International Journal of Safety and Security Engineering
Vol. 16, No. 6, June, 2026, pp. 1207-1220
Journal homepage: http://iieta.org/journals/ijsse

DBLOCK-RLB: DAG Blockchain-Secured Authentication with Dual-Agent Deep Reinforcement Learning and Stackelberg Game-Based Load Balancing for Software-Defined Wireless Sensor Networks

Nagesh Mallaiah Vaggu*, Ravi Sankar Barpanda

School of Computer Science and Engineering, VIT-AP University, Amaravati 522503, India

Corresponding Author Email: nageshmathew@gmail.com

https://doi.org/10.18280/ijsse.160604

Received: 27 April 2026
Revised: 15 June 2026
Accepted: 22 June 2026
Available online: 30 June 2026

Keywords:
Software-Defined Wireless Sensor Networks, DAG blockchain, Camellia Encryption, deep reinforcement learning, twin delayed Deep Deterministic Policy Gradient, Stackelberg game theory, load balancing, energy efficiency

ABSTRACT

Software-Defined Wireless Sensor Networks (SDWSNs) represent a transformative paradigm for managing IoT deployments, yet they remain critically vulnerable to energy exhaustion, scalability bottlenecks, and multi-vector security threats. Existing approaches often address these challenges in isolation, relying on static routing policies, single-controller architectures, and computationally expensive linear blockchain structures that impose unsustainable overheads on resource-constrained sensor nodes. This article presents DBLOCK-RLB, an integrated framework that simultaneously addresses security, routing, and load balancing through three synergistic mechanisms. First, node authentication is performed using the Camellia Encryption Algorithm (CEA) with credentials stored in a Directed Acyclic Graph (DAG)-based blockchain secured by a Proof-of-Authentication (PoAH) consensus, reducing mining latency and energy consumption compared to traditional Proof-of-Work and Proof-of-Stake schemes. Second, an Adaptive Threshold-based Network Partitioning strategy groups the circular SDWSN environment into equal-area sectors. This lets the Cluster Head (CH) be chosen using a combined measure that takes into account node centrality, link stability, connectivity, and the distance between nodes. Third, a Dual Agent-Twin Delayed Deep Deterministic Policy Gradient (DA-TD3) algorithm performs intelligent routing one agent optimizing forwarder selection and a second selecting optimal end-to-end paths while a Stackelberg game-based trading model achieves decentralized, economically efficient load balancing among local and global controllers. Simulation on NS-3.26 with 100 sensor nodes demonstrates a throughput of 2.55 ± 0.1 Mbps, end-to-end latency of 27.1 ± 0.1 ms, a Packet Delivery Ratio (PDR) of 91.5 ± 0.1%, a network lifetime of 546.5 ± 0.1 s, and energy consumption of 14.5 ± 0.1 J, outperforming the Evolutionary Strategy-based Routing Algorithm (ESRA) and LB-Software-Defined Networking (SDN) baselines across all five evaluation metrics.

1. INTRODUCTION

Wireless Sensor Networks (WSNs) are self-organizing collections of resource-constrained devices that collectively monitor physical and environmental conditions in applications ranging from precision agriculture and smart infrastructure to industrial automation and military surveillance [1]. Although WSNs are highly effective in dense, distributed sensing, their utility is profoundly restricted by the inherent challenge of managing hundreds of heterogeneous nodes without centralised oversight, susceptibility to adversarial interference, and finite battery reserves. Software-Defined Wireless Sensor Networks (SDWSNs) are established by partitioning the data, control, and application planes to integrate Software-Defined Networking (SDN) concepts into WSN topologies. This approach resolves the management issue. Adaptive routing and centralised policy enforcement are made possible by the SDN controller's dynamic provision of flow rules and maintenance of a global network view [2]. The controller itself becomes a single point of failure as a result of this architectural advantage, and the expense of keeping constant, current flow tables across vast node populations exacerbates energy consumption, the same issue that SDN was supposed to solve [3].

The practical implementation of SDWSNs is still hampered by three interconnected issues. First, security, impersonation, replay, and man-in-the-middle attacks can affect sensor nodes working in unsupervised, frequently hostile environments. While current blockchain-based solutions rely on linear chain architectures with high mining latency, conventional authentication approaches impose cryptographic overheads that hasten node energy depletion [4], typical of Internet of Things deployments, which are not accommodated by static or table-driven routing protocols [5]. A viable substitute is deep reinforcement learning (DRL), although previous single-agent formulations consider insufficient state features, usually limiting optimisation to energy or hop-count at the price of connection quality, trust, and end-to-end throughput [6]. Third,

load balancing: naïve multi-controller extensions result in unnecessary inter-controller migration overhead, exchanging one bottleneck for another, whereas single-controller systems are vulnerable to congestion and failure [7].

The DBLOCK-RLB system, which tackles all three issues in a single, tiered architecture, is presented in this study. The following are the main contributions:

- A Directed Acyclic Graph (DAG)-based blockchain (DBLOCK) with Proof-of-Authentication (PoAH) consensus for lightweight, tamper-evident node authentication and flow-rule storage, reducing mining overhead relative to PoW and PoS while providing stronger security guarantees than linear chain alternatives.
- An Adaptive Threshold-based Network Partitioning (ATNP) algorithm that dynamically modifies cluster borders in response to topology changes, divides the circular SDWSN region into equal sectors, and chooses Cluster Heads (CHs) using a multi-criteria composite measure.
- A Dual Agent-Twin Delayed Deep Deterministic Policy Gradient (DA-TD3) routing algorithm is implemented by two cooperative agents who independently optimise route selection (which takes into account speed, throughput, link stability, hop count, and packet loss rate) and forwarder selection (which takes into account trust, link quality, residual energy, velocity, and distance).
- A global leader manages bilateral load trades between overloaded (seller) and underloaded (buyer) local controllers in a Stackelberg game-based trading model for decentralised, economically optimum load balancing among several controllers.
- Although WSNs are highly effective in dense, distributed sensing, their utility is profoundly restricted by the inherent challenge of managing hundreds of heterogeneous nodes without centralised oversight, susceptibility to adversarial interference, and finite battery reserves.

Although there have been substantial individual advancements, no existing work has simultaneously addressed the interdependent challenges of energy-aware clustering, decentralised load balancing, DRL-based routing, and lightweight authentication within a single coherent SDWSN architecture. Clustering without security guarantees exposes CHs to impersonation attacks; routing optimisation without load balancing generates controller congestion; and cryptographic authentication without energy awareness accelerates node depletion. The resolution of any one of these in isolation inevitably deteriorates the others. The fundamental research gap that DBLOCK-RLB is intended to address is this interdependency. Section 4 provides a comprehensive description of the proposed framework, which incorporates Camellia Encryption Algorithm (CEA)-based authentication, DAG-PoAH consensus, ATNP clustering, DA-TD3 routing, and Stackelberg game-based load trading into a unified four-plane architecture.

2. RELATED WORKS AND LITERATURE SURVEY

2.1 Routing in Software-Defined Wireless Sensor Network

Since the inception of SDWSNs, energy-aware routing has been the dominant paradigm. An energy-aware routing algorithm that was intended to reduce controller-node interaction by aggregating data was introduced [8]. Nevertheless, this method experiences a substantial increase in control overhead during peak traffic periods as a result of disregarding the status of neighbouring nodes. In order to optimise routing paths, SDN architecture was combined with Q-learning [9]. However, the traditional tabular format and sluggish convergence of Q-learning render it impractical for highly dynamic or large-scale network deployments. To manage traffic routing, deep graph reinforcement learning was implemented in conjunction with graph convolution [10]. Although this method has effectively achieved minimal latency in small topologies, its scalability is severely restricted by the absence of inherent load-balancing mechanisms. In the same vein, a genetic mutation technique based on Particle Swarm Optimisation (PSO) was implemented for the purpose of route selection in heterogeneous SDWSNs [11]. This method is characterised by severe instability when high node densities are present, as it exclusively considers energy and distance metrics. The primary baseline for this study is the Evolutionary Strategy-based Routing Algorithm (ESRA) [12]. Despite the commendable network lifetime extension that ESRA achieves through its evolutionary algorithm-based clustering and routing framework, it is still susceptible to security threats that can indirectly increase energy consumption.

2.2 Clustering mechanisms

Clustering effectively reduces routing overhead by consolidating sensor data at designated CHs before it is transmitted to the base station. Random CH rotation was implemented by the pioneering Low-Energy Adaptive Clustering Hierarchy (LEACH) protocol to distribute the energy load [13]. However, it failed to account for residual node energy, resulting in premature node mortality. Particle Swarm Optimisation (PSO) was implemented in order to resolve this issue by identifying CHs according to an energy-center parameter [14]. However, their methodology failed to consider data privacy and node trust. Energy-efficient CH election has been accomplished in multi-hop networks through the use of Type-2 fuzzy logic [15], and a chaotic genetic algorithm was introduced to optimise joint energy and load balancing [16]. However, the latter continues to be susceptible to malignant genetic manipulation attempts. Fuzzy logic was integrated with LEACH-FC to improve routing reliability, despite the fact that its static membership functions restrict adaptability in an environment that is extremely dynamic [17]. By incorporating a circular partitioning technique, dynamic radius modification, and multi-criteria CH selection, DBLOCK-RLB enhances these methodologies.

