



OPEN Cooperative UAV swarms for zero knowledge verification of edge generative AI using trust-aware multiagent learning

Kuldashbay Avazov¹, Kudratjon Zohirov², Umidjon Ruziev³, Nabijon Abduazizov⁴, Akmalbek Abdusalomov^{5,6}, Abror Buriboev^{7,8} & Heung Seok Jeon⁹✉

The high-level integration of generative artificial intelligence (AI) in edge computing systems has raised the question of the integrity and reliability of deploying Model-as-a-Service. Edge servers are not required to follow the so-called generative model to minimize computational cost, whereas users and service providers want validation mechanisms that do not compromise proprietary model information. To address this challenge, this study proposes a cooperative unmanned aerial vehicle (UAV)-swarm-enabled zero-knowledge verification framework for secure, privacy-preserving verification of edge-based generative artificial intelligence inference. The proposed framework involves edge servers producing an interactive cryptographic zero-knowledge proof to verify the execution of generative AI, and UAV swarms that fly freely to confirm verification operations, subject to mobility and energy constraints. The age of verification metric is proposed to trust verification information, jointly reflecting the unverified server reliability and verification freshness, and to provide dynamic priority to risky edge servers. To effectively plan the behaviour of a UAV swarm, a trust-based multi-agent reinforcement learning approach is developed that enables decentralized decision-making while training is centralized. Extensive simulation results show that the proposed framework significantly improves the state-of-the-art baseline schemes in verification timeliness, malicious server detection delay, energy efficiency, and scalability. The findings validate that integrating cooperative UAV swarms, trust-aware verification, and multi-agent learning is an efficient approach to providing reliable generative AI services in dynamic edge computing environments.

Keywords Generative artificial intelligence, Zero-knowledge proofs, Cooperative UAV swarms, Trust-aware verification, Multiagent reinforcement learning, Edge computing, Model-as-a-service

Generative artificial intelligence (AI) is a fundamentally enabling technology within an expanding ecosystem of smart applications, including smart transportation, smart urban sensing, smart surveillance, and intelligent, real-time, edge-based decision-making^{1–5}. As state-of-the-art generative AI models have proliferated in recent years, advanced Model-as-a-Service (MaaS) models are now deployed on edge servers to provide low-latency inference for resource-constrained end users^{6,7}. Although this means of deployment makes a system more responsive and susceptible to scale, because of this deployment model, a basic issue of trust is presented, as far as users and third-party confirmers must be in a position to ensure that an edge server is faithfully implementing an asserted generative AI model, without stating proprietary model parameters or the internal

¹Department of Computer Engineering, Gachon University, Sujeong-gu, Seongnam-si 13120, Gyeonggi-do, Republic of Korea. ²Department of Software and Technical Support of Computer Systems, Karshi State Technical University, Karshi 180100, Uzbekistan. ³Department of Automation of Production Processes, Tashkent State Technical University, Tashkent 100095, Uzbekistan. ⁴Navoi State University of Mining and Technologies, Galaba Avenue 76V, Navoi 210100, Uzbekistan. ⁵Department of Artificial Intelligence, Tashkent University of Information Technologies Named after Muhammad Al-Khwarizmi, Tashkent 100200, Uzbekistan. ⁶Department of Information Processing and Management Systems, Tashkent State Technical University, Tashkent 100095, Uzbekistan. ⁷Department of Software Engineering, Samarkand State University, Samarkand, Uzbekistan 140100. ⁸Department of Exact Sciences, Kimyo International University in Tashkent, Samarkand 140100, Uzbekistan. ⁹Department of Computer Engineering and Also with AI Convergence Research Center, Konkuk University, Chungju, Republic of Korea. ✉email: hsjeon@kku.ac.kr

computation details^{8,9}. Unmanned aerial vehicles (UAVs) have such an advantageous role in this issue because of their mobility, flexible deployment, and opportunity to cover the dynamic environments on demand. UAVs respond by pushing the edge servers in geographically dispersed, fast-changing environments, unlike fixed infrastructure¹⁰. Besides, UAVs offer line-of-sight communication, low latency, and groundbreaking scalability, which makes them suitable to aid verification assignments within edge AI systems, in which reliability and responsiveness play a key role.

The concept of zero-knowledge proofs (ZKPs) has become one of the strongest cryptographic means of proving the correctness of computation without any disclosure of sensitive internal data, and thus they are useful in privacy-preserving validation of edge-deployed AI inference^{11,12}. In this regard, ZKPs allow an edge server to cryptographically prove that a generative AI inference has been successfully run without revealing any proprietary model parameters or intermediate calculations^{13,14}. Nevertheless, and in spite of these high-security assurances, ZKP-based verification cannot be applied or utilized in real edge conditions because of limited computing resources, energy availability, and due to a changing communication environment.

Although the problem of UAV-assisted verification of edge AI services has recently been investigated, several important limitations remain. Currently, most solutions assume UAV operations on an individual basis and ignore the opportunities of coordinated swarm behavior¹⁵. Such a lack of cooperation may lead to unnecessary verification activities, unnecessary energy use, and delayed verification^{16,17}. Second, current verification scheduling schemes are primarily motivated by freshness-based measures and do not leverage the non-uniform reliability of edge servers. In realistic MaaS environments, edge servers can experience varying levels of reliability due to model updates, partial attacks, or adversarial behavior; therefore, trust-aware verification models are implemented^{18,19}. Third, the joint dynamics of UAV motion, energy constraints, and monitoring schedules render solutions based on conventional optimal control methods insufficient, underscoring the need for intelligent, responsive control systems²⁰.

Multi-agent reinforcement learning (MARL) is a promising approach to distributed decision-making in complex, dynamic systems^{21,22}. MARL enables UAV swarms to optimize verification timeliness, energy efficiency, and coverage by having agents learn cooperative policies through interaction with the environment^{23,24}. Also, by incorporating trust awareness into the learning framework, UAVs can prioritize verification tasks not only based on time freshness but also on the evolving credibility of edge servers, resulting in better, more reliable AI service provisioning.

Based on these findings, this study proposes a joint UAV swarm-based zero-knowledge check system of edge-deployed AI services, where edge servers produce noninteractive zero-knowledge proofs to define a correct AI inference, and multiple UAVs are used as mobile validators that organize the verification in the presence of mobility and energy limitations. It proposes a trust-conscious Age of Verification measurement to simultaneously model verification freshness and server reliability, and presents a trust-conscious multi-agent reinforcement learning policy to assist in scalable, efficient coordination of UAVs. This combined design fills a critical gap in the existing literature, where ZKP-based AI verification is most commonly restricted to centralized systems, UAV-aided verification is more likely to be uncoordinated and lack trust, and MARL-based UAV approaches are usually concerned with communication or task allocation, but not security or verification stability. The proposed solution can minimize unnecessary verification and coordination of cooperative UAVs, as well as trust-based learning into a single model by utilizing privacy-safe verification, critical node server prioritization, and enhancing the trustworthiness of models in dynamic edge settings.

The key contributions to the current study are as follows:

- This study proposes a collaborative UAV swarm-based zero-knowledge verification system for edge-deployed generative AI services, enabling privacy-preserving verification of such models without disclosing proprietary model parameters.
- This work presents a trust-sensitive Age of Verification (TA-AoV) metric that quantifies verification freshness and the degree of trustworthiness change in edge servers, enabling the system to dynamically prioritize high-risk, unreliable edge servers.
- To overcome these joint problems of UAV mobility, energy, and dynamic verification requirements, this study establishes a trust-based multi-agent reinforcement learning model in centralized training and decentralized implementation to achieve efficient UAV swarm coordination.
- This study develops a unified system model that concurrently addresses zero-knowledge proof verification, UAV mobility, energy usage, and trust development to create a scalable, practical edge AI verification model.
- A detailed simulation shows that the proposed framework achieves significant improvements in trust-aware verification timeliness, malicious server detection delay, energy efficiency, and scalability compared with state-of-the-art baseline schemes.

The rest of this paper is organized as follows. Section "[Related work](#)" discusses sorts of related literature regarding zero-knowledge verification of artificial intelligences, UAV-based edge intelligence, and multi-agent learning of the collaboration between UAVs. Section "[System model and problem formulation](#)" illustrates the problem formulation and system model. Section "[Zero knowledge verification framework for edge generative AI](#)" describes the suggested framework of zero-knowledge verification and trust-aware verification utility. Section "[Trust aware multiagent learning for cooperative UAV swarms](#)" explains the network based multiagent learning model in swarm control of UAVs. Section "[Performance evaluation and simulation results](#)" assesses the performance of the suggested framework using simulations, and Sect. "[Conclusion](#)" provides a conclusion of the paper.

Related work

It has become common in recent years to ensure the integrity of artificial intelligence inference within the Model-as-a-Service paradigm. Zero-knowledge proofs were explored in early studies as a general-purpose cryptographic protocol for verifiably computing, allowing a prover to demonstrate that a computation was performed without disclosing any sensitive data^{25,26}. These foundational works laid the theoretical groundwork for noninteractive verification systems, including zk-SNARKs and zk-STARKs, which later enabled real-world applications of zero-knowledge verification.

Building on these, later studies have considered how zero-knowledge proofs can be used in machine learning inference, specifically, how neural network execution can be verified in a privacy-preserving fashion^{27,28}. Specifically, it used arithmetic circuit representations and Rank-1 Constraint System to confirm the accuracy of convolutional and fully connected neural networks. However, as explained in Refs²⁹. and³⁰, such strategies are generally very expensive in terms of computing and only suited to centralized or cloud-based contexts. Furthermore, most of the literature focuses on isolated inference and does not account for the distributed, time-dependent, and resource-limited characteristics of edge-deployed generative AI services^{31,32}. Simultaneously, UAVs have been extensively explored as mobile edge nodes and communication relays due to their flexibility and rapid deployability. In Refs³³. and³⁴, UAV-assisted edge computing architectures were developed to enable task offloading, information gathering, and low-latency service delivery under fluctuating conditions. Similarly, UAV mobility has been exploited to increase the coverage and resilience of emergency communications and to support use in smart cities³⁵. In addition, UAVs have been used as mobile monitors/auditors to enhance security and trust in distributed systems, such as for infrastructure inspection and anomaly detection³⁶.

More recent works have investigated UAV-assisted verification and security, in which UAVs serve as mobile sensors to detect malicious activity within edge networks³⁷. However, as noted in Ref³⁸., most existing UAV-assisted verification systems assume that UAVs operate independently, with each UAV making uncoordinated local decisions. The common effect of this assumption is that verification actions are often superfluous, energy use is inefficient, and verification latency is high, especially in large-scale networks with many edge servers. Timeliness in verification has also been identified as an important performance measure in a dynamic edge environment. Based on the Age of Information concept, freshness-sensitive measures were proposed to assess the staleness of verification or trust information^{39,40}. However, these measures primarily focus on time freshness and disregard the heterogeneous credibility of service providers. As highlighted in Ref⁴¹., edge servers can exhibit varying reliability during model updates, partial compromise, or adversarial manipulation, which makes it difficult to simply verify them using time-based approaches for risky decisions.

Multi-agent reinforcement learning has been widely studied to address decision-making challenges in such dynamic and distributed systems. MARL frameworks were used in Refs⁴². and⁴³ to plan trajectories for cooperative UAV networks and to allocate tasks and resources. The use of actor-critic-based algorithms was also explored to address the continuous state and action spaces characteristic of UAV systems^{44,45}. Moreover, the decentralized implementation of training, previously centralized, has proven effective for the scalable coordination of various agents^{46,47}. However, most available MARL-compliant UAV studies are designed with either communication efficiency or task completion as their primary goals and do not specify security-related goals, such as verification timeliness or trust enforcement.

The secure data management and communications support in distributed systems, especially in IoT- and cloud-oriented infrastructures, has also been explored in recent studies. An example is sensor data management systems enhanced with blockchains to improve the integrity, traceability, and reliability of engineering monitoring data deployed in a distributed IoT setup^{48,49}. It has also established secure cloud networking that enhances data protection and improves the deployment efficiency of industry-specific cloud platforms⁵⁰. Moreover, the need to manage cellular network interference efficiently further underscores the importance of communication reliability and efficiency in coordinating large-scale distributed systems^{2–5,51}. Though these studies do not directly address privacy-preserving edge AI service verification, they offer complementary perspectives on secure and efficient system design and provide additional incentives to propose combined frameworks, such as the one developed in the current study.

The current literature offers significant improvements over zero-knowledge verification, UAV-aided edge system, and multi-agent reinforcement learning, although those streams are mostly studied independently: ZKP schemes emphasize centralized verification with no mobility or resource limitations, UAV-based solutions address communication and task allocation but do not propose cryptographic validation, and MARL systems focus on coordination without the explicit incorporation of trust or security goals. This means that the joint provision of privacy-preserving verification, trust dynamics, cooperative UAV coordination, and energy-efficient operation in realistic edge MaaS settings is under-explored. To overcome this gap, the paper will merge zero-knowledge verification, cooperative UAV swarms, a trust-based verification metric, and multi-agent learning into a unified architecture of secure, scalable, and efficient edge AI verification.

System model and problem formulation

This section presents the system architecture, threat and trust models, the zero-knowledge verification process, and develops the cooperative UAV swarm verification problem.

