SvedbergOpen
DISSEMINATION OF KNOWLEDGE
International Journal of Artificial Intelligence and Machine Learning
Publisher's Home Page: https://www.svedbergopen.com/
Research Paper Open Access

A Blockchain-Integrated SDN Framework for Energy-Efficient, Latency-Aware, and Secure IoT Routing Using Dynamic Objective Selection Reinforcement Learning (DOS-IRL) and Adaptive Sea Horse Optimizer (ASHO)

Mrs. M. Senthamil Selvi[1], Dr. L. Sudha[2]

[1]Research Scholar, Department of Computer Science, VET Institute of Arts and Science (Co-education) College, Erode, Tamil Nadu, India.
[2]Associate Professor, Department of Computer Science, VET Institute of Arts and Science (Co-education) College, Erode, Tamil Nadu, India

Abstract
The rapid growth of the Internet of Things (IoT) presents significant challenges in optimizing network performance, ensuring security, and managing energy consumption in a dynamic and heterogeneous environment. This paper proposed Blockchain-Enabled Software-Defined Networking (SDN) architecture for energy-efficient, latency-aware, and secure IoT routing. The framework integrates an advanced Dynamic Objective Selection Reinforcement Learning (DOS-IRL) agent, which adapts routing decisions based on real-time network conditions, optimizing energy usage, minimizing latency, and improving packet delivery. To enhance the agent's convergence and stability, an Adaptive Sea Horse Optimizer (ASHO) is employed for continuous hyperparameter tuning. A Modified E-Butterfly Optimization Algorithm (MBOA) is introduced to detect and isolate malicious nodes, leveraging trust metrics based on node behavior, and ensuring secure communication. Additionally, blockchain-based trust management provides tamper-proof records of node behaviors, ensuring transparency and accountability in the network. Experimental results demonstrate that the proposed framework outperforms conventional routing protocols by significantly improving network lifetime, reducing latency, enhancing packet delivery ratios (PDR), and strengthening security. The combination of SDN, reinforcement learning, optimization algorithms, and blockchain presents a robust, scalable solution for next-generation IoT networks, addressing key challenges of dynamic adaptability, cybersecurity, and resource efficiency.

Keywords: Internet of Things (IoT), Software-Defined Networking (SDN), Energy Efficiency, Latency Optimization, Secure Routing, Dynamic Objective Selection Reinforcement Learning (DOS-IRL), Adaptive Sea Horse Optimizer (ASHO), Modified E-Butterfly Optimization Algorithm (MBOA), Blockchain, Network Security.

1. Introduction

The rapid proliferation of the Internet of Things (IoT) has ushered in a new era of interconnected devices, driving significant advances in smart homes, healthcare, industrial automation, and more. IoT networks, composed of a wide range of heterogeneous devices, are continuously evolving and expanding, resulting in a dynamic and complex environment that presents several challenges. Among these challenges, ensuring energy efficiency, minimizing latency, and securing data integrity are paramount for the reliable operation of IoT networks. As IoT devices are often resource-constrained, effective energy management is crucial to prolonging network lifetime, while low-latency communication is critical for real-time applications that demand immediate data processing and response.

The exponential growth of Internet of Things (IoT) devices, accelerated by the deployment of 5G networks and the rise in Industrial IoT (IIoT) applications, has transformed network infrastructures while expanding their attack surfaces. While IoT connects consumer devices for automation, IIoT brings these capabilities to industrial environments using sensors, actuators, and edge computing in sectors including manufacturing and energy. The systems demand low latency, reliability, and security for real-time monitoring and predictive maintenance [1,2]. However, their heterogeneity, diverse communication protocols, and resource-constrained endpoints introduce security challenges. For instance, when cloud storage is used for data offloading and processing, concerns are raised over confidentiality, integrity, and protection of sensitive information. Traditional centralised security approaches struggle to meet these requirements at scale, creating demand for more flexible and adaptive security models [3, 4]. Distributed Denial of Service (DDoS) attacks have become a persistent threat capable of overwhelming network resources, disrupting legitimate traffic, and causing large-scale service outages across IoT, IIoT, and Software-Defined Networking (SDN) environments [5, 6]. Although SDN introduces programmability, global visibility, and separation of the control and data planes, its centralised architecture exposes weaknesses such as controller saturation, flow table overflow, and single points of failure (SPF), while

inter-controller communications remain prone to spoofing, replay, and man-in-the-middle (MITM) attacks [7]. Integrating SDN and IoT (SDN-IoT) offers scalable resource allocation and traffic management for large deployments but also introduces new risks, including controller compromise, flow-rule manipulation, topology poisoning, and insecure inter-device communications [8].

Addressing these challenges requires multi-layered security architectures, as traditional perimeter-based models are insufficient for dynamic and distributed IoT ecosystems [9]. In this regard, blockchain (BC) has gained prominence for providing decentralisation, immutability, and trust through consensus mechanisms, tamper-proof logging, and secure access control [10]. Integrating BC with SDN supports multi-controller architecture, load balancing, secure flow-rule management, and energy-efficient routing, thereby mitigating the limitations of centralised control. At the same time, Artificial Intelligence (AI), particularly machine learning (ML) and DL techniques, have shown strong capabilities in detecting and mitigating DDoS attacks. Methods such as Artificial Neural Networks (ANN), Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM), Convolutional Neural Networks (CNN), and ensemble learning models have shown high accuracy and adaptability [11, 12]. In addition, feature engineering strategies, including Recursive Feature Elimination (RFE), Sequential Forward Selection (SFS), and Minimum Redundancy Maximum Relevance (mRMR), further improve model efficiency, while hierarchical and online learning frameworks enable adaptive responses to evolving threats [13]. In the same vein, federated learning (FL) has also emerged as a promising approach by enabling privacy-preserving collaborative training at the network edge, thereby reducing the risks associated with transmitting raw IoT data to central servers [14]. Combined with BC and SDN programmability, FL provides a basis for integrated frameworks that support decentralised trust, secure controller coordination, and intelligent traffic analysis in IoT and 5G environments [15].

Simultaneously, security remains a significant concern as malicious nodes can disrupt network operations, manipulate data, or cause traffic congestion, compromising the network's reliability and trustworthiness. Given the scale and diversity of IoT networks, traditional methods of routing, energy management, and security fail to adequately address the dynamic conditions and evolving demands of these systems. Therefore, there is an increasing need for intelligent and adaptable network management frameworks that can simultaneously optimize performance and ensure security.