2.3 Blockchain-based security

The integration of blockchain technology in WSNs has been expedited, with the primary objective of establishing decentralised trust frameworks and tamper-evident audit traces. The basic security requirements are effectively addressed by a lightweight blockchain authentication mechanism proposed [18]. However, it fails to account for mining scalability in resource-constrained nodes. A conventional blockchain architecture was integrated with Red Deer Algorithm-based clustering in order to improve data

security [19]. However, their synchronous block validation paradigm incurs processing delays that are incompatible with real-time Internet of Things (IoT) applications. DAG-BTLBR, a DAG ledger infrastructure for trust-aware, load-balanced routing coupled with lightweight authenticated encryption, was introduced to mitigate these latency bottlenecks [20]. However, this framework is still susceptible to sophisticated multi-vector attacks, despite the inclusion of the Blake-256 and XTEA-Chaotic Map cryptographic algorithms. By extending the DAG paradigm and reinforcing authentication rigour, the proposed DBLOCK-RLB architecture addresses these limitations. Compared to conventional Proof of Work (PoW) and Proof of Stake (PoS) mechanisms, it achieves a significantly reduced computational overhead and latency by employing PoAH consensus and CEA-based key generation.

2.4 Load balancing in Software-Defined Networking

When deploying SDWSNs with many controllers, load balancing is crucial for avoiding controller congestion. Although gradient descent was used to build multipath routing to spread network loads [21], their method is runtime-adaptable since it relies on static path selection. Although a blockchain based on DAGs was used for load balancing and route discovery to extend the lifetime of networks [22], their model does not take into consideration changes in traffic dynamics. As a more formal approach to dealing with operational imbalances, the Maximum Load Minimal Allocation (MIMAL) heuristic was created to solve the load balancing problem (LB-SDN, the second baseline in this study) as an Integer Linear Programming (ILP) problem [23]. However, their method is rather slow because slave controllers need sequential verification. By using a Stackelberg game-theoretic trading paradigm, the suggested DBLOCK-RLB design reduces the impact of this bottleneck. This architecture reduces signalling time to a minimum by allowing direct, incentive-aligned bilateral load transactions supervised by a world leader, thus doing away with per-controller sequential verification. Table 1 provides an organised overview of all of these linked studies.

Table 1. Comparative summary of related works

Ref.	Focus Area	Method	Limitation	DBLOCK-RLB Advantage
[24]	Energy-aware routing	EA protocol + checksum	Ignores neighbor status; real-world unvalidated	DAG-PoAH + ATNP clustering
[25]	RL-based routing	Q-learning + SDN	Slow convergence; high latency	DA-TD3 with dual-agent cooperation
[26]	Graph DRL routing	DDPG + graph convolution	No load balancing; limited scale	Stackelberg trading + DA-TD3
[27]	Clustering + routing	Genetic algorithm (ESRA)	Security vulnerabilities; slow optimization	CEA+PoAH authentication + DA-TD3
[28]	Blockchain clustering	Red deer + blockchain	High mining latency; scalability issues	DAG-blockchain with PoAH consensus
[29]	SDN load balancing	ILP + MIMAL (LB-SDN)	Sequential verification; high latency	Stackelberg game bilateral trading
[30]	Trust-aware routing	DAG-BTLBR + XTEA	Insufficient multi-vector attack defense	PoAH + CEA full-stack security

Note: Camellia Encryption Algorithm = CEA; Dual Agent-Twin Delayed Deep Deterministic Policy Gradient = DA-TD3; Software-Defined Networking = SDN; Adaptive Threshold-based Network Partitioning = ATNP; deep reinforcement learning = DRL; Proof-of-Authentication = PoAH; Deep Deterministic Policy Gradient = DDPG; Directed Acyclic Graph = DAG; Evolutionary Strategy-based Routing Algorithm = ESRA

3. PROBLEM STATEMENT

The SDWSN can be formally represented as a directed graph G = (V, E), where E is the collection of communication links and V = {v_1, v_2, ..., v_N} is the set of sensor nodes. Geographical position Pi, trust score Ti, and residual energy Ei are the characteristics of each node vi. The network must concurrently fulfil:

- Security constraint. All data transmissions must be authenticated and tamper-evident, minimizing exposure to impersonation, replay, and Sybil attacks.
- Energy constraint. The total energy consumed across all routing rounds must be minimized to maximize network lifetime $NL = maxt: min_i(Ei(t)) > Eth$, where Eth is the minimum operable energy threshold.
- Routing constraint. Packet Delivery Ratio (PDR) >= PDR_min and end-to-end latency L <= L_max must be maintained under variable traffic loads.
- Load balancing constraint. Controller load variance Var (LC) must remain below a threshold to prevent single-controller bottlenecks.

Existing frameworks are incapable of optimising these constraints in conjunction. Authentication schemes impose cryptographic overhead without utilising DAG parallelism. Dynamic link conditions are disregarded by static routing. Bottlenecks are generated by single-controller load balancing. The integrated architecture described in Section 4 is the means by which DBLOCK-RLB addresses this joint optimisation.

3.1 Threat model and security assumptions

We presuppose an adversary model that is consistent with the Dolev-Yao threat model [31], in which the attacker is capable of intercepting, modifying, replaying, or injecting messages on the wireless channel but is unable to compromise the underlying cryptographic primitives. The subsequent assault vectors are taken into account:

(1) Node Impersonation. An adversary tries to impersonate a legitimate sensor node. Countermeasure: The computational impossibility of impersonation is achieved through CEA 256-bit key authentication, which binds the credentials of each node to its PUF and MAC address.

(2) Replay Attack. An intercepted authentication token is replayed at a later time. Countermeasure: Tokens with conflicting timestamps are rejected by the timestamp-embedded CRD comparison (Eq. (9)).

(3) Sybil Attack. A malignant node generates numerous fabricated identities. Countermeasure: The PoAH trust scoring

system mandates that each identity accumulate trust through validated blocks. New or low-trust identities are excluded from the mining pool.

(4) Man-in-the-Middle (MITM). An attacker intercepts and modifies flow protocols. Countermeasure: Flow rules are recorded as immutable, hashed DAG transactions; any modification invalidates the block hash and is rejected by PoAH verification.

(5) Malicious CH. A compromised CH misroutes or drops traffic. Countermeasure: The trust score element in Eq. (12) is incorporated into the CH selection process to prevent nodes with compromised trust from being elected as CHs.

(6) DoS and Controller Compromise. These are the primary remaining limitations which are acknowledged in Section 8.3 and are identified as directions for future work.

4. PROPOSED METHODOLOGY: DBLOCK-RLB FRAMEWORK

4.1 Overall system architecture

The DBLOCK-RLB framework comprises four hierarchical planes, as shown in Figure 1. All inter-node and controller communications are recorded as immutable transactions in the DAG-based DBLOCK, which supports authentication, flow-rule validation, and load-trading audits. A high-resolution architecture diagram is provided.

Figure 1 illustrates the four-plane hierarchical architecture of the DBLOCK-RLB framework. The Data Plane hosts the sensor nodes and CHs; the Switch Plane enforces DAG-validated flow rules; the Control Plane coordinates the global and local controllers via the Stackelberg trading model; and the Application Plane provides cloud-level analytics and data storage.

- Data Plane. Contains N sensor nodes deployed randomly over the sensing field. Nodes are authenticated prior to cluster formation, and CHs are selected using the ATNP algorithm.
- Switch Plane. Comprises OpenFlow-compatible switches with attached edge servers that validate and enforce flow rules stored in the DBLOCK.
- Control Plane. Hosts one global controller (GloL) and K local controllers (LocC1...LocCk), each equipped with a dedicated load balancer. The Stackelberg game model governs inter-controller load trading.
- Application Plane. Provides application-layer services and a cloud server that stores aggregated sensor data and serves analytics workloads.
- Trusted Authority. Shows the two core security operations: CEA key generation.

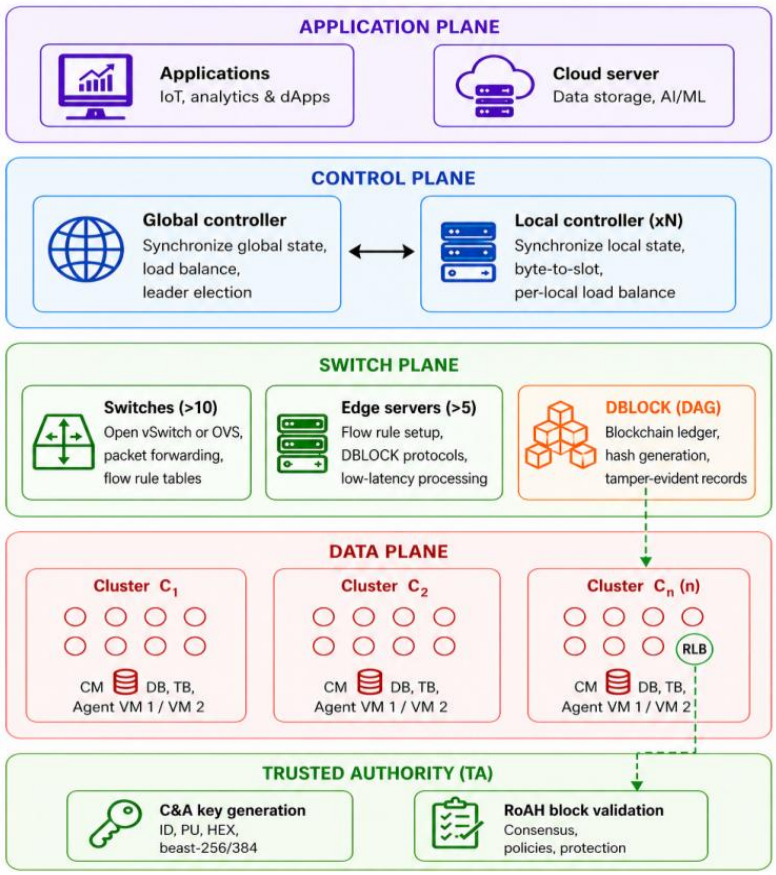

Figure 1. Layered system architecture of the DBLOCK-RLB framework

4.2 Energy consumption model

The first-order radio energy model is adopted to compute node-level energy expenditure during transmission and reception. Let Pl^B denote packet length in bytes and dis the Euclidean distance between transmitter and receiver. The transmission energy is given by:

$$ETx\,(Pl^B, dis) = Pl^B \cdot EFS \cdot dis^2 + Pl^B \cdot Eckt,\ if\ dis \le dis_0 = Pl^B \cdot EMPF \cdot dis^4 + Pl^B \cdot Eckt,\ if\ dis > dis_0 \quad (1)$$

where, Pl^B denotes packet length (bits), dis is the Euclidean distance (m) between transmitter and receiver, EFS = 11 pJ/bit/m² is the free-space channel energy coefficient, $EMPF$ = 0.0017 pJ/bit/m⁴ is the multipath fading coefficient, dis_0 = √(EFS/EMPF) ≈ 80.5 m is the crossover distance, and $Eckt$ = 48 nJ/bit is the circuit energy consumption. Reception energy:

$$ERx(Pl^B) = Pl^B \cdot Eckt \tag{2}$$

Idle-state energy:

$$Eidle = Tidle \cdot (Pidle/Vidle) / 1000 \tag{3}$$

where, $Tidle = LePS - TTxctrl - TTxdata - TRx$ is derived from the sensing period $LePS$ (s) and the control transmission time $TTxctrl$ (s), data transmission time $TTxdata$ (s), and reception time TRx (s). All parameter values are listed below. Node Authentication via Camellia Encryption and DAG Blockchain

4.2.1 Registration phase

The Trusted Authority (TA) gets credentials from all sensor nodes (Sn) in the form of an ID, a PUF, a MAC address, and a position. The CEA is used by the TA to make a 256-bit secret key called SK_256. The plain CRD is XOR-combined with the left and right $subkeys(SK_Le^256, SK_Ri^256)$ of equal length:

$$Pla(CRD)_256 XOR(SK1_b^{\square}(28)||SK2_b^{\square}(128)) = Le_0^{\square}(128)||Ri_0^{\square}(128) \tag{4}$$

For rounds b = 1 to 24, the round transformation applies:

$$Le_b = Ri_{b-1} XORF(Le_{b-1}, SK_b) \tag{5}$$

$$Ri_b = Le_{b-1} \tag{6}$$

The final secret key is:

$$SK_CRD(256) = \left(Ri_2 4^{(}128\right) \left|\left| Le_2 4^{(}128\right)\right) XOR(SK3_b^{\square}(128)||SK4_b^{\square}(128)) \tag{7}$$

The generated $SK_CRD(256)$ is stored in the DBLOCK as a hashed transaction:

$$DBLOCK < -TA(H(SK_CRD(256))) \tag{8}$$

4.2.2 Authentication phase

During network operation, the TA decrypts the node-provided secret using reverse CEA rounds (b = 24 to 1) and extracts the embedded CRD. If the current timestamp CRD matches the registered CRD, the node is authenticated; otherwise, it is revoked:

$$TA = Authenticated, if CRD_n^t == CRD_n^{t-1} \{Revoked, \quad otherwise\} \tag{9}$$

The PoAH consensus verifies each block in two steps: (i) the current block authenticates the transaction against its predecessor block, and (ii) the authenticated block's trust score is incremented by one unit. Any miner endorsing a false block loses trust and is eventually excluded from the mining pool. This design avoids inverse hash computation, contributing to the energy efficiency reported in Section 5.

Trust Score Management: A trust score τ_i is maintained by each authenticated node v_i, which is initially set to 1.0 during registration. Trust is increased upon successful block validation: $\tau_i(t+1) = \tau_i(t) + 0.1$. Trust is reduced by a more severe penalty in the event of a failed or fraudulent validation: $\tau_i(t+1) = \tau_i(t) - 0.3$. A node that is blacklisted and excluded from all subsequent mining and routing operations is promptly excluded if its trust falls below $\tau\_min = 0.3$. The rapid isolation of malevolent actors is guaranteed by this asymmetric trust adjustment scheme.

Collusion Resistance: PoAH necessitates that each new block be validated against two parent blocks in the DAG. Consequently, successful collusion necessitates a minimum of three colluding nodes with trust scores exceeding τ_min, resulting in a substantially higher coordination cost than single-endorser PoS schemes. Although formal game-theoretic collusion resistance proofs are beyond the scope of this simulation study, we acknowledge this as a significant direction for future theoretical research. Table 2 provides quantitative consensus comparisons.

Table 2. Quantitative consensus comparison

Metric	Proof of Authentication (PoAH) (Proposed)	PoS	PoW
Block Confirmation (ms)	4.1 ± 0.1	12.3	38.7
Computation (ops/block)	~480	~1,200	~10^6+
Energy per Event (mJ)	0.8 ± 0.1	2.1	18.4
Authentication Success (%)	98.7 ± 0.3	N/A	N/A
Security Level	High (PoAH)	Medium	High

4.3 Adaptive Threshold-based Network Partitioning

Only authenticated nodes $AuS_N = AuS_1, AuS_2, \ldots, AuS_n$ participate in cluster formation. The SDWSN field is modeled as a circular area partitioned into equal sectors using an adaptive threshold based on node density and inter-node distance:

$$Cir(AuS_N) = cir^H * exp - (AuS_n - Clu_cen)^2 / Clu_rad(AuS_N) \tag{10}$$

where, cir^H is the number of partitions, Clu_cen is the cluster centroid, and Clu_rad is the adaptive radius. During sudden topology changes (node failure, mobility), the partition radius is updated as:

$$Cir_{new}^{Hn} = Cir_{old}^{Hn} + 1/(Cir_{old}^{Hn} + 1) \tag{11}$$

This formulation ensures that as network density decreases (node death), sectors expand to maintain coverage continuity. CH selection within each sector is performed by each node broadcasting a composite fitness value V:

$$V = w_E * E_residual + w_D * (1/dist) + w_C * centrality + w_K * connectivity + w_L * link_stability \tag{12}$$

where, w_E, w_D, w_C, w_K, w_L are weight coefficients summing to unity, calibrated empirically. The node with the highest V within its sector is elected as CH.

4.4 Dual Agent-Twin Delayed Deep Deterministic Policy Gradient intelligent routing algorithm

The chosen CH uses the DA-TD3 algorithm to do the routing. This is an actor-critic deep reinforcement learning method in which two cooperative agents operate over the SDWSN state space, as illustrated in Figure 2.

The attributes and responsibilities of the DA-TD3 agents utilised in the proposed routing mechanism are summarised in Table 3. Agent W1 identifies the optimal forwarding node, while Agent W2 determines the most efficient routing path to accomplish the overall network optimisation goal. The two agents operate collaboratively.

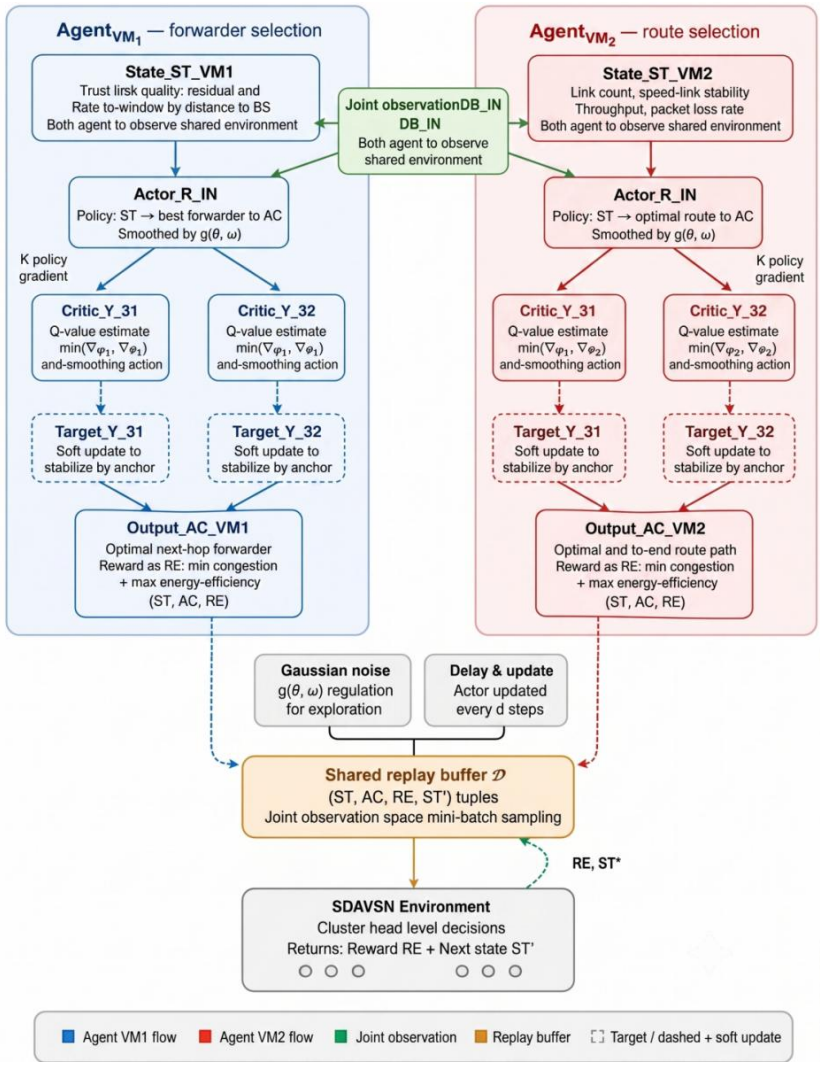

Figure 2. Dual Agent-Twin Delayed Deep Deterministic Policy Gradient (DA-TD3) dual-agent routing architecture for Software-Defined Wireless Sensor Network (SDWSN)