System architecture

We assume a UAV-based edge intelligent network comprising four main elements: a cloud server, a group of edge servers, a cooperative swarm of UAVs, and end users, as shown in Fig. 1. Let $\mathcal{E} = \{1, 2, \dots, E\}$ be the collection of edge servers, and each edge server has a generative AI model and gives inference services according to the Model-as-a-Service paradigm. The cloud server serves as the global coordinator, handling system initialization,

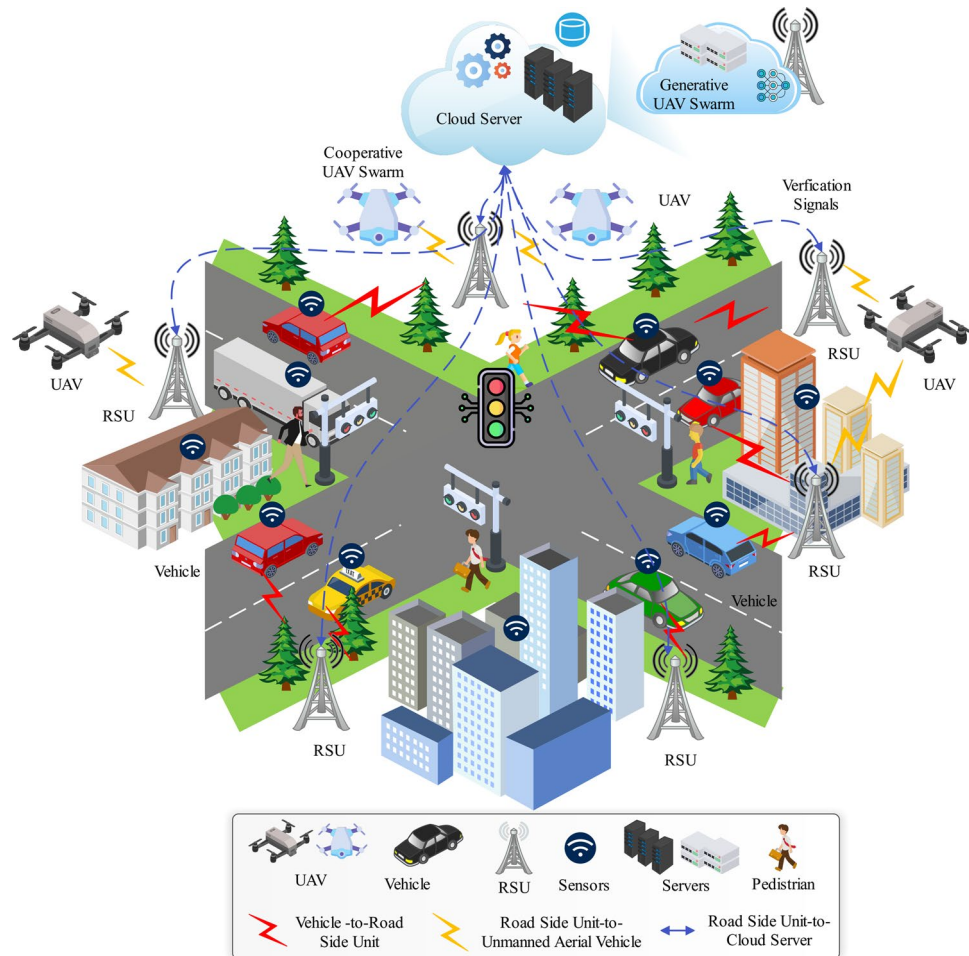


Fig. 1. Proposed system architecture of the DCI framework illustrating cloud, edge, and vehicular interactions.

policy changes, and long-term data storage, but it is not involved in real-time verification processes. In order to make verifications mobile and on demand, a swarm of UAVs, denoted by $\mathcal{U} = \{1, 2, \dots, U\}$, is sent throughout the service area. Every UAV provides communication, computation, and mobility capabilities, allowing it to move dynamically around the server's perimeter, gather evidence related to proofs, and perform local authentication of zero-knowledge proofs. The system is discrete-time, meaning time is divided into discrete slots indexed by $t = 1, 2, \dots$. The authenticated encryption mechanisms are assumed to provide message confidentiality as well as integrity so that all communications between UAVs, edge servers, and the cloud infrastructure are guaranteed. In particular, digital signatures are used to authenticate the source of messages, preventing spoofing and impersonation attacks, while encryption protects against eavesdropping and message tampering. These prevent adversaries of these measures being able to alter or replay proof transmissions and verification results.

Threat and trust model

We assume a heterogeneous adversarial model in this piece of work that represents realistic UAV-assisted edge settings. Edge servers have mostly been modeled as semi-honest adversaries, i.e. they can corrupt the desired generative AI model whilst adhering to the validation checks, so as not to be detected.

UAVs are presumed to be trusted-but-vulnerable, i.e. the assumption that they will perform the verification tasks correctly, but be compromised or spoofed, or manipulated by adversaries, in real-world scenarios. Our one-way adversarial framework does not presume explicit collusion among UAVs, but, as an extension, we treat opposing adversarial behavior, such as collusion, as a threat and discuss it. It is assumed that the communication infrastructure operates over authenticated and encrypted channels, under standard cryptographic assumptions, ensuring message integrity and confidentiality. The cloud server is considered a trusted coordinator responsible for system startup and long-term state management. It is assumed that edge servers are semi-honest, meaning they can violate the stated generative AI model by providing a compressed or altered model, but also adhere to the verification protocol to avoid detection. By contrast, the UAVs and the cloud server are assumed to be trusted parties. Each edge server $i \in \mathcal{E}$ is associated with a time-varying trust score, as defined in Eq. (1).

$$\tau_i(t) \in [0, 1]. \quad (1)$$

The trust score advocates the previous credibility of an edge server, with the highest value signifying a more dependable server and the lowest a less dependable one. It is dynamic, meaning it is updated after a check is performed; a successful verification increases trust, whilst a verification failure decreases it.

Although the existing model presumes that UAVs are completely trusted forces that are tractable, this may not be the case in an adversarial setting. Also, in real-world applications, UAVs can be compromised, spoofed, or colluded with. It is a crucial direction for future work to extend the framework to include adversarial UAV models or the behavior of Byzantine and distributed trust-validation mechanisms. Likewise, the semi-honest assumption about edge servers makes analysis easier, though it can be generalized to completely malicious adversaries that can deviate from protocol execution. These implications point to potential areas of improvement that could make the proposed framework more robust.

Within real-world applications, some of the threats that UAV-assisted systems might encounter include broken communication connections, the breakdown of UAV hardware, and environmental interference. Although the elements mentioned are only abstracted in the existing model, the proposed framework can be further expanded to include the fault tolerance measures, adaptive communication protocols, and redundancy schemes to guarantee the sustainable functioning at the conditions of the real world. To be consistent, we define two types of behavior: semi-honest and malicious. Semi-honest edge servers are honest to the protocol but might seek to optimize computation, meaning malicious edge servers can deliberately not adhere to the claimed model or produce invalid outputs.

Within the existing model, it is the assumptions about the model parameters that give rise to proofs, but a version-binding or explicit model commitment mechanism is not enforced. Realistically, this can be done through cryptographic commitments or hash-based model identifiers to tie knowledge of the model to a particular model version. Such mechanisms would also enhance the integrity guarantees and are a significant step towards future work.

Zero-knowledge verification process

When a UAV u is assigned to check an edge server i at a particular time slot t , the process is as follows. The edge server will first perform generative AI inference and create a zero-knowledge proof that the model execution is correct. This evidence is then sent to the specified UAV, which validates it locally without access to the underlying model parameters or internal calculations. Lastly, the verification result is sent to the cloud server and used to update the trust state of the relevant edge server. To formally model the verification decision, a binary decision variable is defined in Eq. (2).

$$x_{u,i}(t) = \begin{cases} 1, & \text{if UAV } u \text{ verifies edge server } i \text{ at time } t, \\ 0, & \text{otherwise,} \end{cases} \quad (2)$$

where these variable indicates whether UAV u is assigned to verify edge server i at time slot t . To ensure feasible and non-redundant verification assignments, scheduling constraints are imposed as specified in Eqs. (3) and (4).

$$\sum_{i \in \mathcal{E}} x_{u,i}(t) \leq 1, \quad \forall u \in \mathcal{U}, \quad (3)$$

$$\sum_{u \in \mathcal{U}} x_{u,i}(t) \leq 1, \quad \forall i \in \mathcal{E}. \quad (4)$$

These constraints ensure that each UAV can check at most one edge server per time slot and that each edge server can be checked by at most one UAV per time slot, preventing redundant verification and efficient use of UAVs.

Trust-aware age of verification

To jointly capture verification freshness and server trustworthiness, the Trust-Aware Age of Verification (TA-AoV) for edge server i at time slot t is defined as shown in Eq. (5).

$$A_i(t) = \alpha \cdot a_i(t) + \beta \cdot (1 - \tau_i(t)), \quad (5)$$

where $a_i(t)$ denotes the conventional age of verification, defined as the number of time slots elapsed since the most recent successful verification of edge server i , and $\tau_i(t)$ represents the trust score associated with that server. The weighting coefficients α and β satisfy $\alpha + \beta = 1$ and control the relative importance of verification freshness and trustworthiness. With this formulation, the TA-AoV increases when verification becomes stale or when an edge server's trust level decreases, thereby enabling risk-aware prioritization of verification tasks. The temporal evolution of the age of verification is governed by the update rule specified in Eq. (6).

$$a_i(t+1) = \begin{cases} \delta \cdot a_i(t), & \text{if } \sum_{u \in \mathcal{U}} x_{u,i}(t) = 1, \\ a_i(t) + 1, & \text{otherwise.} \end{cases} \quad (6)$$

The update rule is only acted upon when a successful verification is made, i.e. when the verification result is correct. Particularly, when at least one UAV manages to check edge server i during time slot t ($r_i(t) = 1$), the verification age reduces by a factor δ . Otherwise, the age increases linearly. This makes the Age of Verification consistent with the time since the last successful verification, avoiding excessive temporal concordance with its definition or overestimating the freshness of verification.

In order to be robust to the trust model, one should take into consideration the main attacks to manipulate the trust model, which encompass Sybil attacks, collusion, and abuse of rewards. In the formulation used today, the trust scores are updated based on verifiable results from the zero-knowledge proof validation and are thus subject to arbitrary manipulation within a limit. Nevertheless, in more antagonistic environments, additional protection schemes may be necessary, such as secure aggregation, identity checks, and reputation filtering mechanisms to guard against malevolent participants aiming to man-in-the-middle to increase or decrease trust ratings. The extensions represent a significant future direction for the work.

UAV mobility and energy model

Each UAV u is characterized by a two-dimensional position vector $p_u(t)$ at time slot t . The mobility of each UAV between consecutive time slots is constrained by a maximum velocity, as specified in Eq. (7).

$$\|p_u(t+1) - p_u(t)\| \leq v_{\max}, \quad (7)$$

where this constraint ensures physically feasible UAV motion by limiting the distance traveled within a single time slot. The total energy consumption of UAV u at time slot t is modeled as the sum of flight, communication, and computation energy components, as defined in Eq. (8).

$$E_u(t) = E_u^{\text{fly}}(t) + E_u^{\text{comm}}(t) + E_u^{\text{ver}}(t), \quad (8)$$

where $E_u^{\text{fly}}(t)$ represents the energy consumed for UAV propulsion and movement, $E_u^{\text{comm}}(t)$ denotes the energy required for wireless communication, and $E_u^{\text{ver}}(t)$ corresponds to the computational energy expended in verifying zero-knowledge proofs. To ensure sustainable UAV operation, each UAV is subject to a long-term average energy constraint, as expressed in Eq. (9).

$$\limsup_{T \rightarrow \infty} \frac{1}{T} \sum_{t=1}^T E_u(t) \leq E_u^{\max}. \quad (9)$$

This constraint ensures that each UAV's long-term average energy consumption does not exceed its available energy budget, thereby preventing premature energy depletion and enabling persistent verification operations.

Problem formulation

This work aims to develop a policy that will verify the presence of cooperation of UAV swarm such that the long-term average Trust-Aware Age of Verification (TA-AoV) at all work edge servers is minimized, with the policy implementing scheduling and mobility decisions as well as meeting the scheduling, mobility, and long-term energy constraints.

$$\min_{\{x_{u,i}(t), p_u(t)\}} \limsup_{T \rightarrow \infty} \frac{1}{T} \sum_{t=1}^T \sum_{i \in \mathcal{E}} A_i(t), \quad (10)$$

$$\sum_{i \in \mathcal{E}} x_{u,i}(t) \leq 1, \quad \forall u \in \mathcal{U}, t, \quad (11)$$

$$\sum_{u \in \mathcal{U}} x_{u,i}(t) \leq 1, \quad \forall i \in \mathcal{E}, t, \quad (12)$$

$$\|p_u(t+1) - p_u(t)\| \leq v_{\max}, \quad \forall u \in \mathcal{U}, t, \quad (13)$$

$$\limsup_{T \rightarrow \infty} \frac{1}{T} \sum_{t=1}^T E_u(t) \leq E_u^{\max} \quad \forall u \in \mathcal{U}. \quad (14)$$

The objective function in Eq. (10) minimizes the long-term average TA-AoV across all edge servers, taking into account verification freshness and server reliability. The time constraints in both Eq. (11) and Eq. (12) are to make sure that each UAV can verify no more than one edge server during a time slot, and edge servers can be verified no more than once by a UAV in order to avoid sharing the same Verification assignment. The principal mobility constraint in Eq. (13) restricts the movement of every UAV between successive time intervals, which provides physically realistic UAV movement. Lastly, the long-term energy constraint in Eq. (14) ensures that each UAV's average energy use is below its energy budget, thereby ensuring long-term operation is sustainable. The trust evolution process is stochastic; it is time-coupled, and UAV decision variables are strongly coupled, which makes the optimization problem non-convex and NP-hard. Therefore, a solution based on multi-agent reinforcement learning is adopted.