This paper proposed a Blockchain-Enabled Software-Defined Networking (SDN) architecture for energy-efficient, latency-aware, and secure IoT routing. The proposed framework integrates an advanced Dynamic Objective Selection Reinforcement Learning (DOS-IRL) agent for intelligent and adaptive routing decisions. The agent is further enhanced with Adaptive Sea Horse Optimizer (ASHO) for real-time hyperparameter tuning, which allows it to efficiently converge to optimal routing policies under varying network conditions. To address security, the framework employs the Modified E-Butterfly Optimization Algorithm (MBOA) for the detection and isolation of malicious nodes, ensuring the integrity and trustworthiness of the network. Additionally, blockchain-based trust management is incorporated into the framework, enabling transparent, immutable records of node behaviors and ensuring that only trustworthy nodes are involved in routing decisions. This integrated approach combining SDN, reinforcement learning, metaheuristic optimization, and blockchain offers a robust, scalable, and highly adaptive solution tailored for the complexities of modern IoT networks. Through extensive experimental evaluations, we show that our framework delivers substantial improvements over conventional methods, achieving superior network lifetime, latency reduction, packet delivery ratio (PDR), and enhanced security resilience, addressing the dynamic and diverse demands of next-generation IoT systems.

2. Related Works

Numerous studies have focused on improving security, efficiency, and resource management in IoT networks by leveraging Software-Defined Networking (SDN), Reinforcement Learning (RL), blockchain, and metaheuristic optimization techniques.

Al-Quayed et al. [12] introduced a secure decision-making technique that integrates Reinforcement Learning (RL) with blockchain (BC) to improve data protection and trust in cognitive methods. Their approach enhances system reliability by ensuring secure communication and offering more flexibility in dealing with IoT network instability and inaccuracies. Sahu et al. [13] proposed a multi-objective optimization framework for smart parking systems, integrating Digital Twin (DT) technology, Markov Decision Processes (MDP), Particle Swarm Optimization (PSO), and Pareto Front Optimization (PFO). Their framework improves the distribution of solutions through dynamic adjustments based on real-time data, effectively enhancing the parking system's efficiency.

Chen et al. [14] developed an improved geospatial sensor web (GSW) that integrates spatio-temporal modeling (STM) techniques and IoT protocols, validated over a City Sensing Base Station (CSBS). The experiment showed that their framework supports emergency response through fast aerial-ground system formation, utilizing multiple sensing sources and IoT protocols for efficient data handling. In another study, probability-based hybrid whale-dragonfly optimizers (p-H-WDFOA) were introduced by Far et al. [15] for

smart urban vehicle transportation, reducing latency in edge-computing environments. Their approach tackles edge computing issues in 5G localized MEC servers to optimize resource management in smart cities.

Tian et al. [16] proposed a comprehensive security framework integrating an ELM-based replicator neural network (ELM-RNN) with a Deep Q-Networks (DRL-DQN) technique. Their model enhances data protection and privacy by incorporating secure trust-aware philosopher privacy mechanisms and Garson's Algorithm (GA) for optimization, mitigating security threats in IoT networks. Xu, Nagothu, and Chen [17] presented an autonomous and resilient edge framework (AR-EC) combining SDN, blockchain, and AI technologies. Their approach addresses edge resource optimization and security resilience in decentralized IoT networks, utilizing federated microchain fabrics for increased network robustness in intelligent transportation systems (ITS). Wang et al. [19] proposed NeuroSpatialIOT, an intuitive smart home control system using 2D spatial mapping and eye tracking in combination with Deep Learning (DL) to interpret user intent. This system enhances user interaction and IoT device control through adaptive learning techniques.

Far et al. [20] explored integrating blockchain and deep RL models to optimize mobile transmission and secure data exchange in IoT-assisted smart cities. Their model improves privacy, security, and system efficiency by enhancing the data integrity and reducing communication overhead in smart city applications. Khan et al. [21] developed an energy-efficient parallel computation offloading mechanism in multi-task multi-server mobile edge computing (EC) environments, improving latency and energy usage through deep learning-based offloading techniques.

Mishra and Chaurasiya [22] proposed a hybrid DL-based approach, combining Long Short-Term Memory (LSTM) and Support Vector Machine (SVM) classifiers to detect and prevent cyber-attacks in IoT networks. Their method integrates min-max normalization for data preprocessing, optimizing feature selection using the Reptile Search Algorithm (RSA) and blockchain technology. Ficili et al. [23] examined the integration of IoT, Cloud Computing (CC), Edge Computing (EC), and AI to enable real-time decision-making in pervasive environmental systems, advancing environmental intelligence in IoT applications.

Wang et al. [24] proposed an AI-enhanced multi-stage learning-to-learning (MSLL) approach using MMSTransformer to optimize load management in smart cities, improving forecasting accuracy and ensuring data security. Lilhore et al. [25] introduced a hybrid deep Q-network (DQN)-proximal policy optimization (PPO)-graph neural network (GNN)-RL model for resource allocation in dynamic IoT environments. Their model enhances efficiency, reduces operational costs, and improves system performance in real-time applications. Ahmed and Elena [26] explored the integration of AI techniques such as Machine Learning (ML), Federated Learning (FL), and intelligent orchestration frameworks with Edge Computing to enhance decision-making and reduce latency in smart cities, industrial IoT, and 5G/6G networks.

Qasim Jebur Al-Zaidawi and Çevik [27] proposed a hybrid optimization technique, combining Grey Wolf Optimization (GWO) with Particle Swarm Optimization (PSO) and Harris Hawks Optimization (HHO) for real-time anomaly detection in IoT networks. Their method efficiently balances global exploration and local exploitation, enhancing the performance of Deep Learning (DL) models for anomaly detection.