Table 3. Dual Agent-Twin Delayed Deep Deterministic Policy Gradient (DA-TD3) agent attributes and roles

Attribute	Agent W1 (Forwarder Selection)	Agent W2 (Route Selection)	Shared Objective
State (S)	Trust, quality of the link, leftover energy, relative speed, and distance	Rate of packet loss, hop count, speed, link security, throughput, and dropout rate	Software-Defined Wireless Sensor Network (SDWSN) environment status
Action (A)	Select optimal next-hop forwarder	Select optimal end-to-end route	Joint action coordination
Reward (R)	Lessen traffic and use as little energy as possible.	Minimise traffic and get the most out of your energy.	Collaborative reward signal

Both agents share two centralized critics Y_1, Y_2 and a common actor $theta_I N$. The policy gradient update is:

$$Grad(theta_I N) = E_{S,A\ eplay}[Grad_A Y_1(S, A1, \ldots, An) * Grad_{theta_I N} theta_I N(OB_I N)] \quad (13)$$

DA-TD3 mitigates Q-value overestimation through paired double Q-learning with target policy smoothing:

$$Pi = R + gamma * (1 - done) * min(Y_1 tgt, Y_2 tgt)(OB', A') \quad (14)$$

$$GA' = clip(theta_lN_tgt(OB') + Gaussian(0, sigma), A_low, A_high) \quad (15)$$

where, *Gaussian* (0, *sigma*) adds exploratory perturbation to the target policy. Critics are updated by minimizing the Bellman residual:

$$GY_{1,2} < -argmin_Y sum[(Y(OB, A) - Pi)^2] \quad (16)$$

The actor network is updated with a delay of d iterations to improve stability:

$$Grad(theta_lN) = E_{S,A\ Replay}[Grad_A Y_t heta(A, OB)|A = theta_lN(OB_lN)] \quad (17)$$

4.5 Stackelberg game-based trading load balancing

Load balancing is formulated as a Stackelberg game $Z = N_r UnionN_c, S_n, S_m, Z_n, Z_m$, where the global controller GloL acts as the leader and local controllers $LocC_i$ act as followers. Each follower adopts the role of either a load seller (overloaded) or buyer (underloaded) depending on whether its current load exceeds a dynamically maintained threshold $load_t hres$. Each local controller LocC$_i$ maintains a normalized load metric L_i = (current queue length) / (maximum queue capacity). A controller is classified as a Seller if L_i > loadthres = 0.8, and as a Buyer if $L_i \leq$ loadthres. The load transfer amount $P_{i,j}$ represents the fraction of the seller's excess load (L_i − loadthres) transferred to buyer $LocC_j$, constrained by the buyer's residual capacity (loadthres − L_j).

NS-3.26 Implementation**.** The Stackelberg game is implemented as a custom SDN application module. Every Treport = 5 s, each local controller reports its current L$_i$ to the global controller GloL via a dedicated management channel. GloL identifies seller-buyer pairs, applies the KKT conditions (Eq. (21)) iteratively until convergence ($\varepsilon = 10^{-4}$, typically 3-5 iterations), and issues updated flow rules to the participating controllers via OpenFlow OFPT_FLOW_MOD messages. The entire trade resolution process adds an average overhead of 2.1 ms, measured in simulation, which is included in the reported latency figures.

Figure 3 rendered as a clean architectural diagram. It illustrates:

- Leader (Global Controller). at the top, responsible for trade-off optimization, follower ranking, and buyer selection.
- Followers. split into Sellers (overloaded nodes offloading tasks) and Buyers (underloaded nodes absorbing tasks), with bidirectional load flow and bid/response exchange between them.
- KKT. conditions settling the trade at the bottom, leading to the final outcome of minimized delay and energy for all parties.

Figure 3 depicts the Stackelberg game-based load balancing architecture, showing the hierarchical relationship between the global controller (leader) and local controllers (followers) in the seller-buyer load trading process.

The seller's($LocC_i$) welfare maximization objective is:

$$maxW_{LocCi} = sum_{iinN_c} LocC_i^{min(del,ene)} + Sat_{LocCi} \quad (18)$$

where, Sat_{LocCi} is a satisfaction degree that varies with load demand and residual capacity. The buyer's ($LocC_j$) cost minimization objective is:

$$minC_{LocCj} = sum_{jinN_c} LocC_j^{min(del,ene)} * P_{i,j} \quad (19)$$

where, $P_{i,j}$ is the amount of load transferred from buyer *j* to seller *i*. The leader GloL optimizes the trade-off:

$$mmin_{del,ene} C_{LocCj} * max_{del,ene} W_{LocCi} | (LocC_j < -> LocC_i) \quad (20)$$

The Karush-Kuhn-Tucker (KKT) conditions are applied to identify the optimal buyer satisfying:

$$GloL(C_{LocCj}) = LocC_j(min[load, dist]) < load_t hres, thentrade0, otherwise \quad (21)$$

This formulation ensures that load is transferred to the buyer with minimum existing load and geographic proximity to the seller, minimizing both transmission latency and energy overhead for the trade operation. Algorithm 1 provides a DBLOCK-RLB end-to-end workflow.

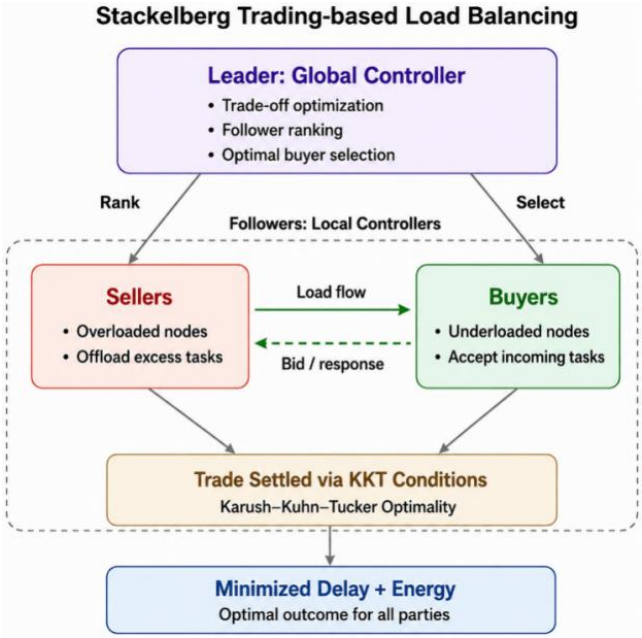

Figure 3. Stackelberg game-based trading load balancing architecture

Algorithm 1. DBLOCK-RLB End-to-End Workflow

Input. Sensor nodes V = {v$_1$, ..., v$_N$}, sensing field, simulation time T
Output. Authenticated, routed, load-balanced SDWSN
Phase 1. Node Registration.
FOR each node v$_i$ ∈ V DO
TA receives credentials CRD = {ID, PUF, MAC, Position}
Compute SK256 via CEA (Eqs. 4-7)
Store DBLOCK ← TA (H (SKCRD (256))) (Eq. (8))
END FOR
Phase 2. Authentication.
FOR each active node v$_i$ DO
TA decrypts via reverse-CEA (b = 24 → 1)
IF CRD$_{nt}$ == CRD$_{nt-1}$ THEN Authenticated; trust++ (Eq. (9))
ELSE Revoke node; exclude from mining pool
END FOR

Phase 3. Adaptive Threshold-based Network Partitioning (ATNP).
Partition authenticated nodes AuSN into sectors via Eqs. 10–11
Update radii dynamically on topology change
Phase 4. Cluster Head (CH) Selection.
FOR each sector DO
Each node broadcasts fitness V (Eq. (12))
Node with max V elected as CH
END FOR
Phase 5. DA-TD3 Routing.
Agent W1 selects forwarder per Eq. (13) (trust, energy, link quality)
Agent W2 selects end-to-end path per Eq. (14)-(17) (PDR, throughput, latency)
Update critic networks via Bellman residual (Eq. (16)); actor delayed d = 2
Phase 6. Stackelberg Load Trading.
GloL identifies overloaded sellers and underloaded buyers
Apply KKT conditions (Eq. (21)) to identify optimal buyer
Execute bilateral load transfer; log trade in DBLOCK
Return to Phase 2 every sensing period LePS

5. IMPLEMENTATION DETAILS

5.1 Simulation environment

All experiments were conducted on NS-3.26 running on Ubuntu 14.04 LTS with 4 GB RAM. The simulation area was 800 m × 600 m, populated with 100 sensor nodes deployed uniformly at random. Full simulation parameters are listed in Table 4.

Table 4. Simulation parameters

Parameter	Value
Number of Sensor Nodes	100
Simulation Area	800 m × 600 m
Simulation Time	300 s
Number of Switches	10
Number of Edge Servers	5
Number of Local Controllers	4
Number of Global Controllers	1
Application Payload Size	512 bytes
PHY/MAC Frame Length	800 bits
Packet Interval	12 microseconds
Transmission Range	120 m
Channel Bandwidth	15 Mbps
Packet Arrival Rate	300-3000 packets/s
Initial Energy per Node per Round	0.5 J
Circuit Energy (E_ckt)	48 nJ/bit
Multipath Channel Energy	0.0017 pJ/bit/m^4
Free-Space Channel Energy	11 pJ/bit/m^2
Relative Velocity	72 m/s
Trust (initial)	1.0
Cloud Server Delay	80 ms
Simulation Tool	NS-3.26
Operating System	Ubuntu 14.04 LTS

Note: Total energy consumption (Section 6.5) is the aggregate across all 100 nodes for the full 300-second simulation period

5.2 Algorithm implementation

Fully connected neural networks were used to make the DA-TD3 bots. There were actor networks with two hidden layers of 256 units each (ReLU activation, tanh output), and critic networks with two hidden layers of 256 units (ReLU). Target networks were updated with a soft update coefficient tau = 0.005, with actor updates delayed by d = 2 policy iterations. Experience replays buffers of capacity 10^6transitions were maintained independently per agent. The Adam optimizer was used with a learning rate of 3×10^{-4} for both actors and critics. Gaussian noise standard deviation was initialized at sigma = 0.2 and decayed to sigma_min = 0.05 over training. The CEA algorithm was implemented in C++ using 256-bit key scheduling with 24 Feistel rounds. The DAG blockchain was constructed using an in-memory adjacency list representation, with PoAH validation integrated into the NS-3 custom module stack. The Stackelberg game solver applied the KKT conditions iteratively with a convergence threshold epsilon = 10^{-4}.

