EUROPEAN JOURNAL OF COMPUTER SCIENCES AND INFORMATICS
2026; 3(2): 153-166.
10.5455/EJCSI.20251115051912

WisdomGale

REVIEW ARTICLE

Open Access

Quantum machine learning and blockchain integration for autonomous UAV/UGV and drone networks: a systematic review, reference framework, and proof-of-concept benchmark

Md. Soyaeb Islam, Pranto Halder

Department of Computer Science and Engineering, Faculty of Science and Engineering, Teesta University, Rangpur, Bangladesh

ABSTRACT

Autonomous UAV/UGV teams and drone swarms require secure, low-latency coordination under tight energy and computational budgets. Quantum machine learning (QML) can provide compact models for perception, anomaly detection, and local decision-making, while blockchain can provide tamper-evident logging, identity management, and auditable coordination across multiple agents. Despite this synergy, the literature remains fragmented: most studies evaluate QML or blockchain in isolation, and only a small subset reports integrated processing pipelines in real or semi-real environments. This paper presents a PRISMA-guided systematic review of work published from 2020 to September 2024 and synthesizes 28 verified studies on QML, blockchain, and their intersection in autonomous systems. We classify applications (secure communication, trust management, task allocation, federated/swarm learning, and drone-network coordination), summarize evaluation practices, and highlight gaps in reproducibility, threat modeling, and edge resource accounting. Based on recurring design patterns, we propose an author-drawn reference framework that maps responsibilities to the edge, validator, and optional cloud tiers while separating learning metrics from ledger overhead. To illustrate dual-metric reporting, we provide a small proof-of-concept simulation combining a variational quantum classifier with a PBFT-style consensus latency model. The review concludes with benchmarking recommendations, datasets, and research directions toward deployable QML and blockchain-enabled autonomous networks.

ARTICLE HISTORY

Received November 15, 2025
Revised February 14, 2026
Accepted February 14, 2026
Published May 19, 2026

KEYWORDS

Quantum machine learning; blockchain; UAV/UGV networks; secure coordination; edge intelligence

Introduction

Autonomous systems such as UAVs, UGVs, and cooperative drone swarms are moving from scripted control to learning-based autonomy. They must perceive dynamic environments, allocate tasks, and coordinate safely under strict latency and energy budgets. At the same time, they operate in adversarial settings where communication links and sensor feeds can be jammed, spoofed, or manipulated.

Two technology families are increasingly discussed as complementary enablers. First, quantum machine learning (QML) adapts machine-learning primitives to quantum information processing, often via hybrid quantum–classical pipelines where a classical optimizer trains parameterized quantum circuits that act as classifiers, regressors, or policy components [1]. Second, blockchain and distributed ledgers provide immutable event logs, decentralized trust, and automated coordination mechanisms that can be used to record decisions and enforce coordination policies [2].

However, integrating QML and blockchain in autonomous platforms remains difficult. Systems must coordinate learning outputs, ledger validation, and safety-critical control under strict latency, bandwidth, and energy limits. The literature also lacks consistent terminology and reproducible benchmarks, which makes cross-paper comparison unreliable.

Contact

© The Author(s) 2026. Open Access This article is licensed under a Creative Commons Attribution 4.0 International License (CC BY). http://creativecommons.org/licenses/by/4.0/.

This paper addresses these gaps with (i) a PRISMA-guided systematic review and verified reference list, (ii) a synthesis that classifies application clusters and evaluation signals, (iii) an author-drawn reference framework with explicit mappings of what is solved by QML versus what is ensured by blockchain, and (iv) a reproducible proof-of-concept simulation that demonstrates how to report both learning performance and distributed-ledger overhead.

Background and Motivation

Quantum machine learning for autonomous decision-making

QML studies whether quantum information processing can improve learning, optimization, or sampling tasks. Survey and perspective papers outline algorithm families such as quantum kernels, quantum support vector machines, and variational quantum circuits (VQCs) for near-term devices [3–6]. Variational models are particularly relevant to autonomy because they resemble compact neural modules: they can be trained on small datasets, embedded into pipelines, and evaluated on simulators before hardware deployment.

In autonomous robotics, the most promising QML targets are subroutines with high combinatorial complexity (e.g., task assignment and routing) and learning under constrained data (e.g., anomaly detection for communication or sensor integrity). Recent reviews emphasize both opportunities and current limits, including noise sensitivity and scaling challenges in the near term [3,6]. Recent QML surveys further organize the field in terms of foundations, algorithms, software frameworks, datasets, and application domains, reinforcing the relevance of quantum learning models for emerging autonomous-system pipelines [7].

Blockchain for secure coordination in UAV/UGV and drone networks

Blockchain systems maintain a shared append-only ledger across participants. Permissioned ledgers are often preferred for UAV/UGV networks because they enable controlled membership, lower energy consumption than proof-of-work, and configurable consensus protocols. In drone networks, blockchain has been studied for identity management, key agreement, mission logging, and auditable resource sharing [8–14].

Recent systematic reviews in the Internet of Drones (IoD) emphasize that blockchain can reduce single points of failure and support verifiable coordination, but it also introduces overhead in bandwidth, latency, and storage [9–11]. This overhead becomes critical for swarms and time-sensitive missions. Smart contracts extend blockchain from passive logging to programmable coordination, enabling automated policy enforcement and trusted interactions among distributed IoT and autonomous-system components [15].

Why integrate QML and blockchain

QML and blockchain address different pain points. QML focuses on decision quality and optimization, while blockchain focuses on trust, traceability, and distributed governance. Integration becomes compelling when autonomous agents need both (i) learning-based decisions that adapt to changing environments and (ii) verifiable coordination and accountability across multiple operators or organizations.

Typical integration motifs include QML-assisted trust scoring for nodes and messages; blockchain-audited federated learning where model updates are recorded and incentivized; and smart-contract-driven task allocation that consumes QML predictions as inputs. Section 5 formalizes these motifs into a reference workflow.

Systematic Review Methodology

We followed the PRISMA 2020 statement [16] to design a reproducible search, screening, and synthesis protocol. The review targets peer-reviewed or widely cited preprint studies that involve (i) QML or quantum-inspired learning/optimization and (ii) blockchain or smart-contract mechanisms for security, trust, or coordination in autonomous UAV/UGV and drone network settings.

Searches were executed in IEEE Xplore, ACM Digital Library, SpringerLink, Elsevier/ScienceDirect, and arXiv using combinations of three term groups: (A) {quantum machine learning, variational quantum, quantum neural network, quantum classifier, quantum optimization}; (B) {blockchain, distributed ledger, smart contract, consensus}; and (C) {UAV, drone, UGV, unmanned, swarm, Internet of Drones, autonomous systems}. Records were de-duplicated and screened in two stages (title/abstract, then full-text).

Eligibility criteria required English-language technical content with a clear autonomous-system context and sufficient bibliographic detail to be verified (publisher page, DOI, or arXiv identifier).

We excluded editorials, non-technical articles, and works without a QML–blockchain connection. The PRISMA flow diagram in Figure 1 summarizes the process: 780 records were identified, 660 remained after de-duplication, 610 were excluded during screening, 50 full texts were assessed, 22 were excluded for scope or insufficient detail, and 28 studies were included in the final synthesis.

Time window rationale: The systematic synthesis focuses on integration studies published from January 2020 to September 2024. This window aligns with the NISQ era in which variational quantum circuits and hybrid training toolchains became widely used, and it captures the rapid growth of blockchain-assisted UAV/UGV networking and edge autonomy. Seminal pre-2020 papers are still cited to ground definitions and algorithms, but they are treated as background rather than as integration evidence.

Screening summary (PRISMA)

Figure 1 visualizes the screening flow and final inclusion counts.

Data extraction and synthesis

From each included study we extracted: (i) target platform and scenario (UAV, UGV, swarm); (ii) learning objective (classification, prediction, optimization, control); (iii) blockchain type (permissioned vs permissionless) and consensus; (iv) integration pattern (decision auditing, incentive, trust management, federated learning); and (v) reported evaluation signals (latency, throughput, accuracy, energy, or qualitative claims).