3. Proposed Methodology

The proposed methodology outlines a comprehensive framework for energy-efficient, latency-aware, and secure IoT routing using a Blockchain-Enabled SDN Architecture. This architecture integrates advanced techniques such as Dynamic Objective Selection Reinforcement Learning (DOS-IRL), Modified Sea Horse Optimizer (MSHO), Modified E-Butterfly Optimization Algorithm (MBOA), and blockchain-based trust management. The methodology is designed to address the challenges of energy efficiency, low latency, security, and dynamic adaptability in modern IoT networks. The following steps outline the key components of the proposed methodology:

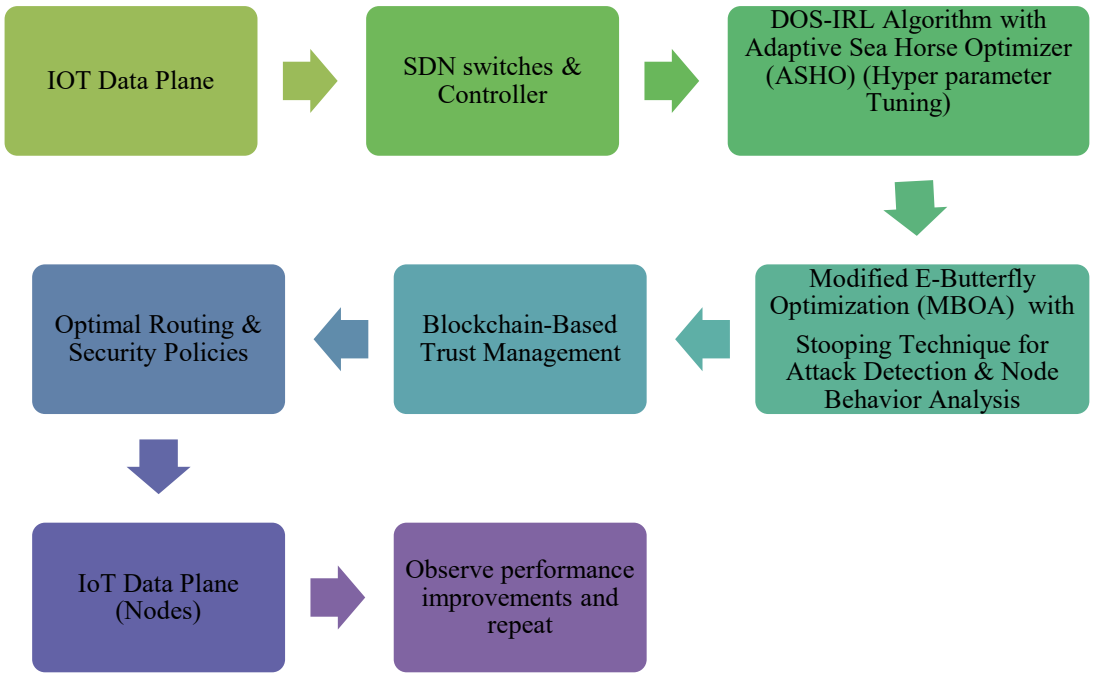


Figure 1: Proposed Methodology flow

A. Initialization and System Overview

The proposed framework introduces an integrated closed-loop Software-Defined Networking (SDN) architecture enhanced with blockchain-based trust management, specifically designed to improve routing efficiency, latency awareness, energy optimization, and security in dynamic Internet of Things (IoT) environments. The system combines intelligent routing, real-time optimization, and trust-aware security mechanisms within a unified architecture.

IoT Data Plane

- Comprises sensor nodes, local gateways, and monitoring units responsible for sensing, data collection, aggregation, and packet forwarding.
- Operates under strict energy constraints and continuously generates network traffic, requiring efficient, low-latency, and reliable data transmission.
- Provides real-time network metrics such as energy levels, delay, packet delivery ratio (PDR), and node behavior, which are forwarded to the SDN controller.

SDN Controller (Control Plane)

- Acts as the centralized intelligence unit for global network monitoring, routing decision-making, and policy enforcement.
- Collects real-time network state information from the IoT data plane and dynamically updates flow rules.
- Coordinates with the DOS-IRL agent, ASHO optimizer, and MBOA security module to ensure optimal routing and secure communication.

Dynamic Objective Selection Improved Reinforcement Learning (DOS-IRL) Agent

- Responsible for adaptive multi-objective routing decisions in real time.
- Dynamically balances key performance metrics, including:
  - Energy consumption
  - Latency
  - Packet Delivery Ratio (PDR)
- Employs an improved reinforcement learning mechanism with dynamic objective prioritization, enabling the system to respond effectively to varying network conditions.

Adaptive Sea Horse Optimizer (ASHO)

- Performs real-time hyperparameter tuning of the DOS-IRL agent to enhance learning efficiency and convergence stability.
- Optimizes parameters such as learning rate, discount factor, and exploration rate.
- Utilizes adaptive exploration–exploitation strategies inspired by seahorse behavioral dynamics to ensure faster convergence and improved routing performance.

Modified E-Butterfly Optimization Algorithm (MBOA) Security Mechanism

- Detects and isolates malicious or unreliable nodes within the network.
- Utilizes a stooping-based search mechanism to accelerate detection accuracy and convergence.

- Incorporates trust metrics, including:
  - Packet forwarding success
  - Historical node reliability
  - Neighbor reputation
- Ensures that only trustworthy nodes participate in routing, thereby improving network security and reliability.

Blockchain-Based Trust Management

- Maintains a decentralized, tamper-proof ledger of node behavior and trust values.
- Ensures transparency, immutability, and accountability in trust evaluation.
- Supports secure decision-making by providing verified trust records to the SDN controller and MBOA module.

Closed-Loop Feedback Mechanism

- Establishes continuous interaction between:
  - IoT Data Plane
  - SDN Controller
  - DOS-IRL Agent
  - MSHO Optimizer
  - MBOA Security Module
  - Blockchain Layer