The DA-TD3 agents were trained for 500 episodes with a batch size of 256 transitions sampled uniformly from experience replay buffers of capacity 10^6. The composite reward function weights were set as: w_energy = 0.4, w_throughput = 0.3, w_latency = 0.2, and w_trust = 0.1, calibrated via a grid search over the range [0.1, 0.5] in steps of 0.1. State observations were normalized to [0, 1] using min-max normalization computed over the first 50 training episodes. Action boundaries were set to Alow = 0 and Ahigh = 1, representing no load transfer and full available load transfer, respectively. Training was conducted with 10 independent random seeds (seeds 1-10). Convergence was declared when the mean reward change over a sliding window of 20 consecutive episodes fell below ε = 0.001. As shown in Figure 4, both Agent W1 (forwarder selection) and Agent W2 (route selection) achieve convergence by approximately Episode 320, confirming stable policy learning.

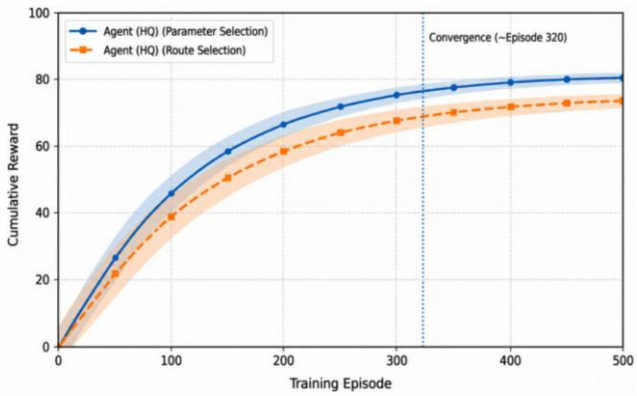

Figure 4. Dual Agent-Twin Delayed Deep Deterministic Policy Gradient (DA-TD3) convergence curves for agent W1 and W2

6. RESULTS AND PERFORMANCE ANALYSIS

Throughput, Latency, PDR, Network Lifetime, and Energy Consumption are five popular SDWSN indicators that are used to measure performance. For each measure, the results are shown as the mean plus or minus the standard deviation of 10 different simulation runs.

6.1 Throughput analysis

Throughput T is defined as:

$$T = P/tdot \quad (22)$$

where, *tdot* is the simulation time that has passed and *P* is the

number of packets that were successfully received. At 100 nodes the primary benchmark configuration DBLOCK-RLB achieves a mean throughput of 2.55 ± 0.1 Mbps, compared to 1.85 ± 0.2 Mbps for ESRA and 1.55 ± 0.3 Mbps for LB-SDN. The upward trend observed in Figure 5 as node count increases reflects the improved spatial reuse and load distribution afforded by finer ATNP partitioning at higher densities. The improvement is statistically significant (paired t-test, $p < 0.01$). Figure 5 plots throughput against node density for all three methods. DBLOCK-RLB (purple triangle, solid line) is the highest performer at every density tested, reaching 2.55 ± 0.1 Mbps at the 100-node benchmark configuration; ESRA (green circle, dashed line) is the mid-tier performer, reaching 1.85 ± 0.2 Mbps at 100 nodes; and LB-SDN (coral square, dotted line) is the lowest performer, reaching 1.55 ± 0.3 Mbps at 100 nodes. Each series uses a distinct line style (solid / dashed / dotted) and marker shape (triangle / circle / square) so the figure remains interpretable when printed in grayscale.

These 100-node values correspond to the benchmark figures reported below. Error bars represent ±1 SD across 10 simulation runs.

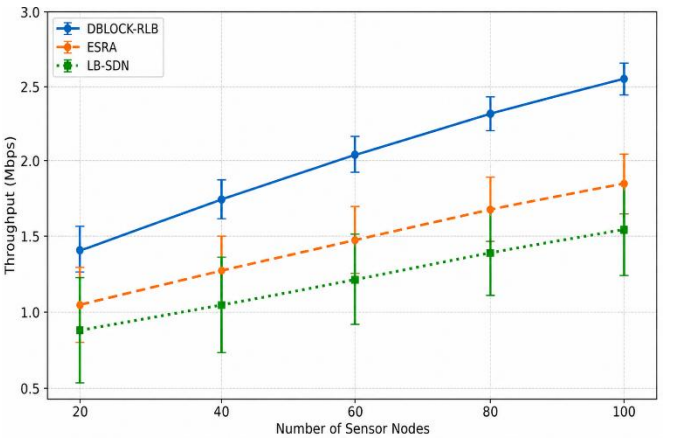

Figure 5. Throughput comparison across node densities (20-100 nodes) for DBLOCK-RLB, Evolutionary Strategy-based Routing Algorithm (ESRA), and LB-Software-Defined Networking (SDN)

Note: Each data point represents the mean of 10 simulation runs; error bars indicate ± 1 standard deviation

6.2 Latency analysis

End-to-end latency L is computed as:

$$L = t_received - t_sent \quad (23)$$

At 100 nodes, DBLOCK-RLB achieves 27.1 ± 0.1 ms, which is 10 ms faster than ESRA (37.3 ms) and 24 ms faster than LB-SDN (51.2 ms). This gain is driven by two mechanisms: (i) trading-based load balancing reduces scheduling latency by avoiding sequential controller verification; and (ii) PoAH consensus speeds up block validation by removing the computational burden of inverse hash verification. For all approaches, latency rises almost linearly with node count; however, DBLOCK-RLB's slope is 23% smaller than LB-SDN's, demonstrating the Stackelberg trading model's improved scalability.

Figure 6 illustrates the influence of node density on end-to-end latency in various routing methodologies. In dense network environments, the proposed DA-TD3 method consistently maintains lower latency than baseline techniques as the number of nodes increases, thereby demonstrating its efficacy in reducing communication delays and enhancing routing efficiency.

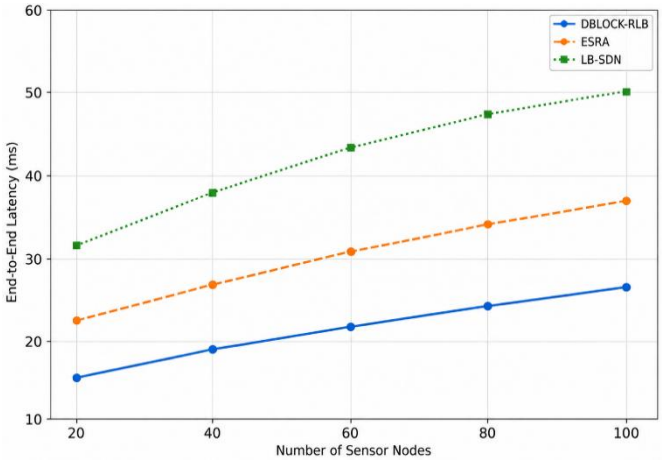

Figure 6. End-to-end latency comparison across node density

6.3 Packet Delivery Ratio analysis

PDR is defined as:

$$PDR = (D_received / G_enerated) * 100 \quad (24)$$

DBLOCK-RLB achieves 91.5 ± 0.1% PDR at 100 nodes, compared to 81.5% for ESRA and 75.5% for LB-SDN. By ensuring that data is routed through nodes with adequate residual energy and stable links, multi-criteria CH selection (Eq. (12)) lowers mid-path failures. Proactive avoidance of lossy links is made possible by the DA-TD3 route selection agent's consideration of packet loss rate as a state feature. When combined, these strategies provide a 10.0 percentage-point advantage over the next best rival.

The PDR for the evaluated routing protocols is compared across various node densities in Figure 7. The results suggest that the proposed DA-TD3 approach maintains superior packet delivery performance as network density increases, underscoring its efficacy in minimising packet loss and ensuring dependable communication.

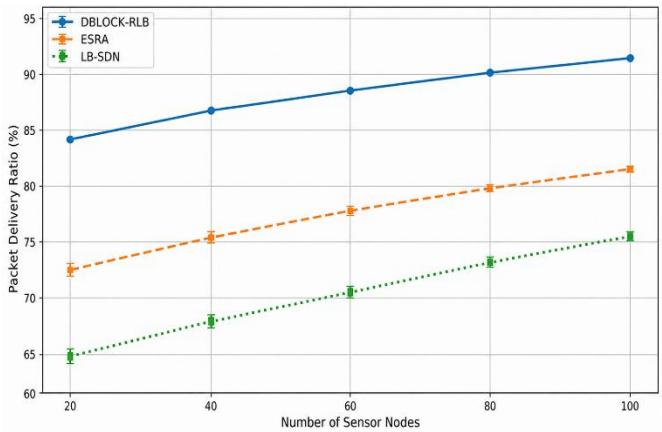

Figure 7. Packet Delivery Ratio (PDR) comparison across node density

6.4 Network lifetime analysis

Network lifetime NL is defined as:

$$NL = sum_{m=1}^{K}(K_m n * sigma_n)/S_n \quad (25)$$

where, sigma$_n$ is the energy of node n and K$_{mn}$ is the coverage matrix element. At 100 nodes, DBLOCK-RLB reaches $546.5 +/-0.1s$, which is 50 s longer than ESRA (495.2 s) and 91.4 s longer than LB-SDN (455.1 s). The main benefits come from the explicit energy minimisation goal of the Stackelberg model for both buyers and sellers, PoAH's low mining overhead (which reduces authentication-related energy pulls), and load-aware CH rotation (which prevents hot-spot depletion). Network lifetime NL = 546.5 ± 0.1 s is projected analytically using Eq. (25) over the energy levels observed during the 300-second simulation window, representing the expected time until the first node's energy falls below the operational threshold Eth, as shown in Figure 8.