To guarantee reproducible and clear results, the general framework is based on a four-step sequential pipeline: (i) system initialization: in this stage, trust scores, UAV states, and model parameters are initialized; (ii) verification scheduling: at this stage, UAVs choose target edge servers with the help of local observations and learned policies; (iii) zero-knowledge proofs generation and verification, where edge servers make proofs and UAVs verify them but data regarding them is not accessed by internals of the model; and (iv) trust

and state update, where the result of verification changes the trust and informs future decision-making. All simulations were carried out in a discrete-time setup, with fixed time slots, often similar to real-world settings, and the MARL architecture was trained under a centralized training and decentralized execution paradigm. Hyperparameters such as learning rates, discount factors, and network architectures are clearly stated in Section 6 to enable reproducibility.

It is notable that the existing formulation of the optimization problem can represent a variety of real-world constraints, which are not detailed: communication bandwidth, wireless interference, and hard (latency) constraints. This would involve more modeling of channel dynamics, network congestion, and stochastic traffic patterns, while accounting for such Quality-of-Service (QoS) restrictions. Moreover, the adversarial attack dynamics are not explicitly represented as stochastic processes using the existing framework. Rather, their implications are inferred indirectly through trust score development. A direction of interest in the future is to extend the formulation to a probabilistic attack model and communication-aware constraints.

Zero knowledge verification framework for edge generative AI

In this section, we introduce our proposed zero-knowledge verification system for generative AI models, which operates on an edge device and enables cooperative take-off UAV swarms to perform secure, privacy-sensitive verification without knowing the model parameters.

Zero-knowledge verification overview

Zero-knowledge proofs allow a prover to prove that some computation is correct, even though they do not reveal any facts about the computation. The system under consideration has UAVs and an edge server, with the UAVs as verifiers and the edge server as a prover. The main aim is to ensure that the edge server faithfully executes the generative AI claims. The zero-knowledge verification is implemented in three consecutive steps. During the setup stage, the cloud server can generate system-wide public parameters and disseminate them to participating entities. During proof generation, the edge server issues a cryptographic attestation that the inference produced by the generative AI was accurate. During the verification stage, the appointed UAV authenticates the received evidence and submits the confirmation to the cloud server. The verification protocol is noninteractive and significantly reduces communication overhead, making it highly applicable to UAV-assisted verification in a dynamic edge computing environment.

The zero-knowledge verification scheme guarantees three important security properties when the adversary exists. First, completeness ensures that honest edge servers receive correct proofs that are accepted by UAV verifiers. Second, soundness guarantees that malicious edge servers do not produce valid proofs of erroneous computations, despite their attempts to manipulate model outputs. Third, zero-knowledge ensures that the knowledge regarding the proprietary model parameters, intermediate activations, or internal computations is not divulged during verification. With these properties, adversarial UAVs or external observers have no opportunity to detect sensitive model information during verification, and the strong secrecy and integrity of generative AI services are guaranteed.

Generative AI inference modeling

We consider a generative AI model deployed at edge server i , which is represented as a composition of L successive layers, as defined in Eq. (15).

$$y = f(x; W) = f_L \circ f_{L-1} \circ \cdots \circ f_1(x), \quad (15)$$

where in this formulation, x and y denote the input and output of the generative AI model, respectively, while W represents the set of model parameters. To enable zero-knowledge verification of the inference process, the overall computation is transformed into a polynomial constraint system. Specifically, each layer operation is encoded as a set of arithmetic constraints over a finite field $\mathbb{F}$, thereby facilitating the construction of a Rank-1 Constraint System (R1CS) suitable for zero-knowledge proof generation.

Constraint construction for generative AI operations

Linear transformation constraints

For linear layers, including fully connected and convolutional layers, the computation performed at layer l is modeled as defined in Eq. (16).

$$z^{(l)} = W^{(l)}h^{(l-1)} + b^{(l)}, \quad (16)$$

where in this expression, $h^{(l-1)}$ denotes the input activation vector to layer l , while $z^{(l)}$ represents the corresponding output activation vector. The parameters $W^{(l)}$ and $b^{(l)}$ denote the weight matrix and bias vector associated with the layer, respectively. This linear transformation is encoded as a set of linear polynomial constraints, which are directly compatible with the Rank-1 Constraint System (R1CS) formulation employed for zero-knowledge proof construction.

Nonlinear activation constraints

To represent nonlinear activation functions, including the rectified linear unit (ReLU), in a representation that is amenable to zero-knowledge verification, the behaviour of neuron-wise activation is represented as algebraic constraints, which may be integrated into a polynomial constraint system.

$$h_j^{(l)} = \max\left(0, z_j^{(l)}\right), \quad (17)$$

$$h_j^{(l)} = o_j^{(l)} \cdot z_j^{(l)}, \quad (18)$$

$$o_j^{(l)} \cdot (o_j^{(l)} - 1) = 0. \quad (19)$$

The nonlinear activation in Eq. (17) defines the standard ReLU operation, where $z_j^{(l)}$ and $h_j^{(l)}$ denote the pre-activation and post-activation values of neuron j at layer l , respectively. To enable compatibility with zero-knowledge proof systems, the activation is reformulated using the auxiliary binary variable $o_j^{(l)} \in \{0, 1\}$, as shown in Eq. (18). The polynomial constraint in Eq. (19) enforces the binary nature of $o_j^{(l)}$, ensuring that the output activation equals the input when the neuron is active and is zero otherwise. Together, these constraints provide an exact algebraic representation of the ReLU function while preserving model parameter privacy and maintaining compatibility with Rank-1 Constraint System formulations.

Normalization constraints

For normalization layers, including batch normalization and layer normalization, the computation performed at layer l is modeled as defined in Eq. (20).

$$h_j^{(l)} = \gamma_j \cdot \frac{z_j^{(l)} - \mu}{\sqrt{\sigma^2 + \epsilon}} + \beta_j, \quad (20)$$

where, $z_j^{(l)}$ and $h_j^{(l)}$ denote the pre-normalization and post-normalization activations of neuron j at layer l , respectively. The parameters γ_j and β_j represent learnable scaling and shifting coefficients, while μ and σ^2 denote the mean and variance used for normalization, and ϵ is a small constant introduced to ensure numerical stability. In order to perform zero-knowledge verification, a mapping of all the floating-point numbers into the normalization, as well as the fixed-point numbers, is carried out, keeping in mind that it needs to be adapted to the finite-field arithmetic to make it easier to generate zero-knowledge proofs that are secure and efficient. The construction, generation, and verification of the constraints are detailed procedures presented in Algorithm 1.

Require: Generative AI model $f(\cdot)$ with L layers; model parameters $\mathbf{W}$; inference input $\mathbf{x}$; fixed-point scale S ; finite field $\mathbb{F}$; public parameters (PK, VK)

Ensure: Proof π_i at edge server; verification result $r \in \{0, 1\}$ at UAV

Phase A: Constraint System Construction (Cloud or Offline Preprocessing)

- 1: **Model parsing:** Decompose inference as $\mathbf{y} = f(\mathbf{x}; \mathbf{W}) = f_L \circ \dots \circ f_1(\mathbf{x})$
- 2: **Fixed-point encoding:** Define $\text{Enc}(v) = \lfloor v \cdot S \rfloor$ and $\text{Dec}(z) = z/S$
- 3: **Initialize R1CS:** Create empty constraint set $\mathcal{C}$
- 4: **for** $l = 1$ to L **do**
- 5: **Linear layers:** Encode linear/conv operations
 $\mathbf{z}^{(l)} = \mathbf{W}^{(l)} \mathbf{h}^{(l-1)} + \mathbf{b}^{(l)}$
as linear/arithmetic constraints over $\mathbb{F}$
- 6: **Activation (ReLU):** For each neuron j , introduce $o_j^{(l)} \in \{0, 1\}$ and add constraints
 $h_j^{(l)} = o_j^{(l)} z_j^{(l)}, \quad o_j^{(l)} (o_j^{(l)} - 1) = 0$
- 7: **Normalization (BN/LN):** Encode normalization parameters (γ, β) and fixed-point approximations of division/square-root using polynomial constraints; add parameter-consistency constraints
- 8: **end for**
- 9: **Finalize circuit:** Output constraint system $\mathcal{C}$ and public input format $\mathcal{I}_{\text{pub}}$ (e.g., commitments to $\mathbf{x}$ and $\mathbf{y}$)

Phase B: Setup (Cloud Server)

- 10: Run trusted (or universal) setup: $(\text{PK}, \text{VK}) \leftarrow \text{Setup}(\mathcal{C})$
- 11: Distribute PK to edge servers and VK to UAVs

Phase C: Proof Generation (Edge Server i)

- 12: **Inference execution:** Compute output $\mathbf{y} = f(\mathbf{x}; \mathbf{W})$
- 13: **Witness construction:** Encode intermediate activations and parameters in fixed-point to form witness $\mathbf{w}$ satisfying $\mathcal{C}$
- 14: **Proof generation:** $\pi_i \leftarrow \text{Prove}(\text{PK}, \mathcal{I}_{\text{pub}}, \mathbf{w})$
- 15: Send $(\mathcal{I}_{\text{pub}}, \pi_i)$ to the verifying UAV

Phase D: Proof Verification (UAV u)

- 16: Compute $r \leftarrow \text{Verify}(\text{VK}, \mathcal{I}_{\text{pub}}, \pi_i)$
- 17: **return** $r = 1$ if valid, else $r = 0$

Algorithm 1. Zero-knowledge proof construction and verification for edge generative AI inference.

Zero-knowledge proof generation

Given the polynomial constraint system constructed from the generative AI inference process, the edge server produces a zero-knowledge proof that attests to the correctness of the computation at time slot t . The proof generation step is defined as in Eq. (21).

$$\pi_i(t) = \text{Prove}(\text{PK}, x, y, W), \quad (21)$$

where PK denotes the public parameters generated during the setup phase, x and y are the public input and output of the inference task, respectively, and W represents the model parameters. More precisely, the proving procedure takes as input the public parameters PK, the public I/O pair (x, y) , and a private witness w that includes fixed-point encoded intermediate activations and model parameters. It then produces a concise witness $\pi_i(t)$ that satisfies the cryptographic properties of completeness, soundness, and zero-knowledge. More specifically, valid computations have verifiable proofs, invalid computations never have accepted proofs other than with negligible probability, and information about the privacy of the witness or the intermediate states is never exposed during verification. This expression ensures that Eq. (21) actually describes a real cryptographic verification scheme rather than a mere abstraction.

The zero-knowledge verification system in this work is implemented with a zk-SNARK-based proving system, due to its succinct proof sizes and efficient verification functionality, the latter of which is appropriate to resource-constrained UAV-aided settings. The compilation of the generative AI inference process is expressed as a Rank-1 Constraint System (R1CS) over a finite field $\mathbb{F}$; fixed-point encoding is used to represent neural network computations in field arithmetic. In this construction, the number of neural network operations adds to the number of constraints in the initial setup, and the complexity of the proof is also determined by the size of the resulting circuit. Let $|C|$ represent the number of constraints caused by the model architecture. The cost of generating proofs is thus about $\mathcal{O}(|C|)$, and the cost of verification is $\mathcal{O}(1)$ with respect to circuit size in the assumed zk-SNARK setting. This asymmetry is why the framework is appealing for UAV-based validation: the computational load shifts to the edge server, while the verifier side remains lightweight.

The number of constraints is directly related to depth and form of the underlying generative model. In a model with L layers and N neurons, the circuit size increases about $\mathcal{O}(L \cdot N)$ due to the joint contribution of linear operations, nonlinear activations, as well as the normalization layers. Consequently, the overriding computational cost is proof generation, which can cause non-negligible latency at the edge server, particularly with more involved or complex models. In comparison, proof verification is efficient and viable for the implementation of UAVs. It should be noted that prover-side and verifier-side costs are critical to determining the feasibility of deployment. Specifically, the latency measurements in Section 6 are only proof-verification latency and do not reflect the full proof-generation latency, which is usually much larger and contingent on hardware capability and circuit complexity. This difference is crucial to a realistic interpretation of the framework's performance.

Other than computational latency, the communication cost of proof transmission is comparatively low since zk-SNARK proofs are constant in size, hence able to use bandwidth-limited UAV communication channels. However, the practical feasibility of the proposed framework will hinge on the trade-off between security assurances, the overhead of generating proofs, communication efficiency, and end-to-end system delay. These points suggest that the proposed strategy can be scaled as a verifier, but its practicality in the edge environment depends primarily on the computational power of the proving edge server.

UAV-assisted cooperative verification

Within the given framework, the UAVs serve as mobile verifiers that work together to verify (using zero-knowledge proofs issued by the edge servers) and disseminate (through the cloud server) verification results, thereby coordinating trust management and the verification timeline.

$$\text{Verify}(\text{VK}, y, \pi_i(t)) \in \{0, 1\}, \quad (22)$$

$$r_i(t) \in \{0, 1\}, \quad (23)$$

$$\tau_i(t+1) = \begin{cases} \tau_i(t) + \eta(1 - \tau_i(t)), & \text{if } r_i(t) = 1, \\ (1 - \eta)\tau_i(t), & \text{if } r_i(t) = 0. \end{cases} \quad (24)$$

The verification operation in Eq. (22) represents the execution of the zero-knowledge proof verification by a UAV, where VK denotes the verification key generated during the setup phase, y is the output of the generative AI inference, and $\pi_i(t)$ is the proof provided by edge server i at time slot t . The verification result is captured by the binary variable in Eq. (23), where a value of 1 indicates successful verification and a value of 0 denotes verification failure. Based on this outcome, the trust score of each edge server is updated according to the rule in Eq. (24), where $\eta \in (0, 1)$ is the trust update rate that controls the speed of trust adaptation. Successful verification increases the trust score toward unity, while verification failure degrades trust, enabling the UAV swarm to dynamically adjust verification priorities in a cooperative and trust-aware manner.