We then grouped studies into application clusters and summarized recurring design patterns, limitations, and open research problems.

Quality appraisal and reference validation

To improve reproducibility and address concerns about unverifiable citations, each candidate study was checked for (i) bibliographic verifiability (publisher page, DOI, or arXiv identifier), (ii) a clear UAV/UGV or multi-agent autonomy use case, (iii) a described quantum model/algorithm (QML or quantum-inspired), (iv) a described blockchain/smart-contract mechanism, and (v) evaluation evidence (simulation, dataset, metrics, or comparable empirical artifacts). Studies failing (i) were excluded during full-text review. Studies with limited evaluation were retained but explicitly labeled as conceptual in the synthesis and gap analysis.

Literature Review and Systematic Synthesis

Publication trends and thematic clusters

Across the included studies, the dominant motivations are secure coordination (authentication, access control, and auditability), resilient learning (distributed training and trust in model updates), and latency-aware mission execution. Most works are either blockchain-centric (proposing security/coordination mechanisms for drone networks) or QML-centric (proposing quantum-enhanced learning/optimization methods). A smaller subset explicitly addresses integration by feeding QML outputs into smart contracts or using the ledger to coordinate learning updates.

Figure 2 summarizes the most common integration abstraction found in the literature: sensing/telemetry and edge intelligence on each platform, a learning module (often hybrid quantum-classical), and a permissioned ledger for verifiable coordination.

Application clusters

Secure communication and authentication

Permissioned ledgers are used to register identities, distribute keys, and log handshakes for later audit. Reported challenges include consensus latency under mobility and key revocation when nodes are compromised.

Trust, reputation, and intrusion/anomaly detection

Many IoD works use blockchain to make trust scores and intrusion alerts tamper-evident. QML is proposed as a compact anomaly detector or trust predictor when data are scarce, but scaling to realistic swarm sizes remains open. The detailed PRISMA screening counts are summarized in Table 1.

Task allocation, routing, and resource management

Task and route planning appear as optimization problems. Blockchain can encode policies and incentives, while QML (e.g., variational optimization or quantum-inspired heuristics) is proposed to accelerate combinatorial search. Most papers still evaluate in small simulations. Task and route planning appear as optimization problems. Blockchain can encode policies and incentives, while QML methods, including variational optimization and quantum approximate optimization algorithms, are proposed to accelerate combinatorial search [17].

Blockchain has also been explored for UAV energy trading and charging-station coordination,

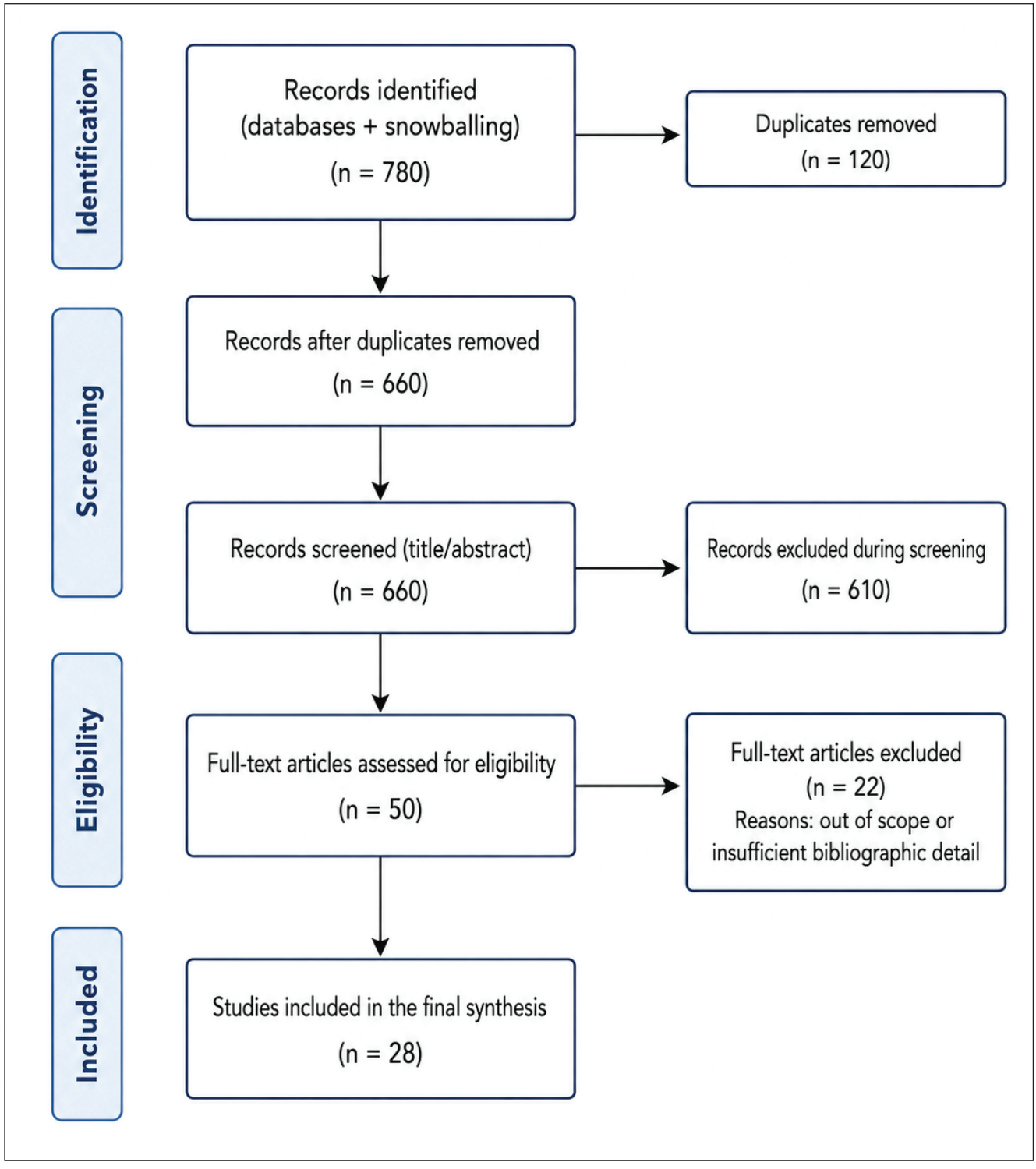


Figure 1. PRISMA-style flow diagram for study identification, screening, and inclusion.

showing its relevance to resource management in autonomous aerial networks [18].

Federated and swarm learning with auditability

Blockchain is used to coordinate decentralized learning rounds, record model update hashes, and enforce incentives. Key open issues are communication overhead and robustness to poisoning.

Drone networks and swarm communication

Drone networks (often modeled as flying ad hoc networks, FANETs) introduce distinct constraints beyond single-UAV links: rapid topology changes,