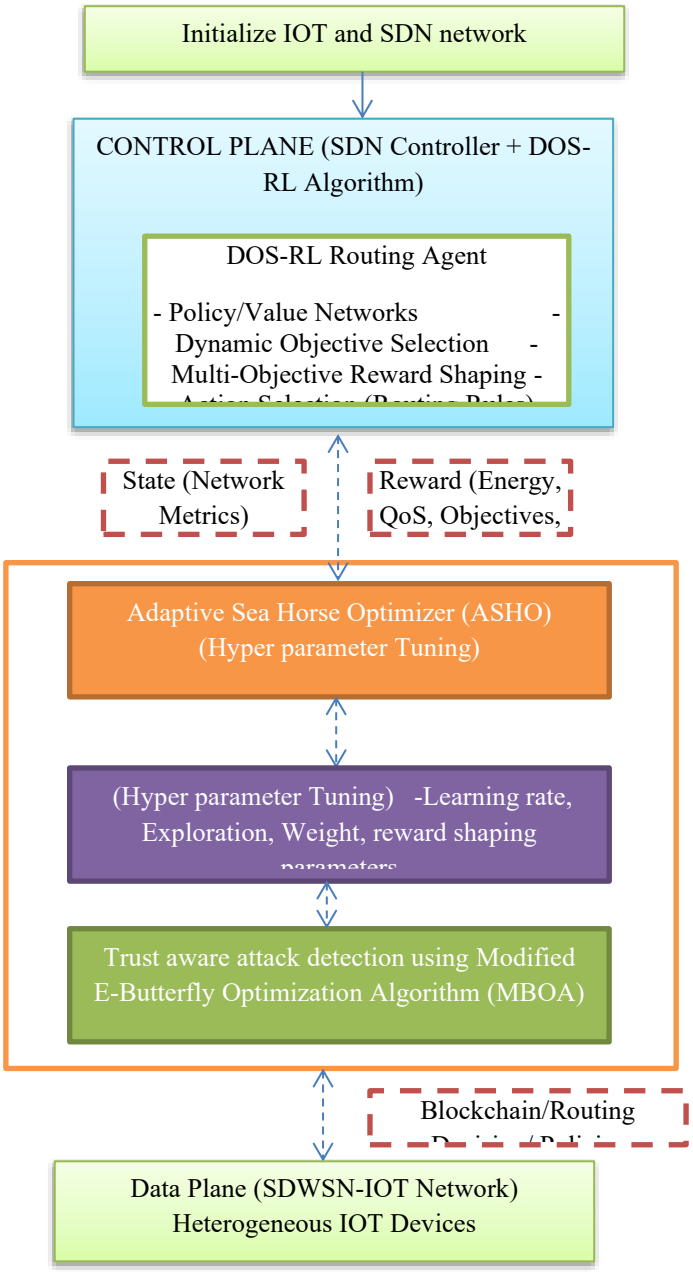


Figure 2: Proposed Interconnected Diagram

B. SDN Controller (Control Plane): Centralized decision-making and policy application

The SDN controller serves as the brain of the network, responsible for centralized decision-making, policy generation, and application. The SDN controller manages the control plane and communicates with the data plane (sensor nodes, gateways, and monitoring systems).

Functions of the SDN Controller:

Centralized Decision Making:

o The SDN controller receives real-time metrics from the data plane (IoT nodes, gateways, and monitoring systems).

o It processes these metrics to make high-level decisions about network behavior, such as optimizing routing, managing congestion, or deciding when to reroute traffic.

Policy Application:

o The SDN controller generates routing policies based on the **DOS-RL agent**'s decisions (using reinforcement learning).

o It pushes these policies to the data plane for execution. The policies can include routing tables, QoS (Quality of Service) parameters, and energy management instructions.

Dynamic Policy Adjustment:

o The controller continuously adjusts policies based on feedback from the monitoring systems. If performance metrics like energy, latency, or packet loss deteriorate, the controller may trigger policy updates to improve the network's performance.

Monitoring and Feedback Loop:

o The controller continuously monitors the feedback from the AIoT data plane and uses Bayesian Optimization (BO) to fine-tune the DOS-RL agent's hyperparameters for better routing decisions.

SDN Controller Workflow:

- Collect network data from the AIoT data plane (sensor nodes, gateways, and monitoring systems).
- Use the DOS-RL agent to evaluate the network state and decide on optimal routing actions.
- Apply these routing decisions as network policies and push them to the data plane.
- Continuously monitor performance and adjust policies based on real-time feedback, using BO to optimize the DOS-RL agent.

C. The DOS-RL Routing Scheme

The realistic conditions of an SDWSN-IoT network are subjected to dynamic changes in resources such as battery capacity, CPU capacity, memory, and bandwidth, as well as link quality changes during network operation. RL-based routing schemes have demonstrated considerable advantages in designing network operation policies that can handle such changes. However, such routing protocols often fail to respond quickly enough to such changes. To address this issue, we propose the DOS-RL routing scheme which allows the RL agent to learn from multiple correlated objectives simultaneously and to adaptively choose the objective it believes is the most promising from the current state. Therefore, we have turned the problem of energy efficiency into an RL process by modeling it into a four-tuple(S, A, P, R), defining the states (S), actions (A), policy (P), and reward functions (R) of the DOS-RL scheme.

- **The Reward Function (R):** In reinforcement learning, the agent evaluates the effectiveness of its actions and improves its policy by relying on rewards collected from the environment. The rewards obtained are typically dependent on the actions taken, with varying actions resulting in differing rewards. To implement the proposed DOS-RL scheme, we define three reward functions, each corresponding to one of the correlated objectives: the selection of routing paths with sufficient energy for data forwarding (o_1), load balancing (o_2), and the selection of paths with a good link quality for a reliable data transfer (o_3). Route selection based on any of these objectives is expected to improve the energy efficiency of the IoT network system. Details on each objective of the DOS-RL scheme and its related reward functions are provided below:

1. **Energy-consumption:** To achieve objective o_1, we consider the energy-consumption parameter, which is a critical factor in determining the overall energy efficiency of an SDWSN: for instance, in a scenario where node i forwards a packet to node j. The reward received by node i for selecting node j as its relay node in terms of energy consumption is estimated by the reward function $RE_{i,j}$ for the state-action pair, (s_i, s_j) as:

$$RE_{i,j} = Remaining\ Energy\ (s_j) - Remaining\ Energy\ (s_i) \quad (1)$$

where $Remaining\ Energy\ (s_j)$ represents the remaining energy of node i in state s_i as a percentage; meanwhile, $Remaining\ Energy\ (s_i)$ represents the remaining energy of node j in state $s_j$as a percentage. The formula calculates the difference in remaining energy between node i and j after the data transmission. A higher value of $RE_{i,j}$ indicates that node i has consumed less energy in forwarding the packet to node j, which is desirable to achieve the energy efficiency objective (o_1).