In compromised UAVs, other mechanisms might be necessary in order to guarantee robustness of the system. For example, redundant verification, in which multiple UAVs confirm the same edge server, can help identify inconsistent or malicious verification results. In much the same way, cross-validation and consensus systems may be employed to detect rogue UAVs that yield wrong outcomes. Although these mechanisms cannot be directly applied to the current structure, the suggested architecture can also be used in conjunction with such extensions, providing greater resilience against adversarial UAV behavior.

Security and privacy guarantees

The suggested zero-knowledge verification model provides strong security and privacy, helping ensure the credibility of edge-deployed generative AI providers. To be more specific, the framework satisfies the completeness property, so that honest edge servers running the purported generative AI model always generate correct, verifiable images. It also has soundness, meaning that malicious or deviant edge servers cannot produce valid zero-knowledge proofs of erroneous inference outputs. Moreover, the zero-knowledge feature ensures that UAVs and external coerced parties are not informed of any information about exclusive proprietary model parameters or internal calculations beyond what is revealed by the successful execution of the assertion. Moreover, cooperative UAV swarms can be useful because they offer collusion resistance through reduced single-point verification failures and attack resilience to isolated or targeted attacks. A combination of these properties ensures that edge generative AI services support model integrity while maintaining a high degree of secrecy.

Discussion and insights

The suggested zero-knowledge verification system will make generative AI inference at the network edges scalable, privacy-preserving, and energy-efficient. The framework, which combines zero-knowledge proofs with collaborating UAV swarms, addresses the trade-off among the timeliness of verification, the enforcement of the trust principle, and the efficient allocation of system resources in the context of a dynamic edge computing system. However, the verification timing issue and the UAV coordination challenge are always complex because the dynamics of the stochastic system, the UAV's mobility, and long-term energy availability are constrained. Such problems have prompted the implementation of a trust-sensitive multi-agent reinforcement learning methodology that enables adaptive, decentralized decision-making and is described in the next section.

Trust aware multiagent learning for cooperative UAV swarms

In this section, we present a trust-based multi-agent reinforcement learning system that enables a cooperative swarm of UAVs to effectively perform zero-knowledge verification in non-stationary network environments and under long-term constraints.

Motivation for multiagent learning

The verification scheduling problem, as developed in Sect. "System model and problem formulation", comprises integrated decision variables, long-term energy limits, UAV mobility, and the dynamic evolution of trust. Conventional optimization methods require a complete understanding of the entire system and cannot be applied under stochastic conditions for real-time control. Multi-agent reinforcement learning (MARL) offers a scalable and flexible approach, as UAVs can acquire cooperative policies by interacting with the environment without having an explicit system model.

In order to prevent the reinforcement learning framework as black-box, the policy learning process is defined explicitly by a state representation, action selection, and reward shaping. The trust-aware Age of Verification applied to the state and reward can ensure the learned policy accurately represents the system goals, such as verification timeliness, reliability, and energy efficiency. In addition, inter-agent dependencies are captured in the centralized critic, and the actors are decentralized to enable execution scaling, a feature that can be interpreted and deployed in practice using the learned policies.

It should also be mentioned that the proposed MARL framework is novel not so much in the sense of implementing a new learning algorithm, but in the sense of integrating a number of trust-aware system dynamics into the learning process. The framework embeds trust-aware Age of Verification into both the state representation and the reward function to enable agents to make decisions that are both energy-efficient and reliable for the system, and that consider verification timeliness. The proposed approach is unique in that it relies on a problem-specific formulation that aligns the learning goal with security-conscious verification needs in edge AI systems.

Multiagent Markov decision process formulation

The cooperative UAV verification problem is modeled as a multi-agent Markov decision process (MDP), formally defined as the tuple given in Eq. (25).

$$\langle \mathcal{U}, \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle, \quad (25)$$

where, $\mathcal{U}$ denotes the set of UAV agents, $\mathcal{S}$ represents the global state space, $\mathcal{A}$ is the joint action space of all agents, $\mathcal{P}$ characterizes the state transition probability function, and $\mathcal{R}$ denotes the reward function. The discount factor $\gamma \in (0, 1)$ controls the relative importance of immediate and future rewards and governs the long-term optimization behavior of the multiagent learning process.

State space

At time slot t , the global system state is defined as shown in Eq. (26).

$$s(t) = \{p_u(t), E_u(t), A_i(t), \tau_i(t) \mid u \in \mathcal{U}, i \in \mathcal{E}\}, \quad (26)$$

where the global state includes the positions and energy levels of all UAVs, as well as the trust-aware age of verification and trust scores associated with each edge server. Each UAV has access only to a partial observation of the global state, referred to as the local state, which is defined in Eq. (27).

$$s_u(t) \subseteq s(t), \quad (27)$$

where the local state comprises the observing UAV's position and energy information, along with state information from nearby edge servers within its sensing or communication range, enabling decentralized decision-making under partial observability.

Action space

At each time slot t , each UAV u selects a joint action as defined in Eq. (28).

$$\mathbf{a}_u(t) = (\Delta \mathbf{p}_u(t), i_u(t)), \quad (28)$$

where in this formulation, the action consists of two components: $\Delta \mathbf{p}_u(t)$, which specifies the movement direction and distance of the UAV, and $i_u(t) \in \mathcal{E} \cup \{0\}$, which denotes the index of the edge server selected for verification, where $i_u(t) = 0$ indicates that no verification is performed during the time slot. The joint action of the entire UAV swarm is defined in Eq. (29).

$$\mathbf{a}(t) = \{\mathbf{a}_u(t)\}_{u \in \mathcal{U}}, \quad (29)$$

where the joint action aggregates the individual actions of all UAVs and fully characterizes the collective mobility and verification decisions of the UAV swarm at time slot t .

State transition

The evolution of the global system state over time is governed by the state transition model defined in Eq. (30).

$$\mathbf{s}(t+1) \sim \mathcal{P}(\mathbf{s}(t+1) | \mathbf{s}(t), \mathbf{a}(t)). \quad (30)$$

This probabilistic transition model captures the system's stochastic dynamics, which are influenced by UAV mobility constraints, verification decisions, energy consumption, and trust updates. Given the current state $\mathbf{s}(t)$ and the joint action $\mathbf{a}(t)$ selected by the UAV swarm, the next state $\mathbf{s}(t+1)$ is generated according to the transition probability function $\mathcal{P}(\cdot)$, reflecting the inherent uncertainty and time-varying nature of the UAV-assisted edge verification environment. To enforce one-to-one matching between UAVs and edge servers and resolve assignment conflicts, a cooperative swarm-based conflict-resolution mechanism is adopted, as detailed in Algorithm 2.

Priority scoring operation adds weights to the contribution of the trust-conscious Age of Verification, spatial proximity, and leftover energy. The fixed weights can be used to give a straightforward, interpretable system of balancing these opposing aims. Even though fixed weights may not adequately represent dynamic system conditions, the sensitivity analysis shows that the framework is robust across a range of plausible parameter values. These weights may be learned or adaptively tuned in practice to respond to varying service demands and environmental characteristics through meta-learning.

Require: Selected targets $i_u(t)$ from each UAV policy; TA-AoV values $\{A_i(t)\}$; UAV positions $\{\mathbf{p}_u(t)\}$; optional residual energies $\{E_u(t)\}$

Ensure: Feasible assignment matrix $x_{u,i}(t)$ satisfying one-to-one matching constraints

```

1: Initialize  $x_{u,i}(t) \leftarrow 0$  for all  $u \in \mathcal{U}$  and  $i \in \mathcal{E}$ 
2: Build candidate set  $\mathcal{C} = \{(u, i_u(t)) \mid i_u(t) \neq 0\}$ 
3: for all edge servers  $i$  that appear in  $\mathcal{C}$  do
4:   Let  $\mathcal{U}_i = \{u \mid (u, i) \in \mathcal{C}\}$  be the UAVs requesting server  $i$ 
5:   if  $|\mathcal{U}_i| = 1$  then
6:     Assign  $x_{u,i}(t) = 1$  for the single  $u \in \mathcal{U}_i$ 
7:   else ▷ Conflict case
8:     for all  $u \in \mathcal{U}_i$  do
9:       Compute distance  $d_{u,i}(t) = \|\mathbf{p}_u(t) - \mathbf{p}_i\|$ 
10:      Compute priority score
11:       $\Psi_{u,i}(t) = \omega_1 A_i(t) - \omega_2 d_{u,i}(t) + \omega_3 \tilde{E}_u(t)$ 
12:    end for
13:    Choose winner  $u^* = \arg \max_{u \in \mathcal{U}_i} \Psi_{u,i}(t)$ 
14:    Assign  $x_{u^*,i}(t) = 1$  and reject all other UAVs for server  $i$ 
15:  end if
16: end for
17: One-UAV-One-Task Enforcement
18: for all UAVs  $u$  assigned to multiple servers do
19:   Keep the assignment with maximum  $\Psi_{u,i}(t)$  and set all others to zero
20: end for
21: return assignment matrix  $x_{u,i}(t)$ 

```

Algorithm 2. Cooperative UAV swarm conflict resolution and one-to-one matching.

Require: Edge servers $\mathcal{E}$; UAVs $\mathcal{U}$; time horizon T ; weights α, β ; trust update rate η ; AoV decay δ ; discount factor γ ; energy budget $E_u^{\max}$; maximum speed

$v_{\max}$
Ensure: Decentralized UAV policies $\{\pi_u\}_{u \in \mathcal{U}}$; verification decisions $x_{u,i}(t)$; trust scores $\tau_i(t)$

Phase 1: Cryptographic Initialization (Cloud Server)

- 1: Generate ZK public parameters (PK, VK) $\leftarrow \text{Setup}(\mathcal{C})$
- 2: Distribute PK to edge servers and VK to UAVs

Phase 2: System Initialization

- 3: Initialize UAV positions, energies, and trust scores $\tau_i(0)$
- 4: Compute trust-aware AoV $A_i(0) = \alpha a_i(0) + \beta(1 - \tau_i(0))$
- 5: Initialize MARL actor and critic parameters

Phase 3: Cooperative Verification and Learning (Per Time Slot)

- 6: **for** $t = 0$ to $T - 1$ **do**
- 7: Each UAV observes local state $\mathbf{s}_u(t)$
- 8: **for all** $u \in \mathcal{U}$ **do**
- 9: Sample action $\mathbf{a}_u(t) \sim \pi_{\theta_u}(\cdot | \mathbf{s}_u(t))$
- 10: **end for**
- 11: Resolve UAV conflicts using matching rule and set $x_{u,i}(t)$
- 12: **for all** $u \in \mathcal{U}$ **do**
- 13: Update UAV position $\|\mathbf{p}_u(t+1) - \mathbf{p}_u(t)\| \leq v_{\max}$
- 14: **end for**
- 15: **for all** $i \in \mathcal{E}$ such that $\exists u : x_{u,i}(t) = 1$ **do**
- 16: Execute inference and generate proof $\pi_i(t)$
- 17: **end for**
- 18: **for all** (u, i) such that $x_{u,i}(t) = 1$ **do**
- 19: Verify proof $r_i(t)$ and share with UAVs/cloud
- 20: **end for**
- 21: **for all** $i \in \mathcal{E}$ **do**
- 22: **if** $r_i(t) = 1$ **then**
- 23: $\tau_i(t+1) = \tau_i(t) + \eta(1 - \tau_i(t))$
- 24: **else**
- 25: $\tau_i(t+1) = (1 - \eta)\tau_i(t)$
- 26: **end if**
- 27: **end for**
- 28: **for all** $i \in \mathcal{E}$ **do**
- 29: Update AoV $a_i(t+1)$ and $A_i(t+1)$
- 30: **end for**
- 31: **for all** $u \in \mathcal{U}$ **do**
- 32: Compute UAV energy $E_u(t)$
- 33: Apply penalty if $E_u(t) > E_u^{\max}$
- 34: **end for**
- 35: Compute reward $R(t)$ and store transition in $\mathcal{B}$
- 36: Every K steps, update critic ϕ and actors $\{\theta_u\}$
- 37: **end for**
- 38: **return** $\{\pi_{\theta_u}\}_{u \in \mathcal{U}}$, $\{\tau_i(t)\}$, and $\{x_{u,i}(t)\}$

Algorithm 3. Cooperative UAV swarm framework for zero knowledge verification of edge generative AI using trust-aware multi-agent learning.

Trust aware reward design

To guide cooperative behavior among UAVs, a global reward function is designed to jointly account for trust-aware verification performance and energy efficiency, as defined in Eq. (31).

$$R(t) = - \sum_{i \in \mathcal{E}} A_i(t) - \lambda \sum_{u \in \mathcal{U}} E_u(t). \quad (31)$$

The parameter in this reward formulation would encourage high trust-Awareness Age of Verification values, thereby promoting timely verification and favoring edge servers with low trust or stale verification parameters. The second term penalizes excessive UAV energy usage, with the weighting coefficient λ determining the trade-off between verification performance and energy efficiency. These goals can be achieved by integrating them into the reward structure to motivate UAVs to work together in special-risk edge servers, verify work twice, and save power by coordinating schedules and routes.