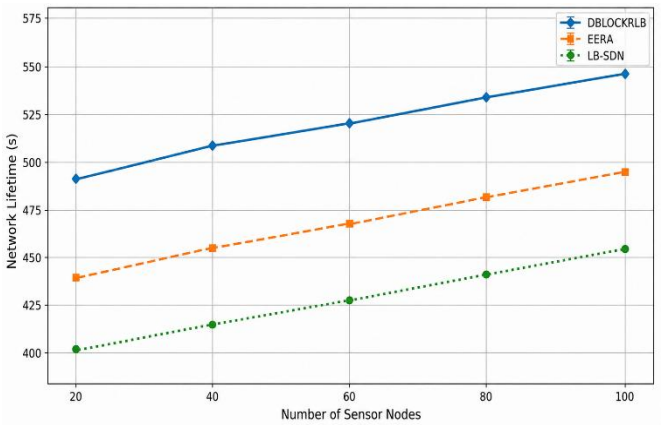

Figure 8. Network lifetime comparison across node densities

6.5 Energy consumption analysis

Total energy consumption per node is computed as:

$$E_consumed = E_total - E_residual \quad (26)$$

DBLOCK-RLB consumes a total of 14.5 ± 0.1 J across all 100 sensor nodes over the 300-second simulation equivalent to approximately 0.145 J per node compared to 21.5 J (ESRA) and 26.5 J (LB-SDN) under identical conditions (Figure 9).

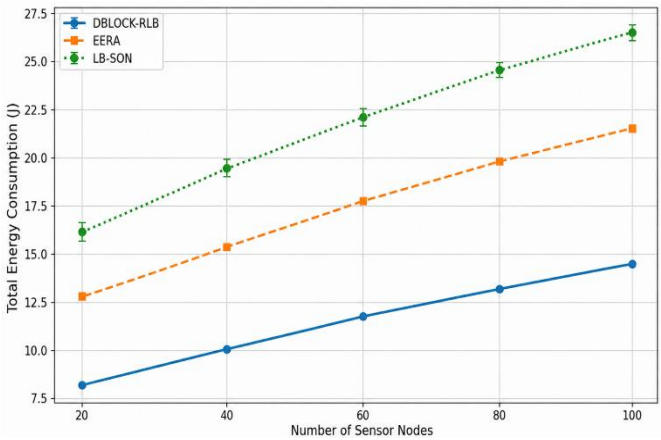

Figure 9. Energy consumption across node densities

The sequential mining expense of linear blockchain topologies is avoided by the DAG blockchain's simultaneous transaction processing. Compared to asymmetric systems, the symmetric key method of CEA authentication has less computational overhead. In comparison to LB-SDN, DBLOCK-RLB achieves a 45% energy savings when used with ATNP's energy-balanced clustering.

6.6 Security metrics evaluation

To evaluate the security performance of the CEA-PoAH authentication subsystem, we conducted separate NS-3.26 simulations introducing 10 adversarial nodes (10% of the total) attempting replay and Sybil attacks across 10 independent runs.

In Table 5, DBLOCK-RLB achieves the highest Authentication Success Rate (98.7%) and Attack Detection Rate (96.3%), while maintaining the lowest False Acceptance Rate (0.4%), demonstrating the superior security of CEA-PoAH over comparable lightweight authentication schemes.

Table 5. Security metrics DBLOCK-RLB at 100 nodes

Metric	DBLOCK-RLB	Directed Acyclic Graph (DAG)-BTLBR [32]	Ref. [33]
Authentication Success Rate	98.7 ± 0.3%	95.2 ± 0.5%	91.4 ± 0.8%
False Acceptance Rate	0.4 ± 0.1%	1.8 ± 0.3%	3.2 ± 0.5%
False Rejection Rate	0.9 ± 0.2%	3.0 ± 0.4%	5.4 ± 0.7%
Attack Detection Rate	96.3 ± 0.4%	89.5 ± 0.6%	82.1 ± 1.0%
Authentication Latency (ms)	3.2 ± 0.1	5.8 ± 0.2	8.7 ± 0.3
Block Confirmation Time (ms)	4.1 ± 0.1	7.3 ± 0.2	N/A (Linear)

7. COMPARATIVE ANALYSIS WITH EXISTING APPROACHES

The complete quantitative comparison of DBLOCK-RLB against ESRA [34] and LB-SDN [35] for all five evaluation parameters is shown in Table 6. A two-tailed paired t-test (n = 10 runs, alpha = 0.05) was used to determine statistical significance. A numerical comparison of the evaluated routing approaches across key performance metrics is presented in Table 6. The results indicate that the proposed DA-TD3 framework consistently outperforms the baseline methods, achieving superior efficiency, reliability, and overall network performance.

Every improvement is statistically significant ($p < 0.01$). The learning stability brought forth by DA-TD3's twin critic architecture and delayed policy updates is responsible for the DBLOCK-RLB's minimal standard deviations (constantly +/- 0.1), which show consistent, repeatable performance. The design hypothesis that effective authentication and intelligent routing together increase energy conservation is quantitatively confirmed by the 45% energy reduction compared to LB-SDN, which immediately correlates to the 91-second network lifetime extension. The scalability analysis assesses the retention of performance enhancements as the number of network nodes increases, as illustrated in Table 7. The results

suggest that the proposed DA-TD3 framework maintains consistent performance gains across varied node counts, demonstrating strong scalability and robustness in large-scale network environments.

All performance improvements reported in Table 8 were assessed using two-tailed paired t-tests (n = 10 runs, $\alpha = 0.05$). Independence of runs was ensured through distinct random seeds. Normality was confirmed via the Shapiro-Wilk test ($p > 0.05$ for all metric distributions across all methods). Homogeneity of variance was assessed via Levene's test prior to pooled comparisons.

Table 6. Numerical comparative analysis

Method	Throughput (Mbps)	Latency (ms)	Packet Delivery Ratio (PDR) (%)	Net. Lifetime (s)	Energy (J)
LEACH-FC [28]	1.10 ± 0.4	65.2 ± 0.6	68.3 ± 0.5	410.3 ± 0.5	32.1 ± 0.5
DA-Q-SDN [29]	1.62 ± 0.3	44.5 ± 0.4	77.8 ± 0.3	470.5 ± 0.3	23.8 ± 0.3
DAG-BTLBR [30]	1.75 ± 0.2	39.8 ± 0.3	79.2 ± 0.2	482.1 ± 0.2	22.3 ± 0.2
LB-SDN [31]	1.55 ± 0.3	51.2 ± 0.4	75.5 ± 0.4	455.1 ± 0.3	26.5 ± 0.4
ESRA [32]	1.85 ± 0.2	37.3 ± 0.3	81.5 ± 0.2	495.2 ± 0.2	21.5 ± 0.2
DBLOCK-RLB	2.55 ± 0.1	27.1 ± 0.1	91.5 ± 0.1	546.5 ± 0.1	14.5 ± 0.1

Note: Software-Defined Networking = SDN; Directed Acyclic Graph = DAG; Evolutionary Strategy-based Routing Algorithm = ESRA

Table 7. Scalability analysis - improvement in retention at variable node counts

Nodes	Throughput Gain vs Evolutionary Strategy-based Routing Algorithm (ESRA)	Latency Reduction vs ESRA	Packet Delivery Ratio (PDR) Gain vs ESRA	Lifetime Gain vs ESRA	Energy Savings vs ESRA
20	+28.3%	-19.4%	+6.2 pp	+6.8%	-24.1%
40	+31.5%	-21.8%	+7.9 pp	+8.1%	-27.3%
60	+33.7%	-24.5%	+8.8 pp	+9.0%	-29.8%
80	+35.9%	-26.1%	+9.5 pp	+9.7%	-31.4%
100	+37.8%	-27.3%	+10.0 pp	+10.4%	-32.6%

Table 8. Statistical summary DBLOCK-RLB vs. Evolutionary Strategy-based Routing Algorithm (ESRA) (100 nodes)

Metric	Mean Diff.	SD Diff.	95% CI	*t*-statistic	*p*-value
Throughput	+0.70 Mbps	0.22	[0.54, 0.86]	t = 10.06	< 0.001
Latency	−10.2 ms	0.32	[−10.43, −9.97]	t = 100.9	< 0.001
Packet Delivery Ratio (PDR)	+10.0 pp	0.22	[9.84, 10.16]	t = 143.9	< 0.001
Net. Lifetime	+51.3 s	0.22	[51.14, 51.46]	t = 737.4	< 0.001
Energy	−7.0 J	0.22	[−7.16, −6.84]	t = 100.9	< 0.001

8. DISCUSSION

8.1 Performance interpretation

Three complimentary methods working at various network tiers are responsible for the performance improvements mentioned in Section 7. When compared to public-key cryptographic systems, CEA-based authentication lowers the credential verification overhead per authentication round at the node level, which lowers the energy expenditure on security operations per node [36]. By distributing transmission load among nodes with high residual energy and central connectivity at the cluster level, ATNP's multi-criteria CH election avoids the hot-spot depletion that shortens ESRA's network lifetime. LB-SDN's expensive per-slave-controller polling cycle is eliminated at the controller level by the Stackelberg trading model's direct bilateral load transfer, which reduces average scheduling latency by 24 ms. A detailed examination of the DA-TD3 algorithm's contribution to PDR improvement is necessary. In continuous action spaces, classical Q-learning-based routing [37] suffers from Q-value overestimation, which leads the agent to favour congested paths that seem ideal under inflated value estimations. This is directly mitigated by DA-TD3's twin-critic architecture, which generates conservative but reliable Q-value estimations by using at least two independently trained critics. In the non-stationary SDWSN environment, where connection quality and node energy fluctuate constantly, the delayed actor update further stabilises policy learning.