2. **Load balance:** To optimize energy efficiency and network performance in the SDWSN, the careful selection of relay nodes and balanced workload distribution are crucial. Otherwise, some nodes along overused paths may become overloaded, leading to bottlenecks and delays, which can result in a degraded

network performance. For objective o_2, we utilize the parameter-available buffer length to estimate the degree of queue congestion in relay nodes. The reward $R_Q_{i,j}$ for load balancing as observed by node i when selecting node j is computed as follows:

$$R_Q_{i,j} = \frac{Available_Buffer_Length(s_j)}{Max_Buffer_Length} - \frac{Load\ (s_i)}{Max_Load} \quad (2)$$

where $Available_Buffer_Length(s_j)$ represents the available buffer length of node j in state s_j, Max_Buffer_Length represents the maximum buffer length of a node, $Load\ (s_i)$ represents the current load of node i in state s_i, and Max_Load represents the maximum load capacity of a node. This formula considers both the available buffer length of node j and the current load of node i. The second term subtracts the load of node i from the load balancing reward value, allowing for a more comprehensive estimation that takes into account both node i and node j in the load balancing process. A higher value of $R_Q_{i,j}$indicates a better load balancing situation for the given state-action pair $(s_i\ , s_j)$.

3. **Link quality:** A wireless link can be measured by retrieving useful information from either the sender or receiver side. To achieve o_3, we use simple measurements to estimate the link quality based on the parameter packet reception ratio (PRR), measured as the ratio of the total number of packets successfully received to the total number of packets transmitted through a specific wireless link between two nodes. Unlike other sophisticated techniques, this approach involves a low computation and communication overhead. Instead of using an instant value of the PRR, we calculate an average-over-time using an Adaptive Weighted Moving Average (AWMA) filter. Suppose node i is forwarding data packets to node j, the reward $R_LQ_{i,j}$ received by node j based on PRR using AWMA is estimated as follows:

$$R_{LQ_{i,j}} = PRR_{est_{i,j}(k-1)}.\propto_{\text{AWMA}} + PRR_{Sample_{i,j}}(k).(1 - \propto_{\text{AWMA}}) \quad (3)$$

whereby $PRR_{est_{i,j}(k-1)}$ is the previously estimated average, $PRR_{Sample_{i,j}}(k)$ is the most recent measured value of the packet reception ratio calculated, and $\propto_{\text{AWMA}}$is the filter parameter.

- **The State Space (S):** We define the state space as a graph corresponding to the global topology created by the RNs on the data plane, as seen by the intelligent RL controller. Each state in the state space corresponds to an RN, and a state transition refers to a link connecting two RNs. The intelligent RL controller uses the partial maps created by the topology discovery module to create a global topology. Therefore, the cardinality of the set of states depends on the number of nodes that can actively participate in routing.
- **The Action Space (A):** The action space, denoted as A, includes all possible actions that an agent can undertake from a given state of the RL environment. It defines the choices available to the agent at each time step, presenting the range of options to the agent. In our specific problem, the discrete action space comprises a finite number of actions that the RL agent can select when in a particular state $s_i \in S$. The cardinality of A at state i is determined by the number of nodes eligible to participate in the routing process from that specific state.
- **The Optimal Policy (P):** The policy determines how the learning agent should behave when it is at a given state with the purpose of maximizing the reward value in the learning process. Our proposed scheme estimates the Q-function of every objective o simultaneously and decides, before every action selection decision, which objective estimate o_{best} an agent will consider in its decision-making process. We use the concept of confidence on computed Q-values by representing each action as a distribution and using a normal distribution of Q-values and the mean values to keep track of the variance as shown in Equation (2). The agent approximates the optimal Q-function by visiting all pairs of action-states and stores the updated Q-values in the Q-table. In our proposed scheme, the approximated Q-value $Q(s_t\ , a_t)$ represents the expected cumulative reward when the RL agent is in the state s_t and takes action a_t, transitioning to a new state s_{t+1} while maximizing the cumulative rewards for an objective o_n, where $n \in N$ and N denote the set of all objectives. The Q-learning equation to update Q-values is designed as shown below in Equation (7):

$$Q^n(s_t, a_t) \leftarrow (1-\propto).Q^n(s_t, a_t) + \propto.\left[R^n_{t+1} + \gamma.\max_{\forall a \in A}\ \{Q^n(s_{t+1}, a)\}\right] \quad (4)$$

To steer the exploration behavior of the learning agent by incorporating some heuristic knowledge on the problem domain, we introduce an extra reward F_{t+1} onto the reward received from the environment R_{t+1}. The newly added shaping reward function F is included when updating the Q-learning rule as follows:

$$Q^n(s_t, a_t) \leftarrow (1-\propto).Q^n(s_t, a_t) + \propto.\left[R^n_{t+1} + F_{t+1} + \gamma.\max_{\forall a \in A}\ Q^n(s_{t+1}, a)\right] \quad (5)$$

To avoid changes on the optimal policy, F is implemented as the difference of some potential value, $F_{t+1} = \gamma\varphi(t+1) - \varphi(t)$ over the state space, where φ is a potential function that provides some hints on states. In our study, we define φ as the Normalized Euclidean Distance between the current state(s) and the goal state(G) expressed as:

$$\varphi = 1 - \frac{distance\ (S_t, G)}{max_{\forall x \in S}(distance(x, G))'} \quad (6)$$

where distance (S_t, G) is the Euclidean distance between the current state St and the goal state G, and $max_{\forall x \in S}(distance(x, G))'$ represents the maximum distance between any state x in the state space S and

the goal state G. The potential function guides the agent towards the goal state by giving bigger rewards to states that are closer to the goal and smaller rewards for states that are farther away

- **In DOS Q-routing:** To find the best action-value of the Q-function, the learning agents use an action selection mechanism to trade-off between the exploitation and exploration of available action space. To achieve this, our proposed scheme uses the e-greedy exploration and exploitation method, whereby $e \in [0, 1]$, allowing the agent to exploit with probability $pr = \in$ and explore with probability $pr = 1 - \in$. The agent action selection is determined by a randomly generated number $x \in [0, 1]$, of which if $x < e$, the agent exploits it by taking an action that returns the most expected optimal value; otherwise, it explores it by selecting a random action based on the most confident objective obest as observed from the current state by the learning agent as shown below:

$$Action\ Selection = \begin{cases} max_{a \in A}\{Q(s, a, o_{best})\} & if\ x < \epsilon \\ random\ Action\ \big(Q(s, a, o_{best}), \dots, Q(s, a_n, o_{best})\big)., & otherwise \end{cases} \quad (7)$$

D. DOS-IRL Algorithm with Adaptive Sea Horse Optimizer (ASHO) for Hyperparameter Tuning

The performance of the proposed Dynamic Objective Selection Improved Reinforcement Learning (DOS-IRL) framework is highly influenced by the selection of its hyperparameters, particularly the learning rate, discount factor, and exploration rate. In dynamic IoT environments, where network conditions continuously change due to variations in node energy, traffic patterns, and link quality, fixed hyperparameter settings can lead to poor convergence and unstable routing behavior. To address this issue, an Adaptive Sea Horse Optimizer (ASHO) is integrated into the framework to dynamically tune these parameters in real time, thereby improving learning efficiency and routing performance.