Centralized training and decentralized execution

To support efficient, scalable learning in the cooperative UAV verification system, we adopt a centralized training and decentralized execution (CTDE) paradigm. In this paradigm, the training period uses a centralized critic that can access the international system state and therefore evaluate joint action and inter-agent dependencies correctly. In operation, each UAV operates independently and makes decisions based solely on local observations, resulting in a fully decentralized system. The joint policy of the UAV swarm is defined as shown in Eq. (32).

$$\pi = \{\pi_u(a_u | s_u)\}_{u \in \mathcal{U}}, \quad (32)$$

where, $\pi_u(a_u | s_u)$ denotes the policy of UAV u , which maps its local observation s_u to a probability distribution over its action space a_u . This CTDE structure enables effective coordination during training while preserving scalability and robustness during decentralized execution.

Trust aware multi-agent actor-critic learning

A trust-aware multi-agent actor-critic learning framework is employed to support collaborative decisions and interactions under dynamic trust, mobility, and energy constraints, in which decentralized actors are trained using feedback from a centralized critic informed by the dynamics of the global system.

$$Q_\phi(s, a), \quad (33)$$

$$\nabla_{\theta_u} J(\theta_u) = \mathbb{E} [\nabla_{\theta_u} \log \pi_u(a_u | s_u) \cdot \hat{A}(t)], \quad (34)$$

$$\hat{A}(t) = Q_\phi(s(t), a(t)) - V_\phi(s(t)), \quad (35)$$

$$\mathcal{L}(\phi) = \mathbb{E} [(Q_\phi(s(t), a(t)) - y(t))^2], \quad (36)$$

$$y(t) = R(t) + \gamma V_\phi(s(t+1)). \quad (37)$$

The centralized critic in Eq. (33) estimates the global action-value function as a function of the joint system state s and joint action a , capturing interdependencies among UAVs. The actor update in Eq. (34) computes the policy gradient for UAV u using the advantage signal, enabling each agent to improve its policy based on expected long-term rewards. The advantage function in Eq. (35) measures the relative benefit of the selected action relative to the state-value baseline, thereby reducing variance in gradient estimation. The critic parameters are optimized by minimizing the temporal-difference loss defined in Eq. (36), which penalizes discrepancies between predicted and target action-value estimates. The target value in Eq. (37) incorporates the immediate reward $R(t)$ and the discounted value of the next state, where $\gamma \in (0, 1)$ is the discount factor. Together, these equations enable stable, trust-aware learning of cooperative verification policies within the centralized training and decentralized execution paradigm.

Although a formal convergence result of multi-agent reinforcement in non-stationary settings has yet to be proven, the offered framework is based on the well-known actor-critic algorithms that have a known convergence rate in specific settings. Specifically, actor-critic algorithms under bounded rewards, adequate exploration, and consistent policy updates have been shown to find locally optimal policies. The centralized training and decentralized execution (CTDE) paradigm further enhances the stability of learning by establishing a uniform global value function during training. Hence, the suggested trust-sensitive MARL architecture preserves the convergence property of regular MAPPO-based algorithms while utilizing domain-tailored changes to states and rewards. The stability of the learning process can also be reinforced by empirical convergence results in Sect. "Learning convergence and stability analysis".

Cooperative trust awareness

Implicit trust awareness is embedded in the learning process through the introduction of the trust-conscious Age of Verification state variable, $A_i(t)$, and through trust-conscious shaping with rewards. Through collaborative encoding of the freshness of verification and the reliability of state servers, and through reward design, UAVs can learn long-run trust dynamics between edge servers and, consequently, adapt their verification approach over time. This collaborative, trust-based learning system makes sound decisions despite partial observability of conditions, enabling the UAV swarm to efficiently prioritize high-risk servers, behave effectively, and coordinate with verification.

The trust-aware multi-agent learning process consists of a series of coordinated actions, executed iteratively. At the first step, the actor networks for individual UAVs and the centralized critic network are initialized. During the learning iteration, UAVs engage with the environment by making decisions based on existing policies and collecting experience trajectories of their actions. The global value estimates are consequently updated using information from the centralized critic and from aggregating experience and the global state. Thereafter, the individual UAVs independently update their actor policies in a decentralized process using critic-based feedback. Meanwhile, the trust scores and age-of-verification metrics, which are trust-aware, are updated based on the verification results. This is repeated until the learning algorithm has reached a fixed policy. Altogether, this learning framework will allow scaling, flexible, and cooperative UAV verification in dynamic edge AI settings. Overall, the cooperative verification and trust-aware learning algorithm is summarized in Algorithm 3.

Performance evaluation and simulation results

In this section, we critically analyze the suggested cooperative UAV swarm-based zero-knowledge verification framework of trust-aware multi-agent learning (TAML). It runs large-scale simulations to assess the timeliness of verification, trust enforcement capacity, energy efficiency, scalability, and learning performance under dynamic network conditions.

Simulation setup and parameter configuration

The simulation environment consists of a two-dimensional square area of $200 \times 200 \text{ m}^2$, with a cloud server located at the center of the region. Within this area, a set of $E = 4$ to 8 edge servers is randomly deployed

to provide generative AI inference services under the Model-as-a-Service paradigm. To perform mobile zero-knowledge verification, a cooperative swarm of $U = 3$ to 5 UAVs is deployed and operates over the same region. To find a more rigorous and fair comparison, the proposed framework is contrasted with various baseline schemes that reflect the philosophy of different designs, such as non-cooperative, heuristic-based, and learning-based schemes. Performance evaluation is based on several measures, including trust-aware Age of Verification (TA-AoV), detection delay, energy usage, and scalability. The outcome of each simulation is the average over numerous independent runs, minimizing randomness to achieve statistical reliability. The chosen baselines, such as DQN, MADDPG, and MAPPO, are powerful benchmarks for reinforcement learning-based coordination, enabling a full assessment of the presented trust-aware multi-agent framework. To guarantee robustness and ensure good statistics, all experimental results are averaged over several independent simulation runs with different random initializations. Although no specific statistical values are provided, like confidence intervals, there was no difference in the same patterns in each run. To ensure diversity, the implementation of all learning-based baselines is conducted with similar network designs, training times, and observation spaces. The proposed approach can be considered as a trust extension of MAPPO, in which the changes are brought about in terms of state representation and reward design, but not in terms of the general structure of the algorithm.

Each simulation run is conducted for 8000 time slots to ensure sufficient learning convergence and stable long-term performance. UAV mobility is constrained by a maximum velocity of $v_{\max} = 8$ m/s, and each UAV is subject to an average energy budget of $E_u^{\max} = 120$ energy units per time slot. The trust update rate is set to $\eta = 0.1$, while the weighting coefficients of the trust-aware Age of Verification (TA-AoV) metric are configured as $\alpha = 0.7$ and $\beta = 0.3$, reflecting a balanced emphasis on verification freshness and server trustworthiness.

The choice of main parameters demonstrates a compromise between the verification performance, energy efficiency, and stability of the system. Selection of the weighting coefficients α and β is intended to favor verification freshness while accounting for trust dynamics. The rate of trust update η dictates how responsive the system is to verification results, ensuring it neither adapts trust too aggressively nor too slowly. Energy weighting parameter λ is chosen to trade-off between the performance optimization and resources. The sensitivity analysis will verify that the framework remains stable within a reasonable range of parameter values, and thus the proposed design can be considered robust.

A summary of the neural network architectures of the actor and critic models, along with the learning hyperparameters, is provided in Table 1. Additionally, Table 2 lists key system-level simulation parameters, including communication, mobility, and energy-related settings, providing a detailed account of the experimental setup.

Baseline schemes

The suggested trust-aware multi-agent learning (TAML) framework is tested by comparing it with several representative baseline schemes that reflect various verification, coordination, and learning plans. A lower-bound benchmark is Independent UAV Verification (IUUV), a counterexample in which UAVs operate without cooperation or learning. Random Verification Scheduling (RVS) randomly verifies edge servers but lacks a smart approach. Freshness-Based Scheduling (FBS) is a verification decision-making method that does not consider trust or energy and instead bases decisions solely on verification age. Trust-Blind Cooperative Scheduling (TBCS) with cooperative verification among UAVs is not based on trust awareness in the decision process. The Energy-Aware Greedy (EAG) scheme optimally allocates UAVs to the nearest edge server to reduce energy consumption, without accounting for verification risk or learning dynamics. Moreover, there are learning-based

Parameter	Symbol/description	Value
Actor network layers	Fully connected layers	3 layers
Actor hidden units	Units per hidden layer	128, 128
Critic network layers	Fully connected layers	3 layers
Critic hidden units	Units per hidden layer	256, 256
Activation function	Nonlinear activation	ReLU
Output activation	Policy output	Softmax
Learning algorithm	MARL framework	Trust-Aware MAPPO
Actor learning rate	α_a	3×10^{-4}
Critic learning rate	α_c	1×10^{-3}
Discount factor	γ	0.99
GAE parameter	λ_{GAE}	0.95
Entropy coefficient	λ_e	0.01
Mini-batch size	Training batch size	256
Replay buffer size	Experience buffer	1×10^6
Training episodes	Total episodes	8,000
Update frequency	Policy update interval	Every 10 time slots

Table 1. Neural network architecture and learning hyperparameters for the proposed trust-aware multi-agent learning framework.

Parameter	Symbol/description	Value
Simulation area	Deployment region	200 m × 200 m
Number of edge servers	E	4–8
Number of UAVs	U	3–5
Cloud server location	Reference point	Center of area
Time slot duration	Δt	1 s
Total simulation time slots	T	8000
UAV maximum velocity	$v_{\max}$	8 m/s
UAV communication range	R_u	60 m
UAV initial energy	E_u^{init}	5000 units
UAV average energy budget	$E_u^{\max}$	120 units/slot
Flight energy coefficient	c_f	1.2 units/m
Communication energy coefficient	c_c	0.8 units/MB
Verification energy per proof	E^{ver}	15 units
ZK proof verification latency	—	≤ 480 ms
Trust update rate	η	0.1
TA-AoV weight (freshness)	α	0.7
TA-AoV weight (trust)	β	0.3
AoV decay factor	δ	0.5
Discount factor	γ	0.99

Table 2. System-level simulation parameters for UAV-assisted zero-knowledge verification.

Scheme	Learning-based	UAV cooperation	Trust awareness	Energy awareness	Verification strategy
IUV	✗	✗	✗	✗	Each UAV independently verifies the nearest edge server
RVS	✗	✗	✗	✗	Edge servers are selected randomly for verification
FBS	✗	○	✗	✗	Verification decisions based solely on age of verification
TBCS	✗	✓	✗	○	Cooperative verification without explicit trust modeling
EAG	✗	○	✗	✓	Nearest-server selection to minimize UAV energy consumption
DQN	✓	✗	✗	○	Centralized learning with discretized action space
MADDPG	✓	✓	✗	✓	Cooperative learning with continuous control but without trust-aware state
MAPPO	✓	✓	✗	✓	Cooperative multi-agent learning without trust awareness
Proposed TAML	✓	✓	✓	✓	Trust-aware cooperative multi-agent learning with centralized training and decentralized execution

Table 3. Comparison of baseline verification schemes and their key characteristics. ✓: Yes; ✗: No; ○: Partial.

baselines: a single-agent Deep Q-Network (DQN) with centralized learning; a discrete-action-space cooperative learning scheme based on MADDPG; a continuous-action-space cooperative learning scheme without trust-driven state design; and cooperative learning schemes based on MAPPO, without trust-driven state design and with trust-driven reward shaping. Table 3 summarizes the comparison of the baseline characteristics and modeling assumptions. The chosen baselines are widely used methods in heuristic scheduling and multi-agent reinforcement learning, such as value-based, policy-gradient, and cooperative learning paradigms. Although more recent variants of the MARL exist, many are based on similar architectural building blocks, and the chosen methods offer a fair comparison of the most important design dimensions. It is also a significant avenue in future research to extend the assessment to other, more recent MARL methods.

Learning convergence and stability analysis

The convergence behavior of the cumulative reward during training provides important insight into the learning efficiency and stability of different schemes. As illustrated in Fig. 2, the proposed trust-aware multiagent learning (TAML) scheme converges in approximately 2,200 training episodes, demonstrating rapid and stable learning. In comparison, MAPPO and MADDPG require around 3,600 and 4,100 episodes, respectively, to reach convergence. In contrast, the DQN-based approach exhibits pronounced oscillations and unstable convergence, which can be attributed to discretizing the continuous action space and its limited ability to capture the coupled dynamics between UAV mobility and verification decisions.

In the steady-state regime, the proposed TAML framework achieves an average cumulative reward of −14.2, outperforming MAPPO, MADDPG, and independent UAV verification (IUV), which attain average rewards

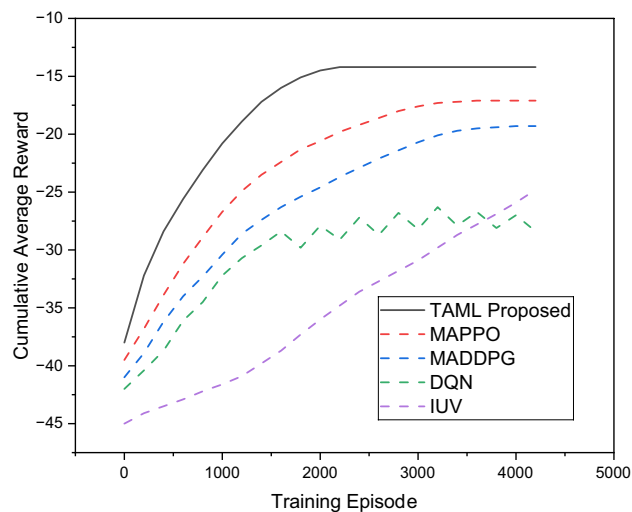


Fig. 2. Training convergence of cumulative average reward under different schemes.