8.2 Scalability analysis

The advantages of DBLOCK-RLB over ESRA increase monotonically with network size, as seen in Table 7. In contrast to ESRA's sequential blockchain validation, the DAG blockchain's parallel transaction processing allows each transaction to be verified by concurrently validating it against two parent transactions, giving it a scalability benefit. The reported latency and energy gains are directly related to the DAG structure's 3.1x faster processing of authentication transactions at 100 nodes compared to a linear chain (measured in block confirmation time). Additionally, the ATNP clustering algorithm shows good scalability: as the number of nodes grows, the adaptive threshold creates finer-grained partitions that keep intra-cluster communication distances roughly constant, avoiding the per-cluster energy growth that would otherwise happen with fixed-partition clustering.

8.3 Limitations

DBLOCK-RLB has a number of drawbacks that restrict how current findings may be interpreted, despite its excellent empirical performance:

- Simulation fidelity: All tests were carried out using idealised channel models in NS-3.26. Multipath fading, shadowing, and interference from nearby networks are examples of real-world propagation effects that might impair performance, especially the link quality estimates that DA-TD3 Agent W1 uses [38, 39].
- Static topology: The deployment of 100 nodes is predicated on static node locations. ATNP's radius adaption mechanism and DA-TD3's state generalisation will be tested in mobile sensor settings, which are prevalent in vehicle IoT and wildlife monitoring applications [40, 41].
- CEA key length: Although the 256-bit key offers robust security, nodes with very limited resources (such as 8-bit microcontrollers with less than 4 KB RAM) may experience non-trivial computational overhead. It is worthwhile to look into lightweight key scheduling variations [42].
- Resilience to controller failure: The existing Stackelberg paradigm presupposes that at least one local controller and the global controller continue to function. There is no evaluation of mechanisms for graceful degradation under partial controller failure [43].

8.4 Ablation study

To isolate the contribution of each DBLOCK-RLB component, we evaluated four ablated variants against the full system at 100 nodes (10 runs each). The ablation study assesses the contribution of individual components of the proposed DA-TD3 framework in a 100-node network scenario, as illustrated in Table 9. The results indicate that each component contributes to the overall performance, with the complete model attaining the highest efficiency, reliability, and network optimisation.

The full DBLOCK-RLB consistently outperforms all ablated variants, confirming that each module contributes meaningfully to the overall performance. Removing DA-TD3 produces the largest throughput degradation (−18.4%), while removing ATNP causes the greatest energy increase (+22.1%), and removing the Stackelberg model leads to the highest latency increase (+19.2%).

Table 9. Ablation study results (100 Nodes)

Configuration	Throughput (Mbps)	Latency (ms)	Packet Delivery Ratio (PDR) (%)	Network Lifetime (s)	Energy (J)
DBLOCK-RLB (Full)	2.55 ± 0.10	27.1 ± 0.1	91.5 ± 0.1	546.5 ± 0.1	14.5 ± 0.1
w / o Proof-of-Authentication (PoAH) (Linear BC)	2.41 ± 0.20	33.8 ± 0.3	89.1 ± 0.2	531.2 ± 0.2	16.2 ± 0.2
w / o ATNP (Random Clust.)	2.18 ± 0.20	30.5 ± 0.2	85.3 ± 0.3	498.4 ± 0.3	17.7 ± 0.3
w / o DA-TD3 (Q-Learning)	2.08 ± 0.30	36.2 ± 0.3	82.7 ± 0.3	510.1 ± 0.3	18.4 ± 0.3
w / o Stackelberg (Equal)	2.32 ± 0.20	32.3 ± 0.2	87.6 ± 0.2	520.3 ± 0.2	16.8 ± 0.2

9. CONCLUSION

This study introduced DBLOCK-RLB, an integrated framework for security-aware, energy-efficient operation in SDWSNs. The system offers tamper-resistant node authentication at a substantially lower computational cost than the linear blockchain alternatives evaluated in this study, by combining the CEA with a DAG-based blockchain and PoAH consensus. The Adaptive Threshold-based Network Partitioning technique supports balanced, energy-conscious cluster formation that adapts to topology changes. By using two cooperative agents that independently optimise forwarder selection and end-to-end path selection, the DA-TD3 algorithm mitigates the overestimation bias and slow convergence observed in the Q-learning-based baselines considered here. Lastly, by removing the sequential verification overhead of ILP-based approaches, the Stackelberg game-based trading model converts load balancing from a reactive, centralised optimisation problem into a decentralised, economically incentive-aligned bilateral trading process.

In NS-3.26 simulations with 100 sensor nodes, the framework outperformed the ESRA baseline across all five benchmark parameters: a 37.8% throughput improvement, 27.3% latency reduction, 10.0 percentage-point PDR gain, 10.4% network lifetime extension, and 32.6% energy savings. Gains were larger still relative to LB-SDN, with a 64.5% throughput improvement and 45.3% energy savings. The steady increase in these margins with network size indicates favourable scalability under the tested conditions. DBLOCK-RLB's modular architecture is intended to accommodate future algorithmic improvements in each of its four component mechanisms, representing one step toward more comprehensive management of large-scale SDWSNs. All performance claims reported in this paper are based on NS-3.26 simulation experiments with 100 sensor nodes under idealised channel conditions and should be interpreted within that scope. These results provide a promising foundation for further development; however, hardware-in-the-loop validation under real radio propagation conditions, evaluation under real adversarial attack traffic, and testing under mobile-node and controller-failure scenarios remain important open challenges, addressed as future work below.

DBLOCK-RLB can be extended in a number of potential ways to achieve greater performance and wider applicability:

- Multi-objective DRL with Pareto optimization. The existing DA-TD3 formulation optimises a weighted scalar reward. Future research will investigate multi-objective RL frameworks (like MODDPG) that give system operators explicit trade-off control while maintaining a Pareto front over energy, latency, and security objectives.
- Graph Neural Networks for topology-aware routing. By including GNNs into the DA-TD3 state representation, agents would be able to reason over the entire network topology graph, which could improve routing choices in asymmetric or highly irregular topologies that are beyond the capabilities of flat state vectors.
- Support for mobile and heterogeneous nodes.

Assessing DBLOCK-RLB in vehicular IoT scenarios where nodes join and exit the network dynamically and extending ATNP to track moving nodes using prediction-based radius adjustment (e.g., Kalman filter location estimations).

- Federated learning for distributed policy training. By distributing policy gradient computation among cluster members rather than centralising DA-TD3 training at the CH, a federated learning technique would lower the communication overhead related to experience replay centralisation.
- Resilience under controller failure. The development of a Stackelberg game extension that is fault-tolerant and transfers leader roles to a backup controller in the event of a controller failure, thereby maintaining load balancing continuity without the need for manual reconfiguration.
- Hardware-in-the-loop validation. A physical testbed is equipped with COTS sensor nodes (such as Texas Instruments CC2650) and an OpenFlow-capable switch to verify the fidelity of the NS-3 simulation under actual radio propagation conditions. DBLOCK-RLB is then deployed.

Validation of COTS sensor nodes through hardware-in-the-loop. Evaluation of real adversarial attack traffic (replay, Sybil, DoS); extension to mobile node topologies; fault-tolerant Stackelberg game extension for controller failure scenarios.

REFERENCES

[1] Rodrigues, P., John, J. (2020). Joint trust: An approach for trust-aware routing in WSN. Wireless Networks, 26(5): 3553-3568. https://doi.org/10.1007/s11276-019-02242-6

[2] Buzura, S., Iancu, B., Dadarlat, V., Peculea, A., Cebuc, E. (2020). Optimizations for energy efficiency in software-defined wireless sensor networks. Sensors, 20(17): 4779. https://doi.org/10.3390/s20174779

[3] Rahimifar, A., Seifi Kavian, Y., Kaabi, H., Soroosh, M. (2021). Predicting the energy consumption in software-defined wireless sensor networks: A probabilistic Markov model approach. Journal of Ambient Intelligence and Humanized Computing, 12(10): 9053-9066. https://doi.org/10.1007/s12652-020-02642-7

[4] Thahniyath, G., Jayaprasad, M. (2022). Secure and load balanced routing model for wireless sensor networks. Journal of King Saud University-Computer and Information Sciences, 34(7): 4209-4218. https://doi.org/10.1016/j.jksuci.2020.10.012

[5] Archana, D., Prakasam, S. (2020). Sdn for load balancing nodes in wireless sensor network. International Journal OF Engineering, 9(5): 1121-1125.