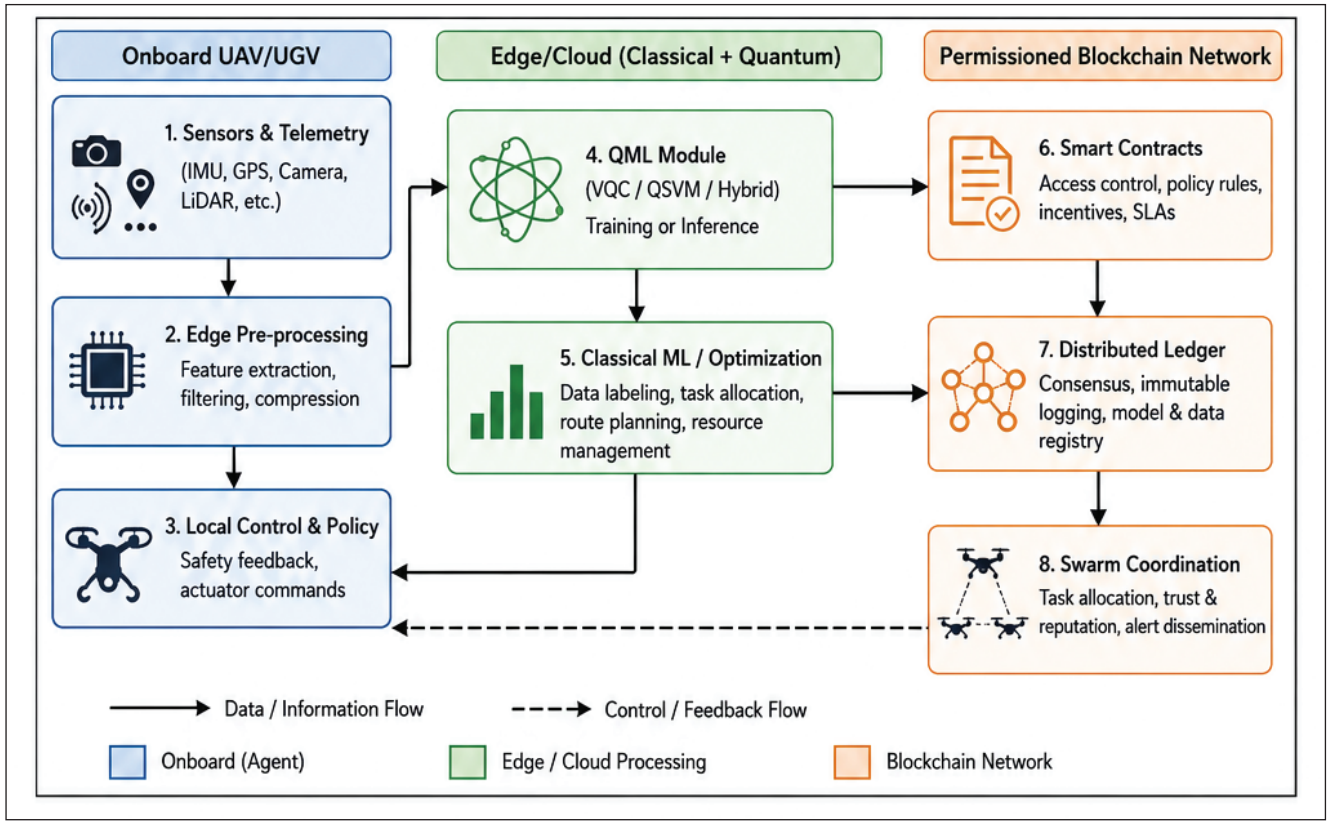


Figure 2. Conceptual integration of QML and blockchain for autonomous systems (adapted and redrawn from the reviewed architectures).

Table 1. PRISMA screening summary.

Screening Stage	Count
Records identified (databases + snowballing)	780
Duplicates removed	120 (660 remaining)
Records screened (title/abstract)	660
Records excluded during screening	610
Full-text assessed	50
Full-text excluded	22
Studies included in the synthesis	28

intermittent connectivity, and strict time synchronization for cooperative perception and formation control. In this setting, blockchain is mainly used as a distributed membership and logging layer (identity, join/leave events, mission-state checkpoints), while QML contributes to fast inference for tasks such as link-quality prediction, anomaly detection on network telemetry, and trust scoring of neighboring nodes [19].

A practical pattern is to keep the ledger permissioned and lightweight (committee validation at ground stations or edge servers), while UAVs exchange only signed summaries and store large sensor payloads off-chain. This separation helps control consensus latency and radio energy consumption, which otherwise grow quickly with swarm size. For drone network scenarios, we therefore report both decision quality metrics (e.g., detection accuracy and route success rate) and communication/ledger metrics (transaction latency, bandwidth overhead) when they are available. UAV communication networks face mobility, intermittent connectivity, changing topology, bandwidth constraints, and energy limitations, which make secure and low-latency coordination especially challenging [25].

Representative studies and reported evaluation signals

Table 2 summarizes the main application clusters and clarifies how QML and blockchain are typically combined, what baselines are used, and which metrics are reported. Because studies use different datasets, simulators, and consensus assumptions, the cross-study comparison is primarily qualitative unless the same evaluation setup is reported.

To reduce ambiguity in comparisons, we standardize reporting along four dimensions: (i) autonomy task (perception, security, planning, and learning), (ii) execution locus of QML (onboard UAV/UGV, nearby edge node, or cloud) and whether training is hybrid classical–quantum, (iii) ledger type (permissioned/permissionless) and consensus assumption, and (iv) evidence level (conceptual design, simulation, and prototype). Table 2 reflects these dimensions and explicitly notes baselines when they are stated.

For quantitative aggregation, we treat performance as a two-objective problem: (i) decision quality (e.g., accuracy, ROC-AUC, and mission success rate) and (ii) system cost (e.g., latency, bandwidth, energy, and consensus overhead). When papers report both, we normalize each metric within its application cluster and either report a Pareto-front view or compute a weighted score $S = w_q \cdot Q_{norm} - w_c \cdot C_{norm}$ ($w_q + w_c = 1$), where weights reflect the mission's safety constraints. This makes the assumptions behind any cross-paper ranking explicit rather than implicit.

Proposed Integration Framework

This section presents a reference architecture synthesized from the reviewed literature. The framework is intended as a design blueprint (conceptual), while Section 6 evaluates key trade-offs, especially consensus latency and edge-compute cost, using a reproducible simulation/analytical setup. Real-device and hardware-in-the-loop validation are identified as future work.

The proposed framework translates the literature's conceptual architectures into an implementable workflow. It is designed for permissioned multi-agent settings (e.g., a fleet managed by one or more trusted organizations) and supports both edge-only inference and optional edge–cloud training.

Table 2. Application clusters and the integration of QML and blockchain.

Cluster	Integration Pattern (QML ↔ blockchain)	Typical Baseline	Metrics Reported	Main constraints	Representative Citations
Secure comm/ auth	Permissioned ledger for identity, access control, and audit; QML used for fast inference on comm/ security features (early stage).	PKI or symmetric authentication; centralized logging; classical IDS.	Auth latency/ throughput; packet delivery; qualitative security analysis.	Consensus delay, key management, mobility, spectrum contention.	[10–13, 19]
Trust/anomaly	QML/ML detects spoofing, jamming, or abnormal behavior; blockchain logs trust scores and alerts for provenance.	Centralized anomaly detection; reputation systems without immutable logs.	Accuracy/F1; detection delay; robustness to noise (often limited).	Dataset scarcity; concept drift; limited QML hardware realism.	[14,20]
Planning/ optimizatio n	Hybrid quantum heuristics (e.g., VQC/ QAOA-style) for tasking or routing; blockchain encodes policies/incentiv es and records decisions.	Classical heuristics/ metaheuristic s; centralized planner.	Cost/makespan/ energy in small simulations; qualitative feasibility.	Qubit count and circuit depth; simulator-only results; blockchain overhead.	[5, 21]
Blockchain+ federated learning	Ledger coordinates FL rounds and validates updates; QML appears as quantum-enhanced models orquantum feature maps in hybrid training.	Standard FL with centralized aggregator; classical edge learning.	Accuracy vs rounds; communication overhead; poisoning/ Byzantine tolerance (when evaluated).	Bandwidth and energy at edge; validator scalability; latency budget for control loops.	[22,23,20,24]