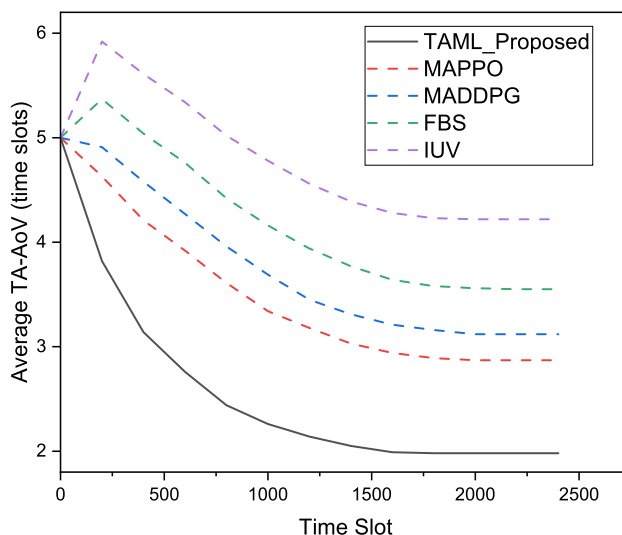


Fig. 3. Evolution of average trust-aware Age of Verification under different verification schemes.

of -17.1 , -19.3 , and -24.8 , respectively. These results correspond to performance improvements of 17.0%, 26.4%, and 42.7%, highlighting the superior learning efficiency, stability, and robustness of the proposed trust-aware multi-agent framework in dynamic edge-verification environments.

To prove that the suggested framework provides real cooperation behavior among UAVs, we discuss coordination efficiency in terms of some other indicators. Task overlap rate: This is the rate of the percentage of time that two or more UAVs are attempting to verify the same edge server. The proposed framework also has less overlap than independent or non-cooperative baselines, suggesting an efficient task allocation. Spatial coverage diversity: This is calculated as the distribution of UAV locations in comparison to edge servers. The findings indicate that UAVs strategically distribute themselves across the deployment area, enhancing coverage while minimizing unnecessary motion. Comparison with independent learners: The proposed method consumes less energy and has lower verification delay than the Independent UAV (IUUV), indicating that coordinated policies are more effective than greedy or independent strategies. All these findings prove that the mastered policies are coordinated, non-redundant, and spatially efficient, and confirm the statement of cooperative UAV swarm intelligence.

Trust-aware age of verification performance

The efficiency of Trust awareness in verification scheduling can be measured by the average Trust-Aware Age of Verification (TA-AoV) across all edge servers. As shown in Fig. 3, the proposed trust-aware multiagent learning (TAML) scheme always achieves the lowest TA-AoV during both the training and execution stages, indicating timely, risk-aware verification behavior.

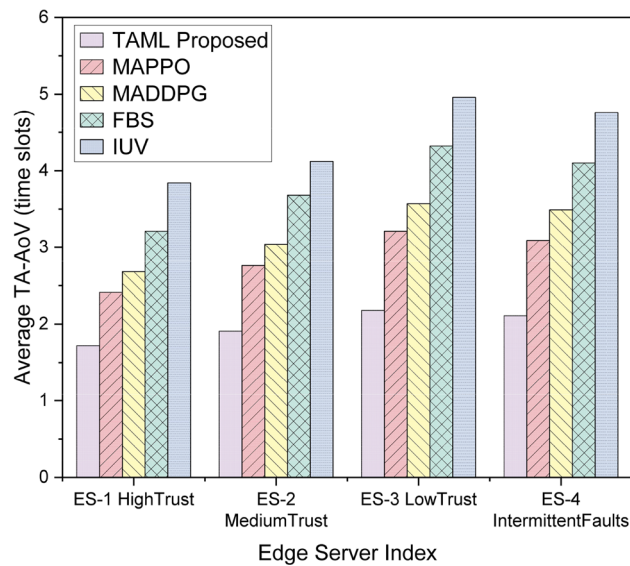
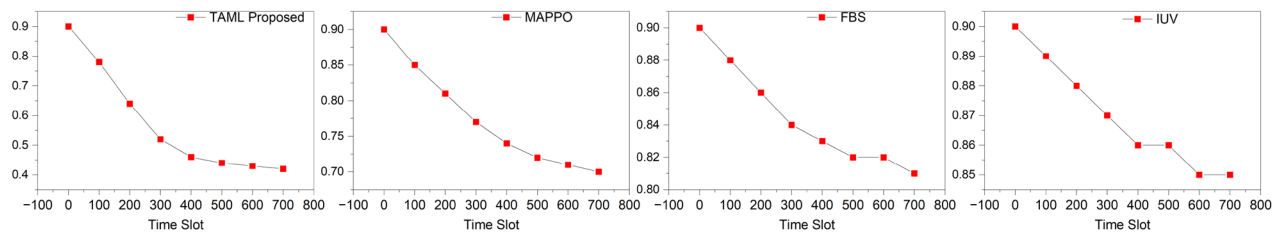
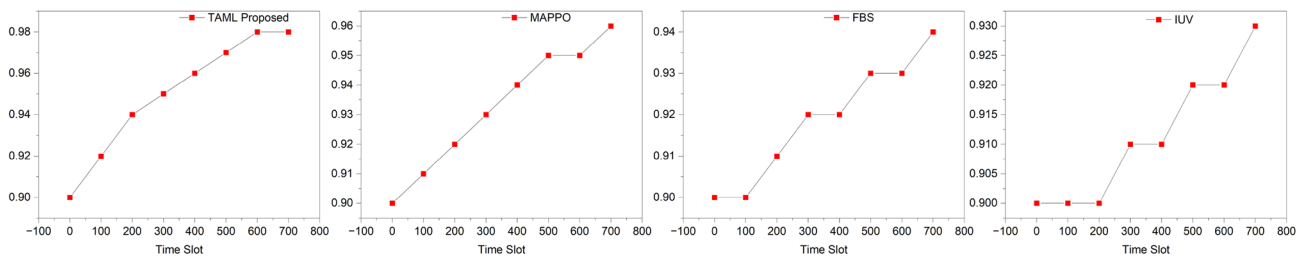


Fig. 4. Per-edge-server average TA-AoV comparison under different verification schemes.



(a) Low-reliability edge server with intermittent faults.



(b) High-reliability edge server.

Fig. 5. Evolution of trust scores for representative edge servers under different verification schemes: **(a)** low-reliability edge server with intermittent faults; **(b)** high-reliability edge server.

At convergence, the proposed TAML framework achieves an average TA-AoV of 1.98 time slots, whereas MAPPO, MADDPG, and freshness-based scheduling (FBS) attain average TA-AoV values of 2.87, 3.12, and 3.95, respectively. Compared with trust-blind cooperative scheduling, TAML reduces the TA-AoV by approximately 31.0%, demonstrating the significant benefit of explicitly incorporating trust information into cooperative verification decisions.

A more detailed per-edge-server analysis further illustrates the advantages of the proposed approach. As shown in Fig. 4, edge servers with lower trust levels or higher verification risk experience the largest TA-AoV reductions under the TAML scheme, confirming that trust-aware learning effectively prioritizes high-risk servers in dynamic edge environments.

Trust evolution and malicious server detection

The dynamic nature of trust scores offers insight into how various schemes detect unreliable or malicious edge servers. Figure 5 also demonstrates that the edge server leads to a reduction in the trust score, rapidly reaching below 0.4 with under 350 time slots when it is intermittently producing invalid zero-knowledge proofs, starting

Scheme	Avg. detection delay (time slots)	Min. detection delay (time slots)	Max. detection delay (time slots)	Detection accuracy (%)
IUV	820	610	1,050	71.4
RVS	910	680	1,120	68.9
FBS	670	490	890	76.2
EAG	590	430	810	79.6
TBCS	520	390	760	82.1
DQN	480	350	720	84.7
MADDPG	430	320	680	87.3
MAPPO	390	290	610	89.8
Proposed TAML	214	160	340	95.6

Table 4. Comparison of malicious edge server detection delay under different verification schemes.

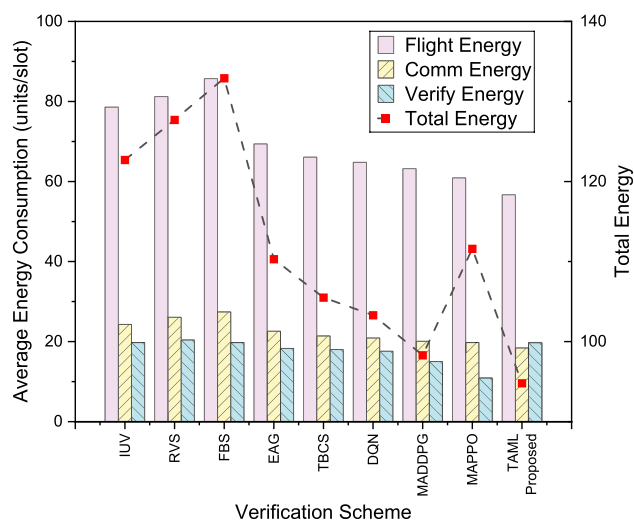


Fig. 6. Average per-UAV energy consumption under different verification schemes.

at 0.9. Such a rapid decrease in trust triggers repeated checks that help identify malicious activity in time and take appropriate measures.

Compared with trust-blind verification schemes, the same edge server is allowed to score above 0.7 for a longer period, resulting in delayed detection and a higher risk of verification. Table 4 offers a quantitative comparison of malicious server detection delay, on which the proposed TAML framework results in a 45.6% longer detection delay reduction than MAPPO. These findings indicate that trust-aware learning is useful for accelerating the detection of malicious edge servers and improving overall system security.

The proposed framework provides better, more timely verification, shorter detection delay, and greater energy efficiency than conventional UAV-assisted edge verification with fixed scheduling or non-cooperative strategies. These advancements outline the advantages of using trust-awareness and cooperative learning in UAV-based validation systems.

Energy consumption and efficiency analysis

The efficiency of various verification schemes can be determined by comparing the average energy consumption per-UAV in time. As demonstrated in Fig. 6, the proposed trust-aware multiagent learning (TAML) frameworks use an average of 94.8 energy units in each time slot, which is much lower than the energy usage in MAPPO, MADDPG, and freshness-based scheduling (FBS), which used 111.6, 118.3, and 132.9 units of energy in each time slot, respectively.

These findings translate to energy savings of 15.0%, 19.9%, and 28.7% compared to MAPPO, MADDPG, and FBS, respectively, demonstrating the efficacy of cooperative and trust-aware task allocation in minimizing total energy expenditure. An additional analysis of the energy components, such as flight, communication, and verification energy, is presented in Table 5, which further indicates that cooperative scheduling significantly reduces flight energy waste and increases energy usage efficiency in the long run.

Scalability with respect to UAV and edge density

The scalability of the proposed framework is tested on the trust-aware Age of Verification (TA-AoV) with different numbers of edge servers and UAVs. As Fig. 7 shows, the average TA-AoV decreases a bit when the number of edge servers is raised over 4 to 8 in the depiction of the proposed trust-aware multiagent learning

Scheme	Flight energy (units/slot)	Communication energy (units/slot)	Verification energy (units/slot)	Total energy (units/slot)
IUV	78.6	24.3	19.8	122.7
RVS	81.2	26.1	20.4	127.7
FBS	85.7	27.4	19.8	132.9
EAG	69.4	22.6	18.3	110.3
TBCS	66.1	21.4	18.0	105.5
DQN	64.8	20.9	17.6	103.3
MADDPG	63.2	20.1	15.0	98.3
MAPPO	60.9	19.8	10.9	111.6
Proposed TAML	56.7	18.4	19.7	94.8

Table 5. Average per-UAV energy consumption breakdown under different verification schemes.

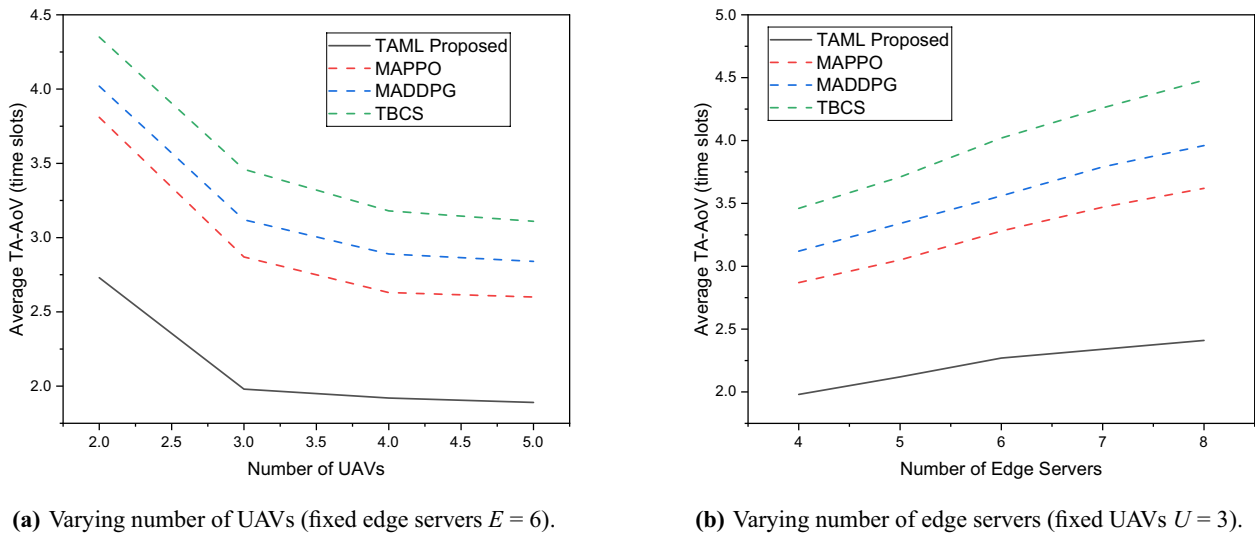


Fig. 7. Impact of the number of UAVs and edge servers on average trust-aware Age of Verification: (a) varying number of UAVs with fixed edge servers; (b) varying number of edge servers with fixed UAVs.

