[16] S. Sai, S. Mishra, and V. Chamola, "Resource allocation in unmanned aerial vehicle networks: A review," *Veh. Commun.*, vol. 52, p. 100 889, 2025. DOI: https://doi.org/10.1016/j.vehcom.2025.100889.

[17] M. Wirth, A. Ortiz, and A. Klein, "Risk-aware bandits for digital twin placement in non-stationary mobile edge computing," in *2024 IEEE International Conference on Communications Workshops (ICC Workshops)*, 2024, pp. 13–18. DOI: 10.1109/ICCWorkshops59551.2024.10615914.

[18] S. Zhang, M. Xiao, G. Gao, Y. Xu, and H. Sun, "Offloading tasks to unknown edge servers: A contextual multi-armed bandit approach," in *IEEE INFOCOM 2023 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2023, pp. 1–6. DOI: 10.1109/INFOCOMWKSHPS57453.2023.10226047.

[19] T. Chen, S. Wang, X. Fan, X. Zhang, C. Luo, and Y. Hong, "Uav-assisted multi-object computing offloading for blockchain-enabled vehicle-to-everything systems," *Comput. Mater. Continua*, vol. 81, no. 3, pp. 3927–3950, 2024. DOI: https://doi.org/10.32604/cmc.2024.056961.

[20] J. Chen, Z. Kuang, Y. Zhang, S. Lin, and A. Liu, "Blockchain-enabled computing offloading and resource allocation in multi-uavs mec network: A stackelberg game learning approach," *Trans. Info. For. Sec.*, vol. 20, pp. 3632–3645, 2025. DOI: 10.1109/tifs.2025.3550812.

[21] Y. Su, Z. Gao, X. Du, and M. Guizani, "User-centric base station clustering and resource allocation for cell-edge users in 6g ultra-dense networks," *Future Gener. Comput. Syst.*, vol. 141, pp. 173–185, 2023. DOI: https://doi.org/10.1016/j.future.2022.11.011.

[22] R. Samir, H. El-Hennawy, and H. Elbadawy, "Cluster-based multi-user multi-server caching mechanism in beyond 5g/6g mec," *Sensors*, vol. 23, no. 2, p. 996, 2023. [Online]. Available: https://www.mdpi.com/1424-8220/23/2/996.

[23] V. Rodríguez-Fernández, H. D. Menéndez, and D. Camacho, "A study on performance metrics and clustering methods for analyzing behavior in uav operations," *J. Intell. & Fuzzy Syst.*, vol. 32, no. 2, pp. 1307–1319, 2017. DOI: 10.3233/jifs-169129.

[24] R. Xing, Z. Su, T. H. Luan, Q. Xu, Y. Wang, and R. Li, "Uavs-aided delay-tolerant blockchain secure offline transactions in post-disaster vehicular networks," *IEEE Trans. on Veh. Technol.*, vol. 71, no. 11, pp. 12 030–12 043, 2022. DOI: 10.1109/TVT.2022.3184965.

[25] A. Miglani, N. Kumar, A. Kishore, and N. Mohammad, "Uav-enabled edge computing and blockchain based secure charging station selection for energy trading in v2g environment," in *Proceedings of the 5th International ACM Mobicom Workshop on Drone Assisted Wireless Communications for 5G and Beyond*, Sydney, NSW, Australia, 2022. [Online]. Available: https://doi.org/10.1145/3555661.3560872.

[26] F. AlHrazi, M. Algabri, and A. A. Al-Khulaidi, "Hierarchical blockchain as line defense of attacks to messages propagation in vanet," *Sana'a Univ. J. Appl. Sci. Technol.*, vol. 1, no. 3, Sep. 2023. DOI: 10.59628/jast.v1i3.370.

[27] Y. Xu, H. Xu, X. Chen, H. Zhang, B. Chen, and Z. Han, "Blockchain-based ar offloading in uav-enabled mec networks: A trade-off between energy consumption and rendering latency," *IEEE Trans. on Veh. Technol.*, vol. 74, no. 11, pp. 18 065–18 080, 2025. DOI: 10.1109/TVT.2025.3581015.

[28] A. A. Khan, A. A. Laghari, M. Shafiq, S. A. Awan, and Z. Gu, "Vehicle to everything (v2x) and edge computing: A secure lifecycle for uav-assisted vehicle network and offloading with blockchain," *Drones*, vol. 6, no. 12, p. 377, 2022. [Online]. Available: https://www.mdpi.com/2504-446X/6/12/377.