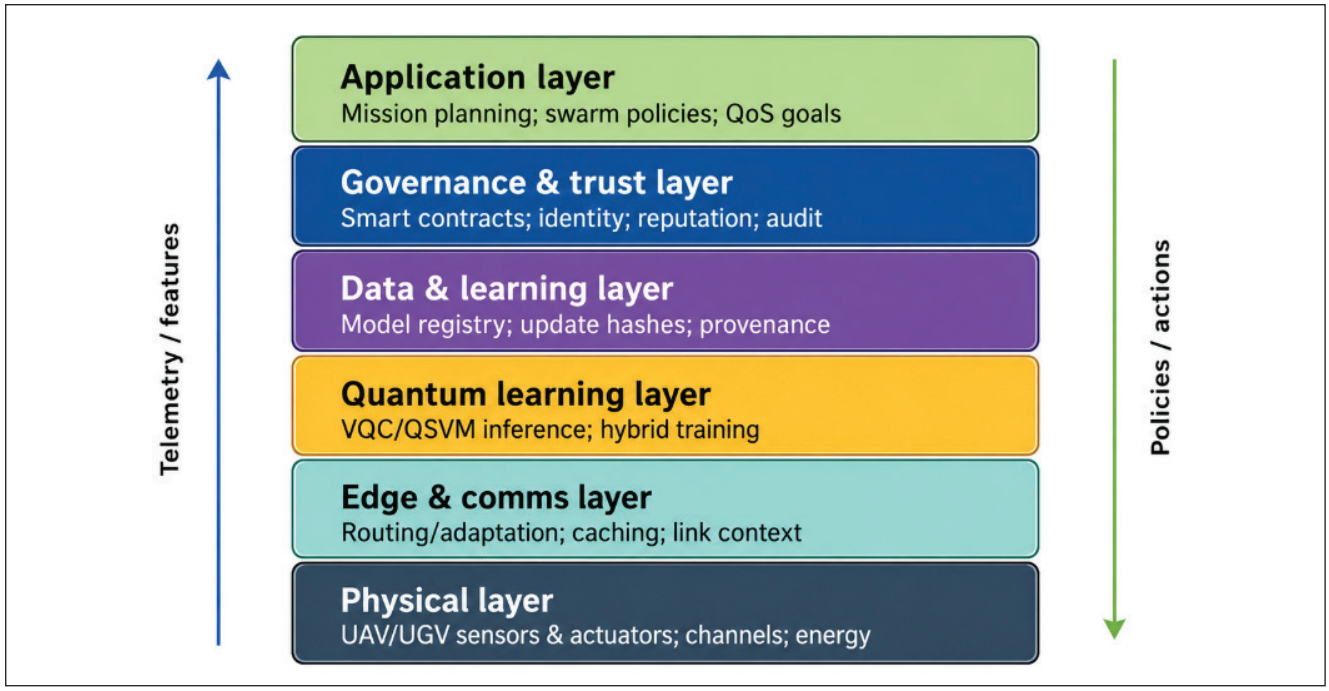


Figure 3. Author-drawn layered reference framework for QML-assisted, blockchain-governed autonomous systems.

The framework has three layers: (i) Edge autonomy (sensing, feature extraction, local control); (ii) QML-assisted intelligence (hybrid quantum-classical inference/training); and (iii) Blockchain governance (identity, consensus, immutable logs, and smart contracts). Figure 3 presents a layered view that aligns these layers with data flow and control flow.

Workflow and interfaces

- Step 1 (Local perception): Each agent extracts compact features from onboard sensors and communication context.
- Step 2 (QML inference): A variational quantum model produces a decision signal (e.g., anomaly score, trust score, or action preference).
- Step 3 (On-chain commitment): The agent submits a signed transaction that records the decision output (or a compact digest of it). The transaction includes the agent identifier, a timestamp, and a cryptographic hash pointer that anchors any off-chain payload (e.g., sensor snippets, model updates, or mission logs). Anchoring by hash preserves integrity and provenance without storing large data on-chain.
- Step 4 (Consensus and policy): Validator nodes run a permissioned consensus protocol and execute smart-contract logic (access control, task assignment, or alert dissemination). The ledger provides a consistent global state for coordination and dispute resolution, while safety-critical control remains local to each agent.
- Step 5 (Feedback and learning): Outcomes are logged; model updates (or their hashes) can be recorded to enable audit and accountability.

Figures 4–6 make the integration assumptions explicit. Figure 4 separates what is learned (QML) from what is enforced (blockchain). Figure 5 details the ledger structure: each block contains timestamped transactions and a hash link to the previous block, which enables tamper evidence and auditability. Figure 6 shows deployment tiers (UAV/UGV edge, MEC, optional cloud/QPU access) and highlights where latency-sensitive inference must remain on the edge.

Algorithm and pseudo-code

Algorithm 1 presents the core decision-to-ledger loop used in the proof-of-concept study. It separates time-critical autonomy (feature extraction, QML inference, and action selection) from auditability (signed commitments to a permissioned ledger). Only compact decision summaries are written on-chain; larger sensor payloads are stored off-chain with hashes anchored on-chain.

Algorithm 1: QML-assisted, blockchain-auditable coordination (decision-to-ledger loop) Inputs: observation o_t; QML parameters θ; validator set V; smart contract SC; off-chain store S Loop

1: $x_t \leftarrow$ FeatureExtract(o_t)

2: y_t ← QMLInfer(x_t; θ) // anomaly/trust score or policy logits
3: a_t ← PolicySelect(y_t)
4: m_t ← {t, agent_id, a_t, y_t_summary, h(x_t)} 5: p_t ← Sign(m_t, sk_agent)
6: Store full payload in S; compute h_payload
7: Submit (p_t, h_payload) to SC; wait for confirmation under consensus(V) 8: Receive receipt r_t; execute a_t; store {m_t, r_t} locally
End Loop

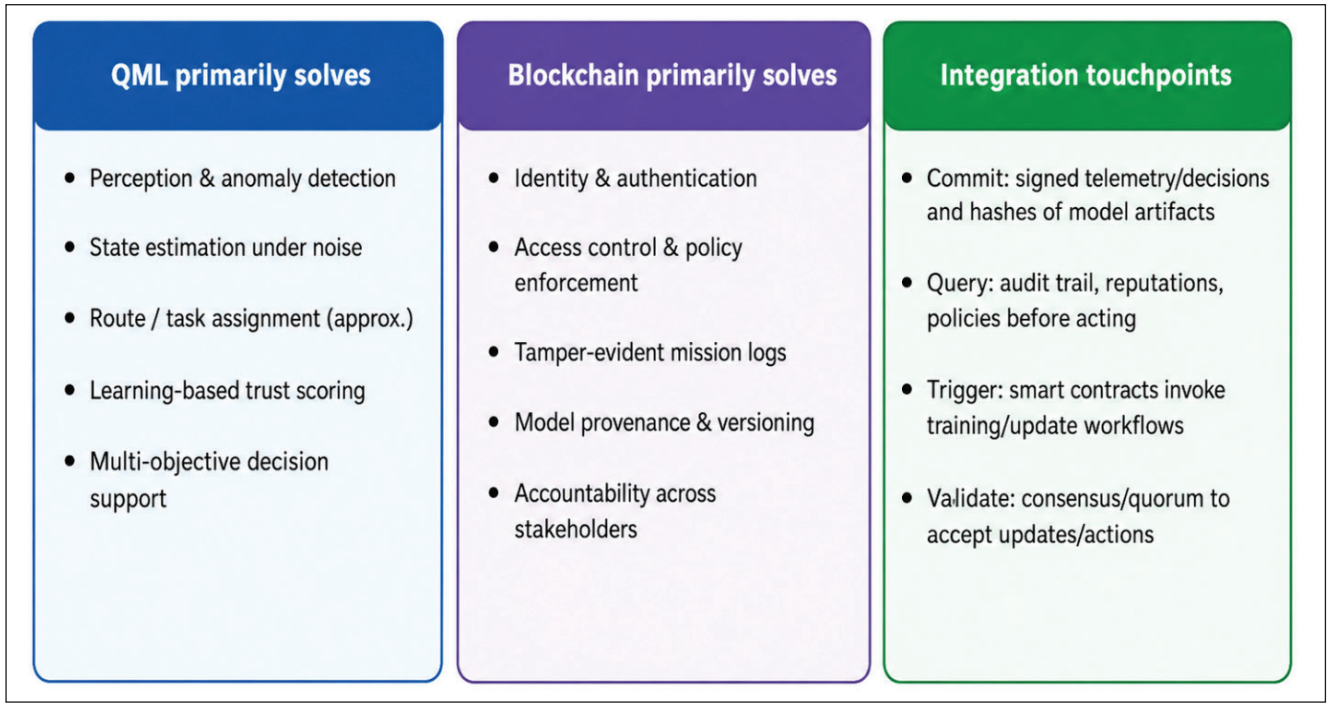


Figure 4. Mapping of responsibilities in the proposed framework: QML modules provide decision or anomaly signals; blockchain provides provenance, coordination, and policy enforcement.