(TAML) framework, to 2.41. On the contrary, MAPPO has even stronger degradation, whose average TA-AoV rises between 2.87 and 3.62, respectively, meaning that it is less scalable as the density of the edge servers grows. The problem of UAV density also shows the benefits of the trust-based cooperative learning. When the number of UAVs is increased to 5, the TA-AoV is decreased by 22.3% on the TAML scheme, whereas the TA-AoV only decreases by 9.4% when trust-blind cooperative scheduling is used. These findings reveal that the presented solution effectively leverages new UAV capabilities to achieve resilient performance in dense edge computing scenarios, and that cooperative swarm intelligence, when combined with trust-sensitive learning, is more scalable.

Although in the existing simulations a moderate-sized deployment has been considered, the proposed framework is designed to scale effectively with more UAVs and edge servers. The decentralized execution paradigm guarantees that decision-making complexity scales linearly with the number of agents, whereas the centralized training step can leverage parallel computation. Additionally, a compact representation of states and local observations makes the framework less dependent on communication overheads, thereby allowing it to be used in larger-scale settings. The evaluation of these properties will be further verified in future work by deploying the system at large scale. These findings show that the suggested framework has better scaling in the assessed simulation environment. However, further validation in larger-scale and real-world deployments is required to fully establish scalability in practical scenarios.

Zero-knowledge verification latency analysis

The latency of verifying zero-knowledge proofs is compared across various depths of generative AI models, enabling assessment of the feasibility of the proposed framework for real-time edge services. As indicated in Fig. 8, for model depths up to 18 layers, the average verification latency remains below 470 ms, and the computationally intensive proof-generation process is performed on edge servers.

This is further demonstrated by a detailed breakdown of the latency contributions of various cryptographic functions of the proposed approach. The findings presented in Table 6 ensure that the verification overhead is

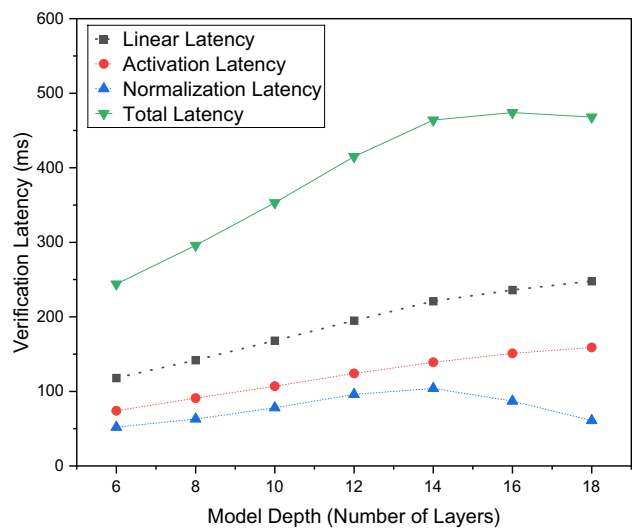


Fig. 8. Zero-knowledge proof verification latency under different generative AI model depths.

Model depth (layers)	Linear operation latency (ms)	Activation constraint latency (ms)	Normalization constraint latency (ms)	Total verification latency (ms)
6	118	74	52	244
8	142	91	63	296
10	168	107	78	353
12	195	124	96	415
14	221	139	104	464
16	236	151	87	474
18	248	159	61	468

Table 6. Zero-knowledge proof verification latency under different generative AI model depths.

small enough to support zero-knowledge verification in real-time edge generative AI systems, which proves that zero-knowledge verification can be efficiently incorporated into latency-sensitive edge intelligence systems. Though the proposed framework shows possible verification latency, it should be mentioned that the overhead in proof generation at edge servers is a bottleneck in computations. The production of zero-knowledge proofs for large generative models can introduce delays under resource-starved conditions. This can be overcome by model compression, by verifying tasks in batches, or by running the verification tasks using hardware (e.g. GPU/FPGA-based proof generation). Also, in real-world applications, edge servers are often larger than end-user devices, so these operations are becoming real-time. This points to a critical trade-off between security assurances and computation cost.

Sensitivity analysis of trust parameters

The impact of trust-related parameters on system performance is examined through a sensitivity analysis of the trust update rate, η , and the Trust-Aware Age of Verification (TA-AoV) weighting coefficients, α and β . As illustrated in Fig. 9, the results indicate that setting the trust update rate η within the range of 0.08 to 0.15 leads to stable trust convergence and balanced verification behavior. In contrast, assigning excessive weight to trust awareness, with $\beta > 0.5$, increases UAV energy consumption by overly aggressively verifying low-trust servers. Based on these observations, optimal parameter settings are determined to ensure stable, energy-efficient system operation. The parameter variations for the trust update and TA-AoV weighting coefficients are summarized in Table 7, providing practical guidelines for tuning the proposed framework for real-world deployments.

Ablation study of framework components

An ablation analysis assesses the contribution of each major component in the proposed framework by sequentially removing key modules and measuring performance afterward. Specifically, we consider three versions of the framework. In the former, called No-Trust, TA-AoV is substituted by the traditional AoV, and there is no longer trust-based prioritization. In the second type, an Independent UAV, UAVs operate independently and decide for themselves whether to verify. In the third one, called No-ZKP, the verification is supposed to be direct and does not involve generating a cryptographic proof. The numerical analysis comparing these variants is in Table 8, and performance indicators, such as TA-AoV, detection delay, and power consumption, are compared. The findings indicate that, without trust-awareness, the

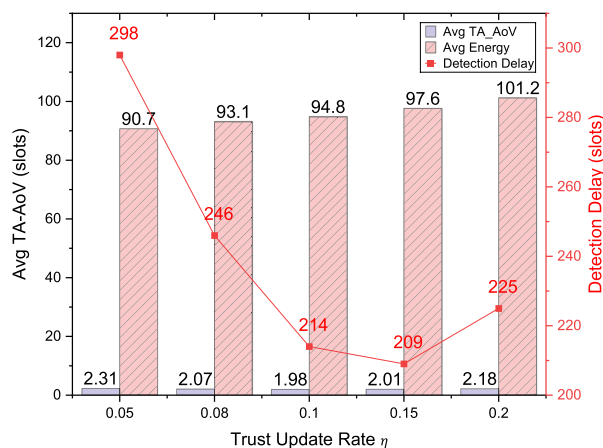
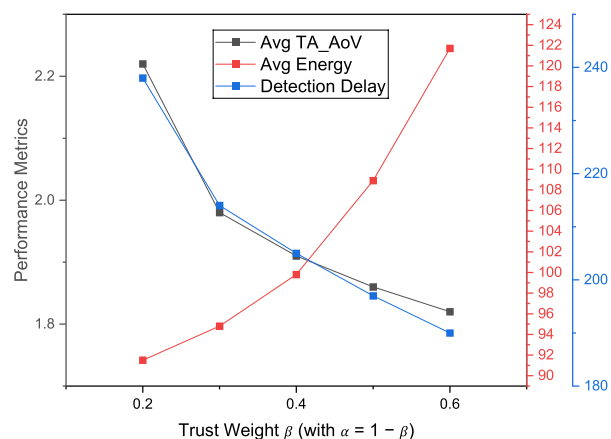
(a) Impact of trust update rate η .(b) Trust weight β (with $\alpha = 1 - \beta$).

Fig. 9. Impact of trust update rate and trust-aware weighting factors on verification performance and energy consumption: (a) sensitivity to trust update rate η ; (b) sensitivity to trust-aware weighting factors α and β .

Case	Trust update rate η	Freshness weight α	Trust weight β	Avg TA-AoV (slots)	Avg energy (units/slot)	Detection delay (slots)
C1	0.05	0.7	0.3	2.31	90.7	298
C2	0.08	0.7	0.3	2.07	93.1	246
C3	0.10	0.7	0.3	1.98	94.8	214
C4	0.15	0.7	0.3	2.01	97.6	209
C5	0.20	0.7	0.3	2.18	101.2	225
C6	0.10	0.8	0.2	2.22	91.5	238
C7	0.10	0.6	0.4	1.91	99.8	205
C8	0.10	0.5	0.5	1.86	108.9	197
C9	0.10	0.4	0.6	1.82	121.7	190

Table 7. Sensitivity analysis of trust update and TA-AoV weighting parameters.

Configuration	Avg. TA-AoV (↓)	Detection delay (↓)	Energy consumption (↓)
Full framework (TAML)	1.98	214	94.8
No-Trust	2.87	320	101.2
Independent UAV	3.45	410	118.6
No-ZKP	2.10	230	82.3

Table 8. Ablation study of framework components and their impact on performance.

verification delay of unreliable edge servers is more pronounced, as observed with trust-based prioritization. Similarly, a lack of collaborative interdependence leads to duplication of verification and increased energy use due to poor distribution of UAVs. Eliminating the ZKP mechanism makes the framework computationally less expensive, but it doesn't protect privacy-preserving verification as strongly as its presence does, highlighting its significance for privacy in verification. Generally, the entire structure is more efficient than all the ablated structures, and it proves that every element in the structure has a significant value to the operation of the system.

Along with the ablation study, we augment the credibility of the evaluation by sensitivity analysis to explore the contribution of individual components and the impact of key parameters. Through these analyses, it is demonstrated that the proposed framework is stable in performance across different configurations, thus enhancing the explainability and robustness of the reported results. Even though more state-of-the-art baselines might enhance the comparison, the chosen methods provide representative coverage of approaches based on heuristics, optimization, and learning.

Alternative optimization approaches

Although the offered trust-sensitive multi-agent reinforcement learning (TAML) model proves to be highly efficient across various metrics, alternative optimization approaches can be considered based on the conditions

of the given system and its deployment limitations. Heuristic and rule-based algorithms, such as greedy or priority-based scheduling, are computationally inexpensive and make quick decisions, yet are usually less adaptive in dynamic settings.

Theoretically optimal solutions can be determined using optimization-based methods such as convex optimization or mixed-integer programming, but under simplified assumptions, they are usually computationally prohibitive and less practical for real-time decision-making in large, dynamic systems. Other knowledge-based approaches, including variants of deep Q-learning or meta-learners, can be more adaptable or converge more quickly in specific situations. However, such strategies often struggle with continuous action spaces or with coordinating multiple actors. The proposed TAML framework offers a balanced trade-off among adaptability, scalability, and computational efficiency compared to these alternatives and should therefore be used for complex UAV-assisted edge verification problems. A further research opportunity is the exploration of hybrid strategies to integrate methods based on learning and optimization.

Impact of external factors

Various external factors may affect the performance of the proposed framework, though they are not explicitly represented in the current formulation. To start with, the heterogeneity of the task, such as the different computational complexities of generative AI models and the discrepancy in workload across load-balanced edge servers, can be a contributing factor to verification latency and resource-allocation efficiency. Second, network congestion and overhead can have serious effects on UAV-assisted verification, especially in dense deployments where numerous UAVs and edge servers have limited wireless bandwidth. Overload can cause more delays in transmitting proofs and less system responsiveness. Third, the stability of learned policies can decrease due to changes in the environment and operations, e.g. the mobility trends and the dynamic user demand. Although these aspects are abstracted in this simulation at the moment, the suggested framework is scalable, and it is possible to add communication-aware scheduling, adaptive bandwidth distribution, and dynamic workload modeling in future work.

Discussion

Overall, the simulation results show that the suggested trust-aware multi-agent learning (TAML) framework tends to achieve higher performance across all dimensions considered. The proposed solution can address timeliness issues in verification, enforce trust, and improve energy efficiency by integrating cooperative UAV swarms, trust-aware verification metrics, and multi-agent learning, thereby leveraging the strengths of each technique. These findings validate the feasibility of the proposed model for enabling reliable, scalable, and privacy-preserving zero-knowledge verification in dynamic edge generative AI systems.

The proposed framework can withstand some of the typical threats in UAV-assisted systems. GPS spoofing can affect UAV positioning, but cooperative swarms and redundancy can help prevent failure at a single point. Manipulation of sensor data may affect data acquisition, but verification decisions about sensor data use cryptographic proofs based on sources of raw sensor data. Rogue behavior in UAVs, such as false verification reporting, can be avoided through cross-validation and trust-based filtering tools. Also, its use is not limited to adversarial multi-agent manipulation, where agents try to exploit reward structures; this can also be mitigated through reward structure design and centralized training of critics. Although these threats are not directly simulated in the present study, the structure is written in a way that allows it to be extended to other applications and to include these defenses.

Although the assessment is currently simulation-based, it offers an environment with controlled conditions to evaluate the performance of the system in different conditions. Further development will focus on semi-realistic and real-world validation, including code deployment in UAV testbeds, interoperability with actual communication networks, and testing with real-world generative AI workloads. This validation will again assess the practical applicability of the proposed framework and identify system-level limitations that are not captured in the simulation.

Even though the current assessment is simulated, the model is modeled to be used in the real world. Future directions are on validation with realistic communication models, hardware-aware simulations, and UAV testbeds to test performance at realistic constraints such as latency, noise, and environmental variability.