[29] M. Song, C. Li, J. Ye, X. Gong, and Y. Luo, "Efficient consensus algorithm based on improved dpos in uav-assisted mobile edge computing," *Comput. Commun.*, vol. 207, pp. 86–99, 2023. DOI: 10.1016/j.comcom.2023.05.008.

[30] M. F. Shaikh, S. H. Hassan, A. Maccaro, G. Pratesi, and D. Piaggio, "A blockchain framework using proof of authority and smart contracts for ethical and secure healthcare asset management," *Front. Public Health*, 2025. DOI: 10.3389/fpubh.2025.1638546.

[31] A. A. Almazroi, M. A. Alqarni, M. A. Al-Shareeda, M. H. Alkinani, A. A. Almazroey, and T. Gaber, "Fca-vbn: Fog computing-based authentication scheme for 5g-assisted vehicular blockchain network," *Internet Things*, vol. 25, p. 101 096, 2024. DOI: https://doi.org/10.1016/j.iot.2024.101096.

[32] A. A. Almazroi, E. A. Aldhahri, M. A. Al-Shareeda, and S. Manickam, "Eca-vfog: An efficient certificateless authentication scheme for 5g-assisted vehicular fog computing," *PLOS ONE*, vol. 18, no. 6, e0287291, 2023. DOI: 10.1371/journal.pone.0287291.

[33] A. A. Almazroi, M. A. Alqarni, M. A. Al-Shareeda, and S. Manickam, "L-cppa: Lattice-based conditional privacy-preserving authentication scheme for fog computing with 5g-enabled vehicular system," *PLOS ONE*, vol. 18, no. 10, e0292690, 2023. DOI: 10.1371/journal.pone.0292690.

[34] A. A. Almazroi, M. H. Alkinani, M. A. Al-Shareeda, and S. Manickam, "A novel ddos mitigation strategy in 5g-based vehicular networks using chebyshev polynomials," *Arab. J. for Sci. Eng.*, vol. 49, no. 9, pp. 11 991–12 004, 2024. DOI: 10.1007/s13369-023-08535-9.

[35] N. Y. Al-Matari and A. T. Zahary, "A conceptual blockchain architecture for dynamic spectrum access in 6g cr-iot networks: An analytical review," *Sana'a Univ. J. Appl. Sci. Technol.*, vol. 3, no. 6, pp. 1467–1480, 2025. DOI: 10.59628/jast.v3i6.1953.

[36] Y. Yang, Y. Shi, X. Cui, J. Li, and X. Zhao, "A hybrid decision-making framework for uav-assisted mec systems: Integrating a dynamic adaptive genetic optimization algorithm and soft actor–critic algorithm with hierarchical action decomposition and uncertainty-quantified critic ensemble," *Drones*, vol. 9, no. 3, p. 206, 2025. [Online]. Available: https://www.mdpi.com/2504-446X/9/3/206.

[37] C. R. Lin and M. Gerla, "Adaptive clustering for mobile wireless networks," *IEEE J. on Sel. Areas Commun.*, vol. 15, no. 7, pp. 1265–1275, 1997. DOI: 10.1109/49.622910.

[38] S. Alhelaly, A. Muthanna, and I. A. Elgendy, "Optimizing task offloading energy in multi-user multi-uav-enabled mobile edge-cloud computing systems," *Appl. Sci.*, vol. 12, no. 13, p. 6566, 2022. [Online]. Available: https://www.mdpi.com/2076-3417/12/13/6566.

[39] G. Yang, L. Hou, X. He, D. He, S. Chan, and M. Guizani, "Offloading time optimization via markov decision process in mobile-edge computing," *IEEE Internet Things J.*, vol. 8, no. 4, pp. 2483–2493, 2021. DOI: 10.1109/JIOT.2020.3033285.

[40] F. Mehmeti, S. Ayvaşık, F. Kaynar, T. F. L. Porta, and W. Kellerer, "Constant playout rates: Resource allocation for improved user experience with live video streaming in 5g," *Comput. Networks*, vol. 269, p. 111 481, 2025. DOI: https://doi.org/10.1016/j.comnet.2025.111481.

[41] M. Baygin, O. Yaman, N. Baygin, and M. Karakose, "A blockchain-based approach to smart cargo transportation using uhf rfid," *Expert Syst. with Appl.*, vol. 188, p. 116 030, 2022. DOI: https://doi.org/10.1016/j.eswa.2021.116030.

[42] D. S. Nireeksha, "Decentralized online voting system using blockchain for secure and transparent elections," *Int. J. Sci. Innov. Eng.*, vol. 2, no. 9, pp. 682–694, 2025. DOI: 10.70849/IJSCI.