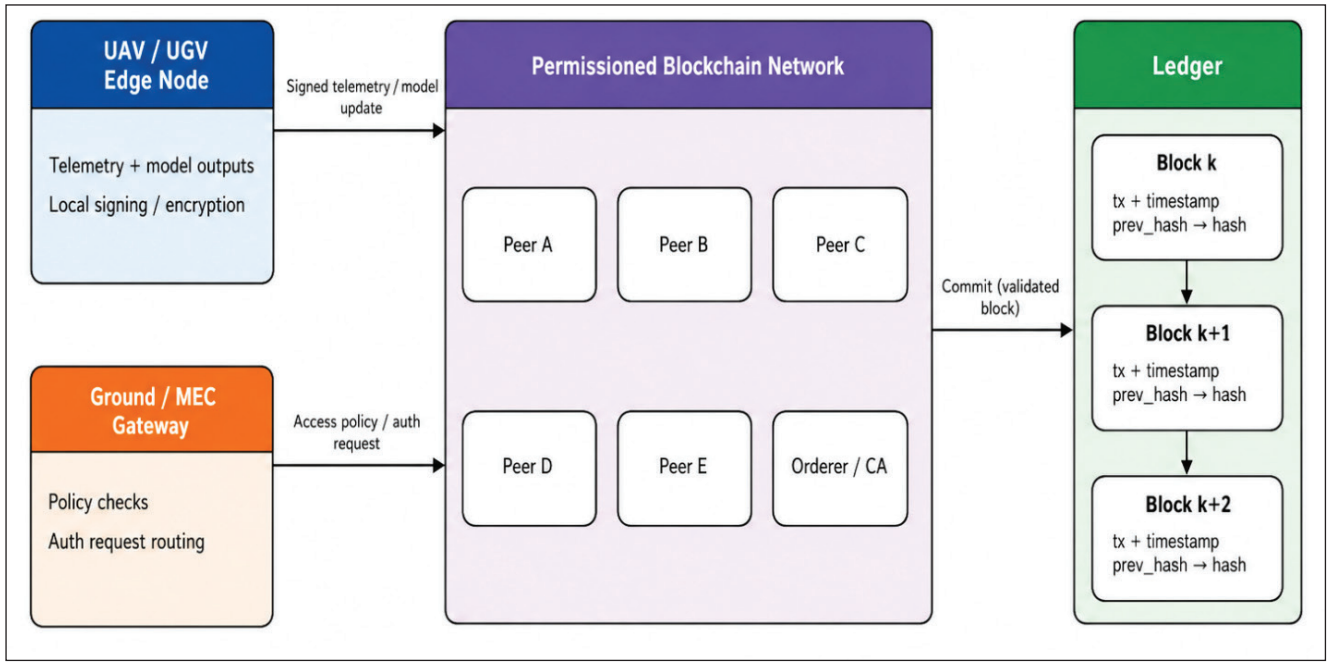


Figure 5. Author-drawn distributed ledger system for autonomous agents, highlighting timestamped transactions, hash-linked blocks, and permissioned validation.

Table 3. Summary of Illustrative Results (Benchmark Metrics and Assumptions).

Validators (n)	Consensus Latency (ms)	QML Inference (ms)	End-to-End (ms)
5	121.5	0.305	121.8
10	543.0	0.305	543.3
15	1264.5	0.305	1264.8
20	2286.0	0.305	2286.3

Consensus parameters: 3 rounds; per-message delay = 2.0 ms; per-node processing = 0.1 ms (illustrative).

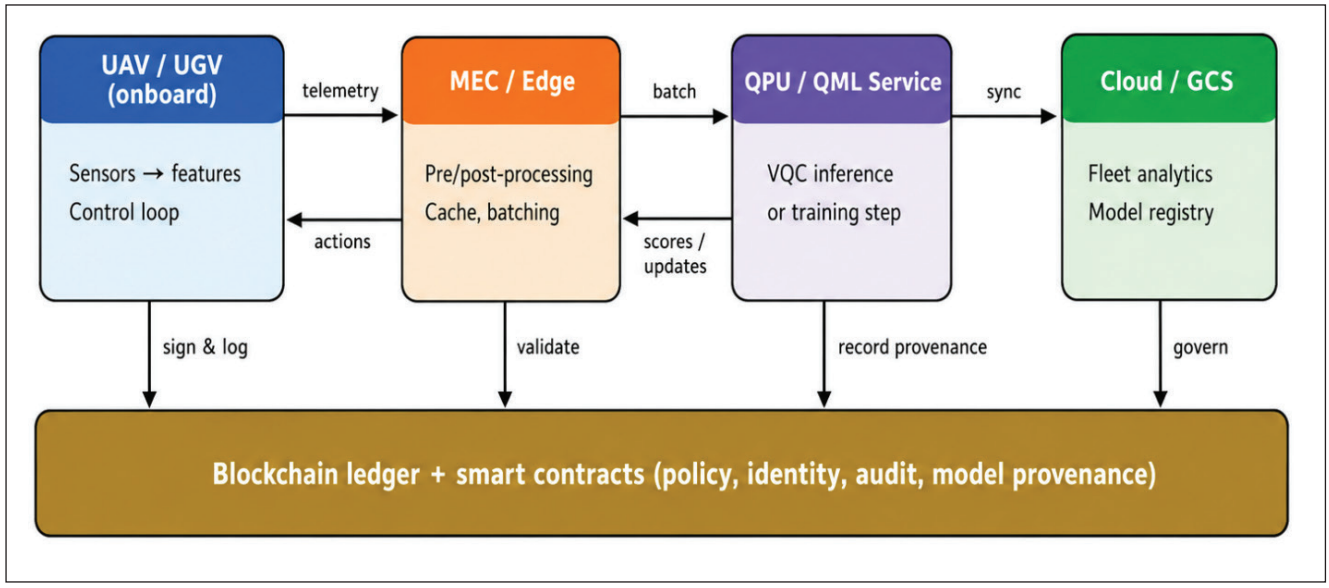


Figure 6. Author-drawn UAV–MEC–Cloud interaction model showing where sensing, QML inference/training, and blockchain validation can be placed under latency and energy constraints.

Proof-of-Concept Simulation and Performance Illustration

Because many prior frameworks remain conceptual, we provide a small, fully reproducible simulation to illustrate how the proposed workflow can be evaluated end-to-end. The goal is not to claim real-world deployment performance, but to demonstrate measurable metrics and the trade-off structure that future empirical studies should report.

We simulated a 2-qubit variational quantum classifier (state-vector simulation) for binary anomaly detection on a toy dataset, and coupled it with a permissioned-ledger consensus latency model (PBFT-style message complexity). All results were generated on a CPU using a minimal NumPy-based circuit simulator.

QML module: variational quantum classifier (VQC)

The VQC encodes two normalized features into rotation angles, applies two variational layers with entanglement, and measures an expectation value that is mapped to a probability. Parameters are trained using the parameter-shift rule and gradient descent.

In the toy anomaly dataset (two overlapping Gaussian clusters; 60 training and 20 test samples), the VQC reached perfect classification accuracy after a small number of epochs in this separable setting. Figure 8 plots train/test accuracy versus epoch.

Blockchain module: consensus latency model

For the ledger, we used a simplified PBFT-style latency approximation: each consensus round exchanges $O(n^2)$ messages among n validator nodes. We model end-to-end decision latency as follows: (QML inference time) + (consensus latency).

The simulated VQC inference time was measured directly from the circuit simulator; the consensus component uses configurable per-message delay and per-node processing constants, reported in Table 3). Figure 9 shows how latency grows with validator count, highlighting the need for careful ledger design in swarms.