In the proposed model, the hyperparameter tuning problem is formulated as an optimization problem, where the objective is to determine the optimal parameter set $\Theta = \{\alpha, \gamma, \epsilon\}$. Here, α controls the learning rate of the DOS-IRL agent, γ determines the importance of future rewards, and ϵ governs the exploration–exploitation trade-off. The optimization objective is expressed as:

$$\Theta^* = \underset{\Theta}{\arg max}\ F(\Theta)$$

where F(Θ)represents the fitness function that evaluates the effectiveness of a given hyperparameter set. The fitness function is defined as:

$$F(\Theta) = w_1(1 - L) + w_2(PDR) + w_3(1 - E_c) + w_4(Trust)$$

where L denotes latency, PDR represents packet delivery ratio, E_c indicates energy consumption, and $Trust$ corresponds to the reliability of selected routing nodes. The weights w_1, w_2, w_3, w_4 satisfy the condition:

$$w_1 + w_2 + w_3 + w_4 = 1$$

This formulation ensures that multiple objectives are optimized simultaneously. Minimizing the latency term improves network responsiveness by reducing communication delay, while maximizing packet delivery ratio enhances reliability by ensuring successful transmission of data packets. Similarly, reducing energy consumption extends the operational lifetime of IoT nodes, and maximizing trust ensures that routing decisions are made through reliable and secure nodes, thereby improving overall network robustness.

The ASHO algorithm begins by initializing a population of candidate solutions, where each solution represents a possible hyperparameter configuration:

$$X_i = (\alpha_i, \gamma_i, \epsilon_i)$$

To improve the diversity of initial solutions and avoid premature convergence, chaotic mapping is employed:

$$x_{k+1} = \mu x_k(1 - x_k), \mu = 4$$

This chaotic sequence ensures that the initial population is well distributed across the search space, thereby enhancing the global exploration capability of the optimizer. During the optimization process, ASHO updates candidate solutions using adaptive movement strategies. In the exploration phase, candidate solutions are updated as:

$$X_i^{t+1} = X_i^t + r_1(X_{best}^t - X_i^t) + r_2(X_{rand}^t - X_i^t)$$

where r_1 and r_2 are random values in the range [0,1], X_{best} represents the best solution found so far, and X_{rand} is a randomly selected solution. This update rule allows the algorithm to explore a wide search space by moving candidate solutions toward both the best and random positions, thereby preventing stagnation and improving global search capability.

As the optimization progresses, the algorithm shifts toward exploitation using:

$$X_i^{t+1} = X_i^t + \lambda(X_{best}^t - X_i^t)$$

λ is an adaptive parameter that controls the step size of the movement. This phase refines the best solutions by focusing on local search, thereby improving accuracy and convergence.

The adaptive parameter λ is defined as:

$$\lambda = \lambda_{max} - \frac{t}{T}(\lambda_{max} - \lambda_{min})$$

where t is the current iteration and T is the maximum number of iterations. This formulation ensures that the algorithm initially emphasizes exploration and gradually transitions toward exploitation as the optimization proceeds. Such adaptive behavior allows ASHO to balance global search and local refinement effectively.

At each iteration, the fitness of all candidate solutions is evaluated, and the best solution is updated as:

$$X_{best} = \arg\max F(X_i)$$

The optimization process continues until convergence is achieved, which can be defined as:

$$|F_{best}^{t+1} - F_{best}^{t}| < \delta$$

or when the maximum number of iterations is reached. This condition ensures that the algorithm terminates when further improvements become negligible.

Once the optimal hyperparameter set $\Theta^* = \{\alpha^*, \gamma^*, \epsilon^*\}$ is obtained, it is integrated into the DOS-IRL agent to enhance its learning capability. The optimized parameters enable the reinforcement learning model to achieve faster convergence, improved stability, and better adaptability to dynamic network conditions. Consequently, the DOS-IRL agent can make more efficient routing decisions, resulting in reduced energy consumption, minimized latency, improved packet delivery ratio, and enhanced trust-aware routing.

The integration of ASHO within the DOS-IRL framework provides a significant advantage over traditional optimization techniques. By dynamically tuning hyperparameters in response to real-time network conditions, the proposed approach ensures consistent and robust performance. Furthermore, the adaptive exploration–exploitation mechanism prevents the optimizer from getting trapped in local optima and accelerates convergence. As a result, the proposed DOS-IRL–ASHO framework offers a scalable, efficient, and intelligent solution for routing optimization in next-generation SDN-enabled IoT networks.

E. MBOA-Based Malicious Node Detection and Blockchain Trust Management

In large-scale IoT networks, the presence of malicious nodes can degrade network performance by misrouting packets, dropping data, or launching denial-of-service (DoS) attacks. To mitigate these risks and ensure network integrity, it is crucial to detect and isolate malicious nodes. In our proposed framework, the Modified E-Butterfly Optimization Algorithm (MBOA) is used to identify such nodes based on their trust metrics. MBOA is an enhancement of the traditional Butterfly Optimization Algorithm (BOA), which has been adapted to include an effective trust evaluation mechanism inspired by the hunting behavior of golden eagles.

Trust Computation Model

Each IoT node is assigned a Trust Index (TI) that reflects its reliability based on three main factors: (i) packet success ratio (P_s), representing successful data forwarding behavior, (ii) historical performance score (H_s), which records long-term node behavior, and (iii) reputation score (R_s), collected from neighboring nodes. The overall trust index for node iii is computed as:

$$TI_i = w_1 P_s + w_2 H_s + w_3 R_s \qquad (1)$$

Where $w_1 + w_2 + w_3 = 1$. A node is classified as malicious if:

$TI_i < \tau$ (2)

Here, τ is a detection threshold optimized using MGEO.