Conclusion

This study explored the problem of safe and reliable validation of edge-deployed generative AI services within the Model-as-a-Service paradigm. To address the shortcomings of current verification methods, we also proposed a joint UAV swarm-based zero-knowledge verification system that enables privacy-preserving verification of generative AI inference while accounting for dynamic trust and resource constraints. By combining zero-knowledge proofs and cooperative UAV swarms, the initiative framework enables edge servers to verify decisions made by the correct model in a non-transparent manner, without exposing proprietary parameters, and to use UAVs as a dynamic verifier that adapts to the network environment. The Age of Verification metric (with a trust analysis component) was introduced, combining verification freshness and server reliability into a single measure, enabling the system to prioritize high-risk, stale edge servers. Moreover, a multi-agent reinforcement learning method that accounts for some degree of trust among agents was developed to organize UAV swarm behavior, enabling decentralized decision-making under long-term energy and mobility constraints.

Simulation results from widely used studies show that the proposed framework has strong potential to outperform state-of-the-art baseline schemes. Specifically, the offered technique registered large decreases in trust-aware Age of Verification and malicious server detection delay, at the same time enhancing the energy efficiency and scalability. These findings show that cooperative swarm intelligence and trust-aware learning are necessary to achieve reliable and efficient zero-knowledge verification in dynamic edge AI setups. There are

some avenues of work that can be further promoted. To begin with, the suggested framework is scalable to more advanced generative AI models, including diffusion models and large transformer-based models, which require more sophisticated zero-knowledge circuit designs. Second, it is possible to consider post-quantum zero-knowledge proof systems, which would improve security in the long term. Third, real-world deployment issues, such as imperfect UAV communication, adversarial UAV behavior, and hardware-in-the-loop experiments, warrant further in-depth study. Lastly, the combination of the proposed verification framework, federated learning, and adaptive model modifications is a promising study direction for building trustworthy and scalable edge AI. Future work will extend the proposed verification framework to support Transformer-based and large-scale generative models, enabling broader applicability to modern generative AI systems.

Data availability

The datasets used and/or analyzed during the current study are available from the corresponding author on reasonable request.

Received: 11 February 2026; Accepted: 10 June 2026

Published online: 21 June 2026

References

1. Xiaoming, Y. et al. Generative AI for the internet of vehicles: A review of advances in training, decision-making, and security. *Mathematics* **13**, 2795 (2025).
2. Abdusalomov, A. et al. Lightweight UAV-based system for early fire-risk identification in wild forests. *Fire* **8**, 288 <https://doi.org/10.3390/fire8080288> (2025).
3. Mohamed Salem, Z. et al. Artificial intelligence driven internet of things framework for wind energy monitoring and performance enhancement in smart cities. *Int. J. Energy Res.* **2025**, 1 <https://doi.org/10.1155/er/7215655> (2025).
4. Samaranayake, L. et al. The transformative role of artificial intelligence in dentistry: A comprehensive overview. Part 1: Fundamentals of AI and its contemporary applications in dentistry. *Int. Dent. J.* **75**, 383–396 <https://doi.org/10.1016/j.identj.2025.02.005> (2025).
5. Makhmudov, F., Kilichev, D., Giyosov, U. Akhmedov, F. Online machine learning for intrusion detection in electric vehicle charging systems. *Mathematics* **13**, 712 <https://doi.org/10.3390/math13050712> (2025).
6. Zhan, S. et al. A review on federated learning architectures for privacy-preserving AI: Lightweight and secure cloud-edge-end collaboration. *Electronics* **14**, 2512 (2025).
7. Haider, Z. A. Surveillance and monitoring in smart cities with UAV and machine learning integration. In *Unmanned Aerial Vehicles Swarm for Protecting Smart Cities: Future Trends and Challenges* (ed. Haider, Z. A.) 241–284 (Springer, 2025).
8. Li, W., Meng, W. & Kwok, L. F. Surveying trust-based collaborative intrusion detection: State-of-the-art, challenges and future directions. *IEEE Commun. Surv. Tutor.* **24**, 280–305 (2021).
9. Guo, J. et al. An online optimization escape entrapment strategy for planetary rovers based on Bayesian optimization. *J. Field Robot.* **41**, 2518–2529 (2024).
10. Alshehri, M. et al. Unmanned aerial vehicle based multi-person detection via deep neural network models. *Front. Neurorobot.* **19**, 1582995 (2025).
11. Yuan, X. et al. Generative AI for the internet of vehicles: A review of advances in training, decision-making, and security. *Mathematics* **13**, 2795 (2025).
12. Altaibek, M. et al. A survey of cross-layer security for resource-constrained IoT devices. *Appl. Sci.* **15**, 9691 (2025).
13. Haider, Z. A., Fayaz, M., Zhang, Y. & Ali, A. Advanced hyperelliptic curve-based authentication protocols for secure internet of drones communication. *ICCK Trans. Adv. Comput. Syst.* **1**, 138–153 (2025).
14. Behura, A., Kumar, A. & Jain, P. K. A comparative performance analysis of vehicular routing protocols in intelligent transportation systems. *Telecommun. Syst.* **88**, 26 (2025).
15. Kong, X. et al. Mobile trajectory anomaly detection: Taxonomy, methodology, challenges, and directions. *IEEE Internet Things J.* **11**, 19210–19231 (2024).
16. Haider, Z. A. A survey on anomaly detection in IoT: Techniques, challenges, and opportunities with the integration of 6g. *Comput. Netw.* **270**, 111484 (2025).
17. Li, L., Cherouat, A., Snoussi, H. & Wang, T. Grasping with occlusion-aware ally method in complex scenes. *IEEE Trans. Autom. Sci. Eng.* **22**, 5944–5954 (2024).
18. Sahu, D. et al. Edge assisted energy optimization for mobile AR applications for enhanced battery life and performance. *Sci. Rep.* **15**, 10034 (2025).
19. Khan, F. M. et al. Intelligent cyber-attack detection for autonomous vehicles using advanced deep learning models. *ICCK Trans. Adv. Comput. Syst.* **1**, 180–192 (2025).
20. Wan, A. et al. Real-time aero-engine fault diagnosis using 5g edge computing and deep learning. *Measurement* **257**, 118784 (2026).
21. Hady, M. A., Hu, S., Pratama, M., Cao, Z. & Kowalczyk, R. Multi-agent reinforcement learning for resources allocation optimization: A survey. *Artif. Intell. Rev.* **58**, 354 (2025).
22. Cao, H., Hu, Q., Jiang, H., Ge, S. S. & Li, D. Predefined-sequential-synchronized control with exploration of event-triggered mechanism. *Automatica* **183**, 112645 (2026).
23. Wu, T. et al. Joint UAV deployment and edge association for energy-efficient federated learning. *IEEE Trans. Cogn. Commun. Netw.* **11**, 4126–4140 (2025).
24. FengYing, Y., Din, A., HuiChao, L., Babar, M. & Ahmad, S. Decentralized consensus in robotic swarm for collective collision and avoidance. *IEEE Access* **12**, 72143–72154 (2024).
25. Akram, J., Anaissi, A., Rathore, R. S., Jhaveri, R. H. & Akram, A. Galtrust: Generative adversarial learning-based framework for trust management in spatial crowdsourcing drone services. *IEEE Trans. Consum. Electron.* **70**, 6196–6207 (2024).
26. Wan, P., Li, H., Zhang, Z., Duan, C. & Wang, J. Motion parameter estimation of near-field drone based on DNSP under sensor position error. *IEEE Wirel. Commun. Lett.* **15**, 1613–1617 (2026).
27. Ghourab, E. M. Extended reality-aware wireless communication networks: A systematic literature review. *IEEE Open J. Commun. Soc.* <https://doi.org/10.1109/OJCOMS.2025.3599091> (2025).
28. Xiong, J.-J. & Wang, X.-Y. Adaptive sliding mode control for uncertain tilting quadrotors using randomized feedforward neural networks. *Int. J. Robust Nonlinear Control* <https://doi.org/10.1002/rnc.70396> (2026).
29. Zuccotto, M., Castellini, A., Torre, D. L., Mola, L. & Farinelli, A. Reinforcement learning applications in environmental sustainability: A review. *Artif. Intell. Rev.* **57**, 88 (2024).
30. Xue, B. et al. Perturbation defense ultra high-speed weak target recognition. *Eng. Appl. Artif. Intell.* **138**, 109420 (2024).
31. Wandelt, S. & Zheng, C. Toward smart skies: Reviewing the state of the art and challenges for intelligent air transportation systems (IATS). *IEEE Trans. Intell. Transp. Syst.* **25**, 12943–12953 (2024).

32. Li, S. Distributed bearing-based fault-tolerant formation control of fixed-wing UAV swarm with prescribed performance. *Aerosp. Sci. Technol.* **168**, 110897 (2025).
33. Kanupriya, C. I. & Goyal, R. K. Computation offloading techniques in edge computing: A systematic review based on energy, QoS and authentication. *Concurr. Comput. Pract. Exp.* **36**, e8050 (2024).
34. Xu, S., Zhang, J., Meng, S. & Xu, J. Task allocation for unmanned aerial vehicles in mobile crowdsensing. *Wireless Netw.* **30**, 3707–3719 (2024).
35. Behura, A., Kumar, A. & Jain, P. K. A multi-objective approach for secure cluster based routing & attack classification in vanets. *Peer-to-Peer Netw. Appl.* **18**, 1–40 (2025).
36. Haider, Z. A., Rahman, M. M., Khan, M. A. & Sohail, Q. Optimizing collaborative task allocation in internet of vehicles (IOV) through blockchain-enabled incentive mechanisms. *ICCK Trans. Sens. Commun. Control* **2**, 147–167 (2025).
37. Zhang, F. et al. Breaking the edge: Enabling efficient neural network inference on integrated edge devices. *IEEE Trans. Cloud Comput.* **13**, 694–710 (2025).
38. Jiang, H. Cooperative content caching in vehicular edge computing networks: A two-stage deep reinforcement learning approach. *IEEE Trans. Mob. Comput.* **25**, 1–18 (2026).
39. Ogenyi, F. C., Ugwu, C. N. & Ugwu, O. P.-C. Securing the future: Ai-driven cybersecurity in the age of autonomous IoT. *Front. Internet Things* **4**, 1658273 (2025).
40. Guo, Y., Huang, C. & Liu, R. Development of an attention mechanism for task-adaptive heterogeneous robot teaming. *AI* **5**, 555–575 (2024).
41. Ali, G. et al. Blockchain and deep q-learning for trusted cloud-enabled drone network in smart forestry: A survey. *Babylonian J. Netw.* **2025**, 207–241 (2025).
42. Zhang, T. et al. Toward attack-resistant route mutation for vanets: An online and adaptive multiagent reinforcement learning approach. *IEEE Trans. Intell. Transp. Syst.* **23**, 23254–23267 (2022).
43. Ren, J. & Qin, T. Decentralized blockchain-based and trust-aware task offloading strategy for healthcare IoT. *IEEE Internet Things J.* **11**, 829–847 (2023).
44. Haider, Z. A., Ullah, I., Shareha, A. A., Nasimov, R. & Memon, S. A. Artificial intelligence (ai)-enabled unmanned aerial vehicle (UAV) systems for optimizing user connectivity in sixth-generation (6g) ubiquitous networks. *Comput. Mater. Continua* **86**, 1 (2026).
45. Yin, R. et al. Radar-assisted predictive beamforming for UAV-aided networks: A deep-learning solution. *IEEE Trans. Veh. Technol.* **74**, 16079–16093 (2025).
46. Zhang, Y., Jiang, D., Zhang, P., Guizani, M. & Kostromitin, K. I. Adaptive resource management in edge networks: Task-aware virtual network embedding. *IEEE Trans. Veh. Technol.* <https://doi.org/10.1109/TVT.2025.3628126> (2025).
47. Yin, R. et al. Intelligent 3d trajectory and resource control for multi-UAV 6g networks via GNN and deep unfolding. *IEEE Trans. Commun.* **1**, 1–1 (2026).
48. Wang, Q. & Wang, Z. Blockchain-enhanced IoT sensor data management for engineering monitoring. *IEEE Sens. J.* **25**, 7363–7375 (2024).
49. Xu, W. et al. Blockchain-based verifiable decentralized identity for intelligent flexible manufacturing. *IEEE Internet Things J.* **12**, 32366–32378 (2025).
50. Wang, Q., Wang, Z. & Wang, W. Research on secure cloud networking plan based on industry-specific cloud platform. *IEEE Access* **11**, 51848–51860 (2023).
51. Wang, Q. Categorizing and efficient handling for cellular network interference. *IEEE Wirel. Commun.* <https://doi.org/10.1109/MWC.2025.3621567> (2025).

Author contributions

Conceptualization, Kuldashbay Avazov; Data curation, Kudratjon Zohirov; Formal analysis, Kudratjon Zohirov; Funding acquisition, Akmalbek Abdusalomov, Abror Buriboev and Heung Seok Jeon; Investigation, Umidjon Ruziev; Methodology, Kuldashbay Avazov; Project administration, Akmalbek Abdusalomov, Abror Buriboev and Heung Seok Jeon; Resources, Umidjon Ruziev, Nabijon Abduazizov, Akmalbek Abdusalomov and Heung Seok Jeon; Software, Kuldashbay Avazov; Supervision, Heung Seok Jeon; Validation, Kuldashbay Avazov, Nabijon Abduazizov, and Akmalbek Abdusalomov; Visualization, Akmalbek Abdusalomov and Heung Seok Jeon; Writing – original draft, Kuldashbay Avazov; Writing – review & editing, Akmalbek Abdusalomov, Abror Buriboev and Heung Seok Jeon.

Funding

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. RS-2024-00412141). And This paper was also supported by Konkuk University in 2025

Declarations

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to H.S.J.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

© The Author(s) 2026