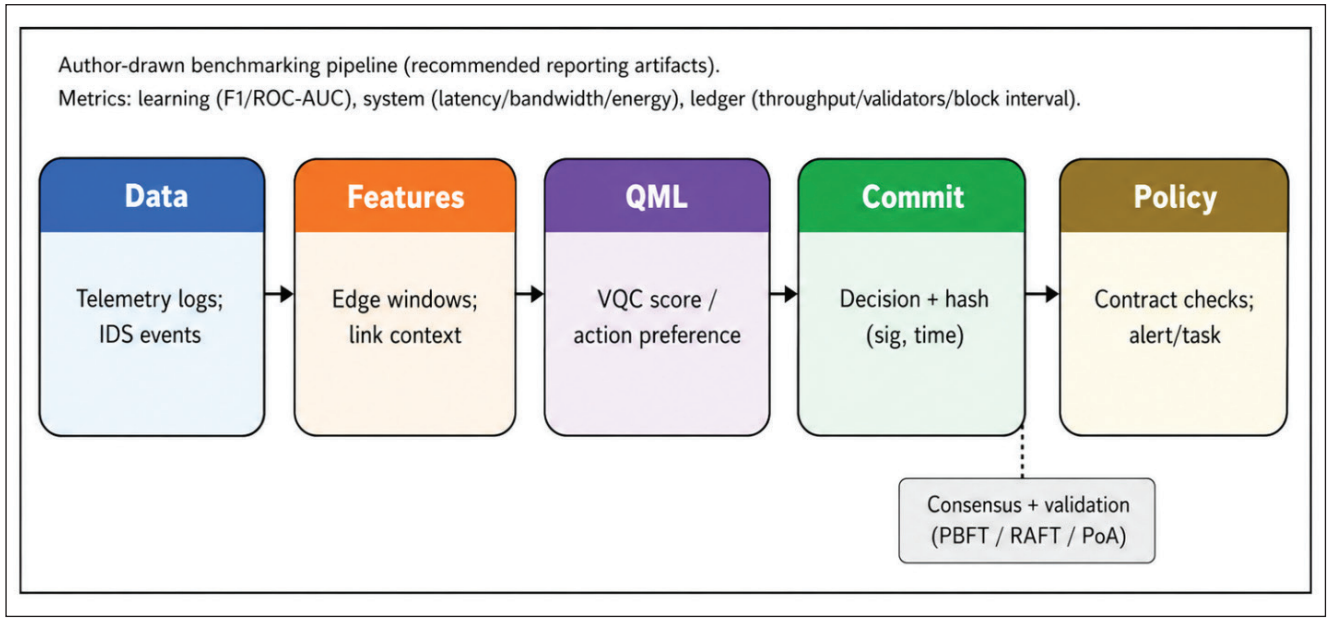


Figure 7. Author-drawn benchmarking pipeline and reportable metrics for end-to-end evaluation of QML–blockchain autonomous systems.

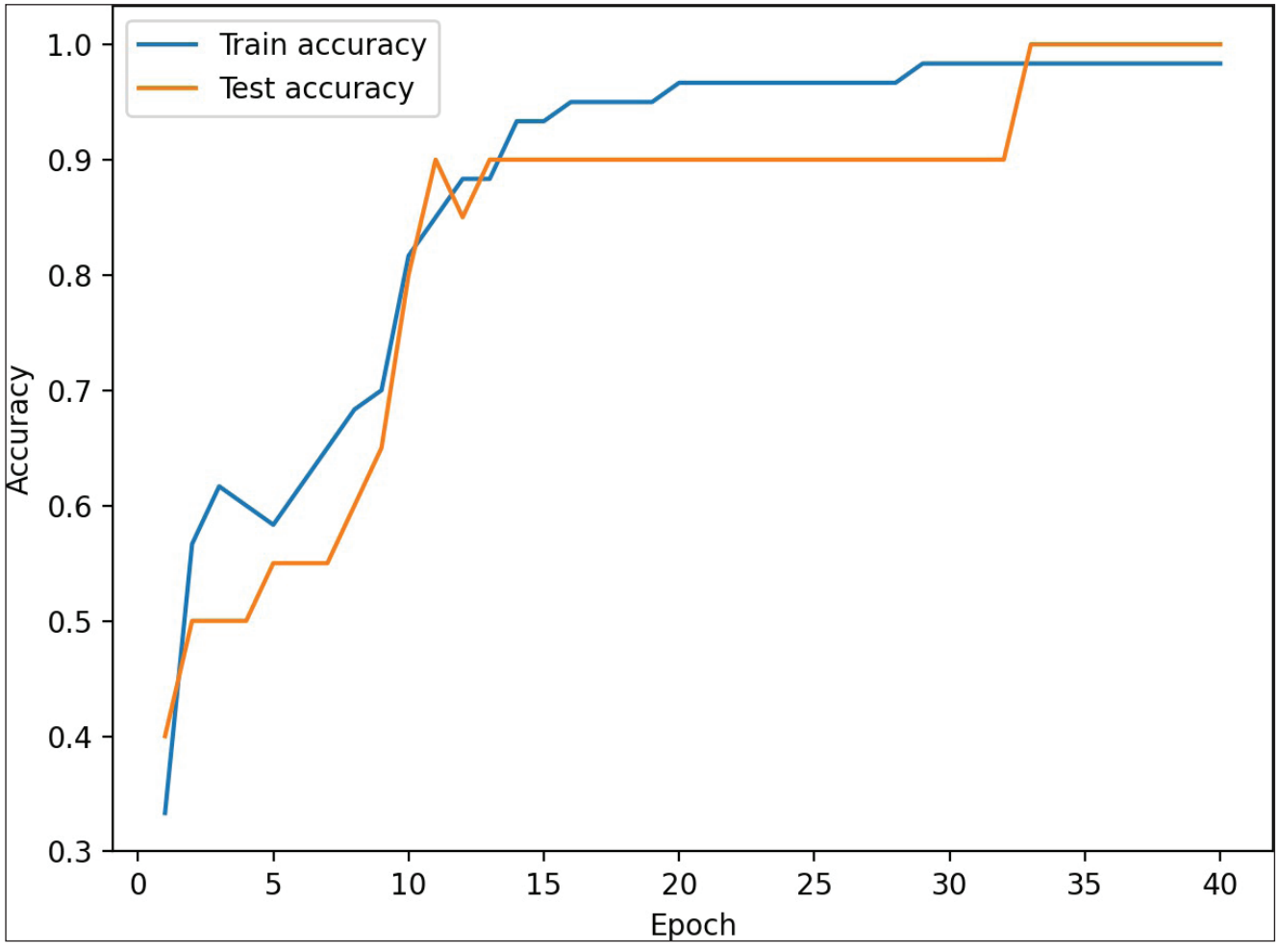


Figure 8. VQC train/test accuracy over epochs (toy binary anomaly-detection dataset).