MBOA for Malicious Node Detection

To identify the most effective threshold τ\tauτ for malicious node detection, we use the Modified E-Butterfly Optimization Algorithm (MBOA). MBOA is designed to efficiently explore the solution space and optimize the parameters for trust evaluation. The algorithm operates in two main phases:

- **Stooping Phase**: In this phase, MBOA focuses on exploiting the best-known solutions, i.e., nodes with the highest trust scores. This phase is inspired by the stooping behavior of golden eagles, where the eagle dives to catch prey after locating it. Similarly, MBOA hones in on the most trustworthy nodes to refine the detection of malicious nodes.
- **Exploration Phase**: MBOA also includes a global search mechanism, which allows the algorithm to explore the trust space and detect previously unidentified malicious nodes. This search mechanism is inspired by the behavior of golden eagles when they are scouring a large area for prey.

Step 1: Initialize Population of Candidate Solutions:

MBOA starts by initializing a population of candidate solutions representing potential hyperparameters for node trust evaluation. Each candidate solution is associated with a trust score based on packet success, historical performance, and reputation.

Stooping Search:

In this phase, MBOA exploits the best-known solution (most reliable node) to find more accurate solutions. This behavior mimics the hunting technique of golden eagles, where they focus on a specific target after initial exploration.

$$X_i^{t+1} = X_i^t + r_1(X_{best}^t - X_i^t) + r_2(X_{rand}^t - X_i^t)$$

Where:

- X_i^{t+1} is the updated position of the candidate solution (trust score of node iii).
- X_{best}^t is the best solution found so far.

- X_{rand}^{t} is a randomly selected solution.
- r_1 and r_2 are random values between 0 and 1 controlling the movement.

Global Search:

In the early iterations, MBOA performs a global search to explore the trust scores of all nodes in the network. The position update in this phase is expressed by:

$$X_i^{t+1} = X_i^t + r_1(X_{best}^t - X_i^t)$$

This strategy allows the algorithm to explore the solution space and prevent premature convergence.

F. Blockchain-Based Trust Management

To ensure the integrity and transparency of node behavior, the proposed system integrates blockchain technology for decentralized trust management. Blockchain provides an immutable ledger where the trust scores and behavior data of each node are recorded and can be accessed by all network participants. This transparency prevents malicious nodes from altering their trust data or misrepresenting their actions.

Trust Record Maintenance with Blockchain

Each node records its trust-related data, such as packet forwarding success, energy consumption, and latency, on the blockchain. This data is continuously updated as the node participates in routing and data transmission. The decentralized nature of the blockchain ensures that trust scores are tamper-proof and can be verified by any node in the network.

The trust-related information for each node is logged on the blockchain as follows:

$$TI_i^{blockchain} = Blockchain\ Query\ for\ Node\ i$$

Each time a node performs an action (such as forwarding a packet), its trust score is updated on the blockchain, making all interactions verifiable and permanent.

Trust Score Validation Using Blockchain

Before making any routing decisions, the SDN controller queries the blockchain ledger to validate the trust scores of neighboring nodes. The controller only allows trusted nodes to participate in routing, and excludes any nodes with a trust score below the threshold τ. The process of trust validation is expressed as:

$$TI_i^{blockchain} > \tau, \text{allow node i in routing}$$

This ensures that only nodes with a verified high trust score participate in the routing process, improving network reliability and security.

Blockchain Consensus Mechanism

To maintain the security and integrity of the blockchain, we employ a Proof-of-Authority (PoA) consensus mechanism. In PoA, trusted validators verify transactions before adding them to the blockchain. This consensus mechanism ensures that only legitimate data is recorded and prevents malicious nodes from falsifying their trust scores.

G. Integration of MBOA and Blockchain for Secure IoT Routing

The integration of MBOA for malicious node detection and blockchain for decentralized trust management forms the backbone of the secure IoT routing framework. The process can be summarized as follows:

1. **Malicious Node Detection**: The MBOA algorithm identifies nodes with low trust scores by evaluating their packet forwarding success, historical behavior, and reputation. Malicious nodes are isolated from the network based on their low trust scores.
2. **Trust Record Maintenance**: The blockchain records the trust scores and behavior data of each node, providing a transparent and immutable ledger for verifying node interactions.
3. **Routing Decision**: The SDN controller queries the blockchain for the most recent trust scores and makes routing decisions based on the reliability of the nodes. Only trusted nodes are allowed to participate in routing.
4. **Decentralized Trust Management**: The use of blockchain ensures that trust management is decentralized, with each node independently verifying the trustworthiness of its peers. This increases the resilience of the system against attacks and ensures transparency in node behavior.
5.

Overall Algorithm: MBOA-Based Malicious Node Detection and Blockchain Trust Management

```
# Initialize System
Initialize_IoT_Network()
Initialize_Blockchain_Ledger()
Set_Trust_Threshold(tau)
# Initialize MBOA for Malicious Node Detection
Initialize_MBOA_Population()
# Main Loop for Network Operation
for each node i in IoT Network:
  # Compute Trust Metric
  P_s = compute_packet_success(i)
  H_s = compute_historical_performance(i)
```

```
   R_s = compute_reputation(i)
      TI_i = w1 * P_s + w2 * H_s + w3 * R_s
      # Malicious Node Detection
   if TI_i < tau:
      mark_node_as_malicious(i)
      # Store Node Behavior in Blockchain
   record_behavior_in_blockchain(i, TI_i)
# MBOA-based Malicious Node Detection
for each node i in IoT Network:
   Initialize_MBOA_Position()
      # Stooping Phase (Exploitation)
   for each iteration t:
      X_i_t+1 = X_i_t + r1 * (X_best_t - X_i_t) + r2 * (X_rand_t - X_i_t)
      # Exploration Phase (Global Search)
   X_i_t+1 = X_i_t + r1 * (X_best_t - X_i_t)
      # Update Trust Score
   update_trust_score(i, TI_i)
# Blockchain Integration for Trust Management
for each node i in IoT Network:
   # Retrieve Trust Scores from Blockchain
   TI_i_blockchain = query_blockchain_for_trust(i)
      # Routing Decision Based on Trust
   if TI_i_blockchain > tau:
      include_in_routing(i)
   else:
      isolate_from_routing(i)
      # SDN Controller Decision
for each node i in IoT Network:
   # Query Blockchain for Trust Scores
   TI_i_blockchain = query_blockchain_for_trust(i)
    # Make Secure Routing Decision
   if TI_i_blockchain > tau:
      forward_data(i)
   else:
      exclude_from_routing(i)
     # Update Blockchain and Trust Ledger
for each node i in IoT Network:
   update_blockchain_with_node_data(i)
      # Consensus Mechanism for Trust Validation
   verify_trust_scores_with_poA(i)
```