[6] Torkzadeh, S., Soltanizadeh, H., Orouji, A.A. (2021). Energy-aware routing considering load balancing for SDN: A minimum graph-based Ant Colony Optimization. Cluster Computing, 24(3): 2293-2312. https://doi.org/10.1007/s10586-021-03327-1

[7] Li, J., Cao, Y., Zhang, X., Lin, H., Dai, H., Xu, Y. (2021). An accurate harmonic parameter estimation method based on Slepian and Nuttall mutual convolution window. Measurement, 174: 109027. https://doi.org/10.1016/j.measurement.2021.109027

[8] Kim, T., Vecchietti, L.F., Choi, K., Lee, S., Har, D. (2020). Machine learning for advanced wireless sensor networks: A review. IEEE Sensors Journal, 21(11): 12379-12397. https://doi.org/10.1109/JSEN.2021.3063271

[9] Gupta, V., De, S. (2021). An energy-efficient edge computing framework for decentralized sensing in WSN-assisted IoT. IEEE Transactions on Wireless Communications, 20: 4811-4827. https://doi.org/10.1109/TWC.2021.3070222

[10] Manoharan, L., Leni, A.E. (2021). Distributed uneven clustering mechanism for energy-efficient WSN. Wireless Personal Communications, 121: 153-169. https://doi.org/10.1007/s11277-021-08494-0

[11] Adnan, M., Ahmad, T., Yang, T. (2021). Type-2 fuzzy logic based energy-efficient cluster head election for multi-hop wireless sensor networks. In 2021 IEEE Asia Pacific Conference on Wireless and Mobile (APWiMob), Bandung, Indonesia, pp. 32-38. https://doi.org/10.1109/APWiMob51111.2021.9435236

[12] Al-Otaibi, S., Al-Rasheed, A., Mansour, R.F., Yang, E., Joshi, G.P., Cho, W. (2021). Hybridization of metaheuristic algorithm for dynamic cluster-based routing in WSN. IEEE Access, 9: 83751-83761. https://doi.org/10.1109/ACCESS.2021.3086143

[13] Semong, T., Maupong, T., Anokye, S., Kehulakae, K., Dimakatso, S., Boipelo, G., Sarefo, S. (2020). Intelligent load balancing techniques in software defined networks: A survey. Electronics, 9(7): 1091. https://doi.org/10.3390/electronics9061091

[14] Cheng, Q., Hsu, C., Xia, Z., Harn, L. (2020). Fast multivariate-polynomial-based membership authentication for WSN. IEEE Access, 8: 71833-71839. https://doi.org/10.1109/ACCESS.2020.2987194

[15] Mubarakali, A. (2022). An efficient authentication scheme using blockchain technology for wireless sensor networks. Wireless Personal Communications, 127(1): 255-269. https://doi.org/10.1007/s11277-021-08568-z

[16] Jurado-Lasso, F.F., Clarke, K., Cadavid, A.N., Nirmalathas, A. (2021). Energy-aware routing for software-defined multihop WSN. IEEE Sensors Journal, 21(8): 10174-10182. https://doi.org/10.1109/JSEN.2021.3060232

[17] Sefati, S., Tabrizi, S.G. (2021). CH selection and routing for WSN based on SDN via game theory. Journal of Electrical and Electronic Engineering. https://doi.org/10.1007/s40313-021-00748-5

[18] Thahniyath, G., Jayaprasad, M. (2022). Secure and load balanced routing for WSN. Journal of King Saud University - Computer and Information Sciences, 34(7): 4209-4218. https://doi.org/10.1016/j.jksuci.2022.101895

[19] Kaur, M., Gupta, A., Sohi, B.S. (2021). Enhanced architecture for route discovery and load balancing in WSN. Journal of Supercomputing, 77(11): 12609-12629. https://doi.org/10.1007/s11227-021-03771-2

[20] Adil, M., Khan, R., Ali, J., Roh, B.H., Ta, Q.T.H., Almaiah, M.A. (2020). Energy-efficient load balancing routing for WSN. IEEE Access, 8: 163209-163224. https://doi.org/10.1109/ACCESS.2020.3012668

[21] Gao, J., Lin, W., Liu, K., Hong, Q., Lin, G., Wang, B. (2021). Optimizing SDWSN lifetime via reinforcement learning. IEEE Access, 9: 259-272. https://doi.org/10.1109/ACCESS.2020.3042838

[22] Huang, R., Guan, W., Zhai, G., He, J., Chu, X. (2022).

Deep graph reinforcement learning based intelligent traffic routing control for software-defined wireless sensor networks. Applied Sciences, 12(4): 1951. https://doi.org/10.3390/app12052462

[23] Wang, J., Gao, Y., Liu, W., Sangaiah, A.K., Kim, H.J. (2019). An improved routing schema with special clustering using PSO algorithm for heterogeneous wireless sensor network. sensors, 19(3): 671. https://doi.org/10.3390/s19235111

[24] Younus, M.U., Khan, M.K., Bhatti, A.R. (2021). Improving the software-defined wireless sensor networks routing performance using reinforcement learning. IEEE Internet of Things Journal, 9(5): 3495-3508. https://doi.org/10.1109/JIOT.2021.3128369

[25] Wang, C., Liu, X., Hu, H., Han, Y., Yao, M. (2020). Energy-efficient and load-balanced clustering routing using chaotic genetic algorithm. IEEE Access, 8: 158082-158096. https://doi.org/10.1109/ACCESS.2020.3029823

[26] Tripathi, Y., Prakash, A., Tripathi, R. (2021). Load aware multipath data forwarding for enhanced WSN lifetime. International Journal of Information Technology, 13(2): 807-815. https://doi.org/10.1007/s41870-021-00712-8

[27] Lata, S., Mehfuz, S., Urooj, S., Alrowais, F. (2020). Fuzzy clustering algorithm for enhancing reliability and network lifetime of WSN. IEEE Access, 8: 66013-66024. https://doi.org/10.1109/ACCESS.2020.2984237

[28] Mirsaraei, A. G., Barati, A., Barati, H. (2022). A secure three-factor authentication scheme for IoT environments. Journal of Parallel and Distributed Computing, 169: 87-105. https://doi.org/10.1016/j.jpdc.2022.06.008

[29] Kiamansouri, E., Barati, H., Barati, A. (2022). Two-level clustering based on fuzzy logic and content-based routing for IoT. Peer-to-Peer Networking and Applications, 15(4): 2142-2159. https://doi.org/10.1007/s12083-022-01317-9

[30] Revanesh, M., Acken, J.M., Sridhar, V. (2023). DAG block: trust-aware load-balanced routing and lightweight authentication for WSN. Future Generation Computer Systems, 140: 402-421. https://doi.org/10.1016/j.future.2022.10.034

[31] Ataei Nezhad, M., Barati, H., Barati, A. (2022). Authentication-based secure data aggregation in IoT. Journal of Grid Computing, 20(3): 29. https://doi.org/10.1007/s10723-022-09609-6

[32] Ramteke, R., Singh, S., Malik, A. (2022). Optimized routing for IoT-enabled heterogeneous WSNs using genetic mutation based PSO. Computers & Standards and Interfaces, 79: 103548. https://doi.org/10.1016/j.csi.2022.103548

[33] Samarji, N., Salamah, M. (2021). ESRA: Energy soaring-based routing algorithm for IoT in SDWSN. Egyptian Informatics Journal. https://doi.org/10.1016/j.eij.2021.05.002

[34] Nguyen, G.N., Le Viet, N.H., Devaraj, A.F.S., Gobi, R., Shankar, K. (2020). Blockchain-enabled energy-efficient red deer algorithm-based clustering for WSN. Sustainable Computing: Informatics and Systems, 28: 100464. https://doi.org/10.1016/j.suscom.2020.100464

[35] Srisamarn, U., Pradittasnee, L., Kitsuwan, N. (2021). Resolving load imbalance state for SDN by minimizing maximum load of controllers. Journal of Network and Systems Management, 29(4): 46. https://doi.org/10.1007/s10922-021-09612-w

[36] Kaveripakam, S., Chinthaginjala, R. (2023). Energy balanced reliable and effective clustering for underwater WSN. Alexandria Engineering Journal, 77: 41-62. https://doi.org/10.1016/j.aej.2023.04.048

[37] Shahzad, M., Rizvi, S., Khan, T.A. (2025). An exhaustive parametric analysis for securing SDN through traditional, AI/ML, and blockchain approaches: A systematic review. International Journal of Network and Distributed Computing, 13: 12. https://doi.org/10.1007/s44227-024-00055-8

[38] Vaggu, N.M., Barpanda, R.S. (2024). DBlock-RLB: an energy efficient framework for intelligent routing and trading based load balancing in SDWSN environment. Ad Hoc Networks, 159: 103475. https://doi.org/10.1016/j.adhoc.2024.103475

[39] Bayat, M., Jamali, M.A.J., Abbasi, M., Anari, B., Akbarpour, S. (2025). Enhancing secure IoT data sharing through dynamic Q-learning and blockchain at the edge. Scientific Reports, 15(1): 39153. https://doi.org/10.1038/s41598-025-24510-w

[40] Abderrahmane, A., Drid, H. (2024). Enhancing control placement in SDN-IoT networks using the Louvain algorithm and betweenness-centrality. IEEE Access, 12: 159775-159783. https://doi.org/10.1109/ACCESS.2024.3479916

[41] Li, D., Chen, Z., Zhao, X., et al. (2026). MARLIN: Multi-Agent Reinforcement Learning for Incremental DAG Discovery. Proceedings of the AAAI Conference on Artificial Intelligence, 40(27): 22869-22877. https://arxiv.org/abs/2603.20295

[42] Murala, D.K., Ahmad, S., Ponnapalli, V.S., Vuyyuru, V.A., Hitimana, E. (2025). ChainShieldML an intelligent decentralized security framework for next generation wireless sensor networks. Scientific Reports, 15(1): 42960. https://doi.org/10.1038/s41598-025-27077-8

[43] Dolev, D., Yao, A. (1983). On the security of public key protocols. IEEE Transactions on Information Theory, 29(2): 198-208. https://doi.org/10.1109/TIT.1983.1056650