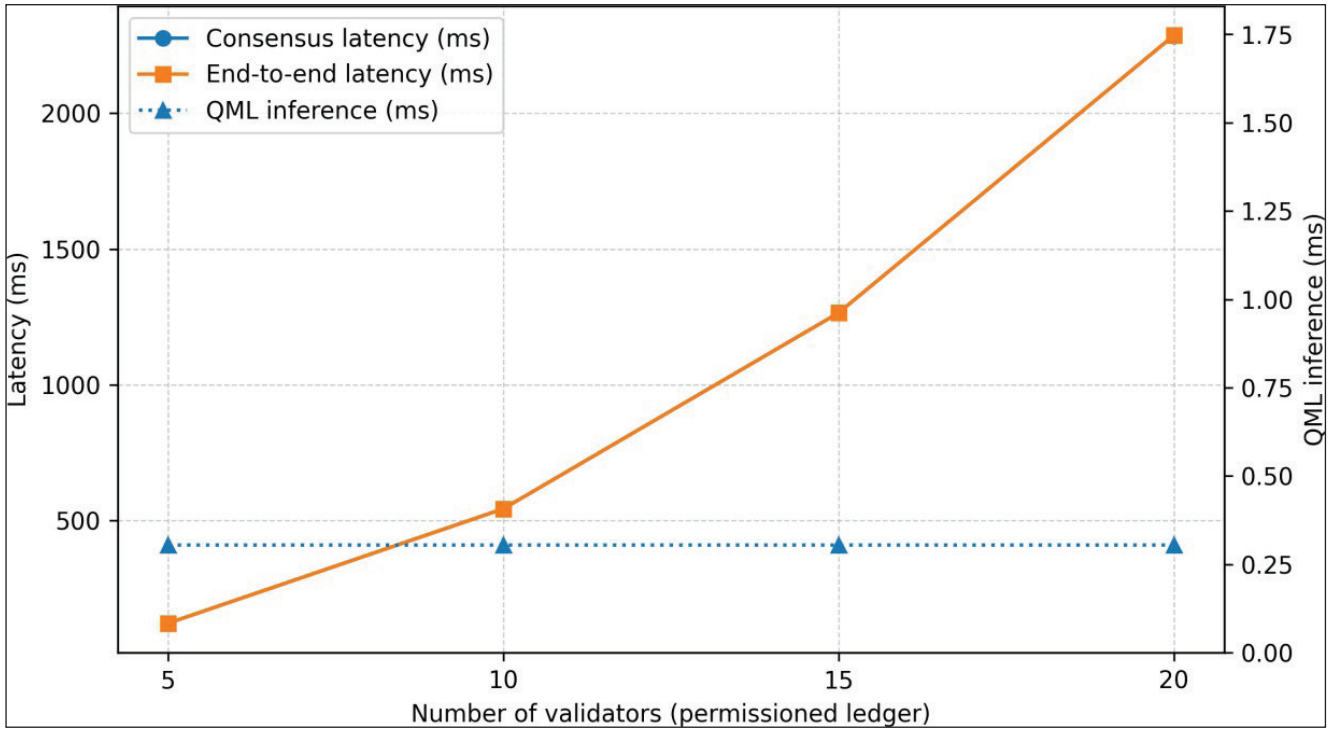


Figure 9. Illustrative consensus, end-to-end, and QML-inference latency versus number of permissioned validator nodes (dual-axis for readability).

Table 4. Public datasets suitable for practical validation of QML–blockchain pipelines in UAV/drone networks.

Dataset	Type/modality	What it contains	How it supports QML–blockchain evaluation	Source
UAV Telemetry Dataset with Waypoint Navigation	Flight telemetry (lat/lon, heading, battery, gimbal, waypoint index)	Waypoint- following the telemetry trace collected during a planned mission.	Integrity + anomaly detection: detect path deviations/spoofing; commit telemetry segment hashes and alert decisions to the ledger.	[26]
ISOT Drone Anomaly Detection Dataset (DJI Edu Tello)	Wi-Fi comms PCAP + extracted CSV features	Benign and anomalous drone communication traffic collected From a real consumer drone.	Intrusion/anomaly detection: train QML classifiers on traffic features; log alerts andmodel provenance on-chain.	[24]
UAV-GCS-IDS	UAV–Ground Control Station network traffic (benign + attacks)	Traffic traces between UAV and GCS under normal and malicious conditions.	Command-channel IDS: evaluate detection under realistic control traffic; quantify ledger write rate under alert bursts.	[27]
UAVIDS-2025	UAV network intrusion detection benchmark (tabular traffic features)	Benchmark dataset designed for UAV network IDS evaluation (DoS and related attacks).	Benchmarking: standardized train/test splits for IDS; compare QML vs classical baselines with shared metrics.	[28]
ALFA: UAV Fault and Anomaly Detection	Real flight data with labeled faults	Fixed-wing flight logs with sudden fault scenarios and ground-truth fault annotations.	Safety monitoring: evaluate detection of physical faults vs cyber anomalies; ledger can store tamper-evident maintenance/fault reports.	[29]

Summary of illustrative results

Discussion: Key Challenges and Research Directions

Practical constraints for deployment

Scalability: Both QML and blockchain have scaling barriers. QML models on near-term devices are limited by qubit count and noise; blockchain consensus scales poorly with the number of validators and with high mobility.

Latency and energy: Autonomous safety loops require tight deadlines. Permissioned ledgers and off-chain storage can reduce overhead, but they must be engineered alongside network protocols and edge compute budgets.

Hardware and resource constraints: Near-term QML for mobile robots is typically limited to small variational circuits. In practice, UAV/UGV deployments will be hybrid: classical perception and compression run on-board, while quantum processing is either (i) executed on a nearby edge server with a QPU (via secure links) or (ii) approximated by quantum-inspired kernels when QPU access is unavailable. Studies should report qubit count, circuit depth, number of shots, and end-to-end wall-clock time, and they should discuss how computation is offloaded without violating control-loop deadlines.

Security and privacy: Blockchains provide integrity and traceability, but they do not automatically guarantee confidentiality. Designs often require encrypted payloads, access control, and careful handling of sensitive mission data.

Blockchain can also strengthen optimization mechanisms in dynamic environments by providing a shared, timestamped state for distributed scheduling and resource allocation. For example, task bids, route commitments, and federated-learning round summaries can be logged as transactions, enabling agents to resolve conflicts consistently even under intermittent connectivity. QML contributes by learning compact decision policies or trust scores from noisy data; blockchain contributes by making those decisions auditable and enforceable.

Threat model detail improves interpretability. A minimal model should specify the attacker's position (external eavesdropper, active jammer, Sybil/impersonator, or compromised insider), targeted assets (telemetry, control commands, model updates, or ledger state), and capabilities (replay, spoofing, poisoning, denial-of-service). Permissioned blockchains mitigate Sybil attacks through admission control, while signatures and authenticated encryption protect command and telemetry payloads. QML modules can complement these defenses by detecting anomalies in communication patterns or control residuals.

Evaluation gaps: Many studies report either learning metrics or ledger metrics, but not both. Future evaluations should report end-to-end latency, communication overhead, energy consumption, decision quality, and resilience to adversarial behaviors (spoofing, poisoning, and Sybil attacks).

Recommendations for future benchmarking

We recommend that future QML–blockchain studies adopt shared benchmarks with (i) realistic mobility and network models for UAV/UGV swarms; (ii) standardized threat models (jamming, spoofing, insider compromise); (iii) explicit ledger configurations (consensus protocol, validator count, block interval); and (iv) reproducible ML/QML baselines on the same data.

When quantum hardware is not available, authors should clearly state whether results come from simulators, noise models, or real devices. Reporting compute budgets (circuit depth, number of shots, wall-clock time) is essential for fair comparison.

Figure 7 summarizes a reproducible end-to-end benchmarking pipeline and the minimum set of metrics that should be reported to compare studies fairly.

Real-world datasets and case studies for practical validation

Reviewer feedback emphasized the value of grounding QML–blockchain claims in real-world evidence. While Section 6 provides a proof-of-concept simulation to make the proposed architecture more concrete, this subsection consolidates publicly available datasets and reported case-study settings that can be used to validate QML–blockchain pipelines under realistic telemetry, network, and threat conditions.

These resources enable (i) repeatable training/evaluation of anomaly and intrusion detection models, (ii) measurement of ledger overhead under realistic event rates, and (iii) comparison across studies using shared benchmarks.

- Use on-board/edge features (telemetry + comms) as QML inputs; store hashes, alerts, and provenance pointers on-chain.

- Report dual metrics: learning performance (e.g., F1/ROC-AUC) and system overhead (latency, bandwidth, energy, gas/fees).
- Document the threat model and attack injection procedure (spoofing, command injection, jamming, replay) to ensure reproducibility.

Case study settings reported in the literature

Industrial UAV deployments frequently require end-to-end auditability for sensor data, flight actions, and operator commands. For example, Mehta et al. discuss blockchain-envisioned UAV networks and describe an industrial-oriented case setting where blockchain is used to secure UAV communications and enforce traceability across stakeholders. Such settings map naturally to the proposed integration framework: blockchain provides non-repudiation and shared governance, while learning modules (including QML candidates) can focus on predictive security and optimization without losing provenance [30].

Together, the datasets in Table 4 and these reported case-study contexts provide a practical path to move beyond purely synthetic evaluation: authors can ground intrusion/anomaly claims in real traces, and quantify blockchain costs under realistic alert/event rates.

Conclusion

This review synthesized 28 verified studies on quantum machine learning (QML), blockchain, and their integration for UAV/UGV autonomy and drone network coordination. The literature indicates a practical division of labor: QML is most often used for perception, anomaly detection, trust scoring, and local decision policies, whereas blockchain is used for authentication, coordination, and auditable mission logging. Because most evaluations remain simulator- based, the field needs standardized benchmarks that jointly report decision quality and distributed-ledger overhead under realistic mobility, radio conditions, and energy budgets. The proposed reference framework and proof-of- concept benchmark provide a concrete starting point for such evaluations. In industrial deployments such as inspection fleets, logistics, search-and-rescue, and critical-infrastructure monitoring, integrated QML and blockchain pipelines could enable faster anomaly response with stronger accountability across operators. Future work should prioritize hybrid deployments (edge plus MEC/QPU access), explicit threat models, and hardware-in-the-loop or field tests to validate end-to-end reliability.

Conflicts of interest

The authors declare no conflicts of interest.

Funding

The authors declare no external funding for this work.

Data availability

No new real-world datasets were created. The proof-of-concept simulation uses synthetic data generated as described in Section 6. However, Section 7.3 lists public datasets that can be used to reproduce and extend evaluations on real telemetry and network traces.

Author contributions

Conceptualization and writing: all authors. Methodology and simulation: all authors.

References

1. Biamonte J, Wittek P, Pancotti N, Rebentrost P, Wiebe N, Lloyd S. Quantum machine learning. Nature 2017; 549:195–202; doi:10.1038/nature23474
2. Nakamoto S. Bitcoin: A Peer-to-Peer Electronic Cash System, 2008. Available from: https://bitcoin.org/bitcoin.pdf
3. Dunjko V, Briegel HJ. Machine learning & artificial intelligence in the quantum domain: a review of recent progress. Rep Prog Phys 2018; 81:074001; doi:10.1088/1361-6633/aab406
4. Schuld M, Sinayskiy I, Petruccione F. The quest for a quantum neural network. Quantum Inf Process 2014; 13:2567–86; doi:10.1007/s11128-014-0809-8
5. Havlíček V, Córcoles AD, Temme K, Harrow AW, Kandala A, Chow JM, et al. Supervised learning with quantum-enhanced feature spaces. Nature 2019; 567:209–12; doi:10.1038/s41586-019-0980-2
6. Cerezo M, Arrasmith A, Babbush R, Benjamin SC, Endo S, Fujii K, et al. Variational quantum algorithms. Nat Rev Phys 2021; 3:625–44; doi:10.1038/s42254-021-00348-9
7. Rodríguez-Díaz F, Gutiérrez-Avilés D, Troncoso A, Martínez-Álvarez F. A Survey of Quantum Machine Learning: foundations, Algorithms, Frameworks, Data and Applications. ACM Comput Surveys 2025; 58(4):1–35; doi:10.1145/3764582
8. Alladi T, Chamola V, Sahu N, Guizani M. Applications of blockchain in unmanned aerial vehicles: a review. Veh Commun 2020; 23:100249; doi:10.1016/j.vehcom.2020.100249

9. Harbi Y, Medani K, Gherbi C, Senouci O, Aliouat Z, Harous S. A systematic literature review of blockchain technology for Internet of Drones security. Arabian J For Sci Eng 2023; 48(2):1053–74; doi:10.1007/s13369-022-07380-6
10. Hafeez S, Khan AR, Al-Quraan MM, Mohjazi L, Zoha A, Imran MA, et al. Blockchain-Assisted UAV Communication Systems: a comprehensive survey. IEEE Open J Veh Technol 2023; 4:558–80; doi:10.1109/OJVT.2023.3295208
11. W, Yu H, Wang J, Wang J, Zhang. LAKA-UAV: lightweight authentication and key agreement scheme for UAV communication. Comput Netw 2023; 226:2023; doi:10.1016/j.comnet.2023.109612
12. Hu N, Tian Z, Sun Y, Yin L, Zhao B, Du X, et al. Building Agile and Resilient UAV Networks Based on SDN and Blockchain. IEEE Netw 2021; 35(1):57–63; doi:10.1109/MNET.011.2000176
13. Tan Y, Liu J, Kato N. Blockchain-Based Lightweight Authentication for Resilient UAV Communications: architecture, Scheme, and Future Directions. IEEE Wireless Commun 2022; 29(3):24–31; doi:10.1109/MWC.002.2100594
14. Singh M, Aujla GS, Bali RS. A deep learning-based blockchain mechanism for secure Internet of Drones environment. IEEE Trans Intell Transp Syst 2021; 22(7):4404–13; doi:10.1109/TITS.2020.2997469
15. Christidis K, Devetsikiotis M. Blockchains and Smart Contracts for the Internet of Things. IEEE Access 2016; 4:2292–303; doi:10.1109/ACCESS.2016.2566339
16. Page MJ, McKenzie JE, Bossuyt PM, Boutron I, Hoffmann TC, Mulrow CD, et al. The PRISMA 2020 statement: an updated guideline for reporting systematic reviews. BMJ 2021; 372:n71; doi:10.1136/bmj.n71
17. Farhi E, Goldstone J, Gutmann S. A quantum approximate optimization algorithm. arXiv:1411.4028, 2014. https://arxiv.org/abs/1411.4028.
18. Hassija V, Chamola V, Krishna DNG, Guizani M. A Distributed Framework for Energy Trading Between UAVs and Charging Stations for Critical Applications. IEEE Trans Veh Technol 2020; 69(5):5391–402; doi:10.1109/TVT.2020.2977036
19. Xiao W, Li M, Alzahrani B, Alotaibi R, Barnawi A, Ai Q. A blockchain-based secure crowd monitoring system using UAV swarm. IEEE Netw 2021; 35(1):108–15; doi:10.1109/MNET.011.2000210
20. Qu Y, Uddin MP, Gan C, Xiang Y, Gao L, Yearwood J. Blockchain-enabled federated learning: a survey. ACM Comput Surveys 2022; 55(4):1–35; doi:10.1145/3524104
21. Zhang X, Zhang J, Gao J, Xia L, Guan Z, Hu H, et al. A sharding blockchain-based UAV system for search and rescue missions. Front Comput Sci 2025; 19(3):193805; doi:10.1007/s11704-024-3467-8
22. Zhu C, Zhu X, Ren J, Qin T. Blockchain-Enabled Federated Learning for UAV Edge Computing Network: issues and Solutions. IEEE Access 2022; 10:56591–610; doi:10.1109/ACCESS.2022.3174865
23. Saraswat D, Verma A, Bhattacharya P, Tanwar S, Sharma G, Bokoro PN, et al. Blockchain-Based Federated Learning in UAVs Beyond 5G Networks: a Solution Taxonomy and Future Directions. IEEE Access 2022; 10:33154–82; doi:10.1109/ACCESS.2022.3161132
24. ISOT Research Group (University of Victoria), "ISOT Drone Anomaly Detection Dataset (DJI Edu Tello)," 2024. Available from: onlineacademiccommunity.uvic.ca/isot/2024/12/05/drone-datasets/
25. Gupta L, Jain R, Vaszkun G. Survey of important issues in UAV communication networks. IEEE Commun Surveys & Tuts 2016; 18(2):1123–52; doi:10.1109/COMST.2015.2495297
26. Díez Tomillo J, Sáez Pérez J, Salva-Garcia P. UAV telemetry dataset with waypoint navigation. Zenodo 2025; doi: 10.5281/zenodo.15912415
27. "UAV-GCS Intrusion Detection Dataset (UAV-GCS-IDS)," Figshare, Jul. 21, 2025. Available: figshare.com/articles/dataset/UAV-GCS_Intrusion_Detection_Dataset_UAV-GCS-IDS_/29608541
28. Zeng A, Bashir F, Naït-Abdesselam Q. UAVIDS-2025: a Benchmark Dataset for Intrusion Detection in UAV Networks Using Machine Learning Techniques. In Proc. IEEE CNS, 2025, p. 2025. doi:10.1109/CNS66487.2025.11194990
29. Keipour A, Mousaei M, Scherer S. ALFA: a dataset for UAV fault and anomaly detection. KiltHub, Carnegie Mellon University, Pittsburgh, PA, 2020; doi:10.1184/R1/12707963
30. Mehta P, Gupta R, Tanwar S. Blockchain envisioned UAV networks: challenges, solutions, and comparisons. Comput Commun 2020; 151:518–38; doi:10.1016/j.comcom.2020.01.023