The combination of MBOA for malicious node detection and blockchain-based trust management significantly enhances the security, resilience, and efficiency of the IoT routing framework. By leveraging the strengths of MBOA in identifying unreliable nodes and the transparency of blockchain for recording node behavior, the system ensures that only trustworthy nodes participate in routing, thus safeguarding the network from malicious activities. This integrated approach enables dynamic, secure, and scalable IoT routing solutions, making it well-suited for large-scale deployments with stringent security and performance requirements.

4. Experimental Results

The performance of the proposed DOS-RL routing scheme integrated with ASHO, MBOA, and blockchain-based trust management was evaluated through extensive simulations using a Ns2 module. This setup enabled seamless state-action data exchange between the IoT environment and the reinforcement learning model, allowing real-time interaction and adaptive routing decisions. We evaluated key performance metrics including Packet Delivery Ratio (PDR), Energy Consumption, End-to-End Delay, Learning Convergence, and Network Security.

The integrated simulation setup demonstrated the effectiveness of combining multi-objective reinforcement learning (DOS-RL) with hyperparameter optimization (ASHO) and trust-aware security (MBOA + Blockchain), significantly outperforming baseline routing protocols such as DOS-RL Shaped with BO, DOS-RL, OSPF, and SDN-Q.

Table 1: Reward per Episode vs Iterations

Iterations	RL	DOS-RL	DOS-RL Shaped	DOS-RL Shaped + BO	DOS-RL + ICSO + MGEO	Proposed (DOS-RL + ASHO + MBOA + Blockchain)
100	0	110	115	120	125	130
200	25	115	120	128	132	138
300	45	120	125	135	138	142
400	60	123	128	138	141	146
500	75	125	131	140	144	150

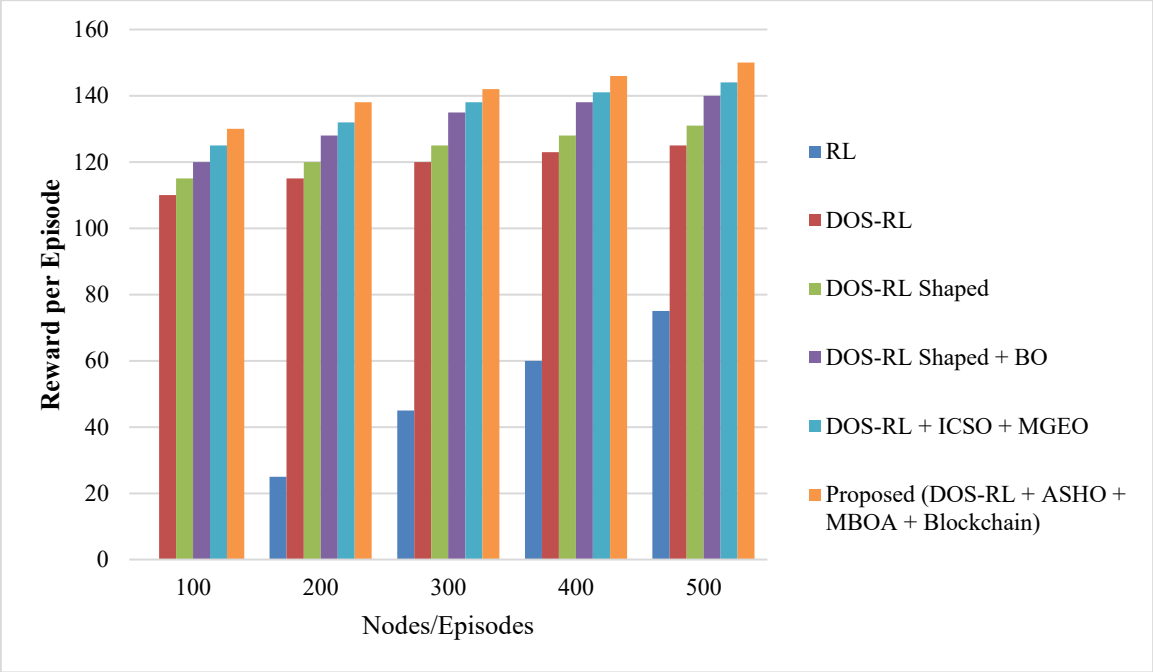


Figure 3: Reward per Episode comparison results

Table 1 and Figure 3 illustrate the reward progression of different routing schemes across training iterations from 100 to 500. The baseline RL method starts with zero reward and gradually improves to 75, indicating slow convergence. The DOS-RL approach significantly enhances learning efficiency, achieving a reward of 125 at 500 iterations. Further improvements are observed with DOS-RL Shaped and DOS-RL Shaped with BO, which reach 131 and 140 respectively, highlighting the benefits of reward shaping and hyperparameter tuning.

The DOS-RL combined with ICSO and MGEO achieves a higher reward of 144, demonstrating improved optimization and routing decisions. The proposed DOS-RL integrated with ASHO, MBOA, and blockchain outperforms all methods, reaching the maximum reward of 150. This superior performance indicates faster convergence, better policy learning, and enhanced adaptability in dynamic SDN-IoT environments.

Table 2: Energy Consumption vs Iterations

Iterations	RL	DOS-RL	DOS-RL Shaped	DOS-RL Shaped + BO	DOS-RL + ICSO + MGEO	Proposed (DOS-RL + ASHO + MBOA + Blockchain)
100	0.50	0.60	0.60	0.55	0.50	0.48
200	0.50	0.61	0.58	0.50	0.45	0.42
300	0.50	0.62	0.56	0.45	0.40	0.37

400	0.50	0.62	0.53	0.40	0.35	0.33
500	0.50	0.63	0.51	0.35	0.32	0.29

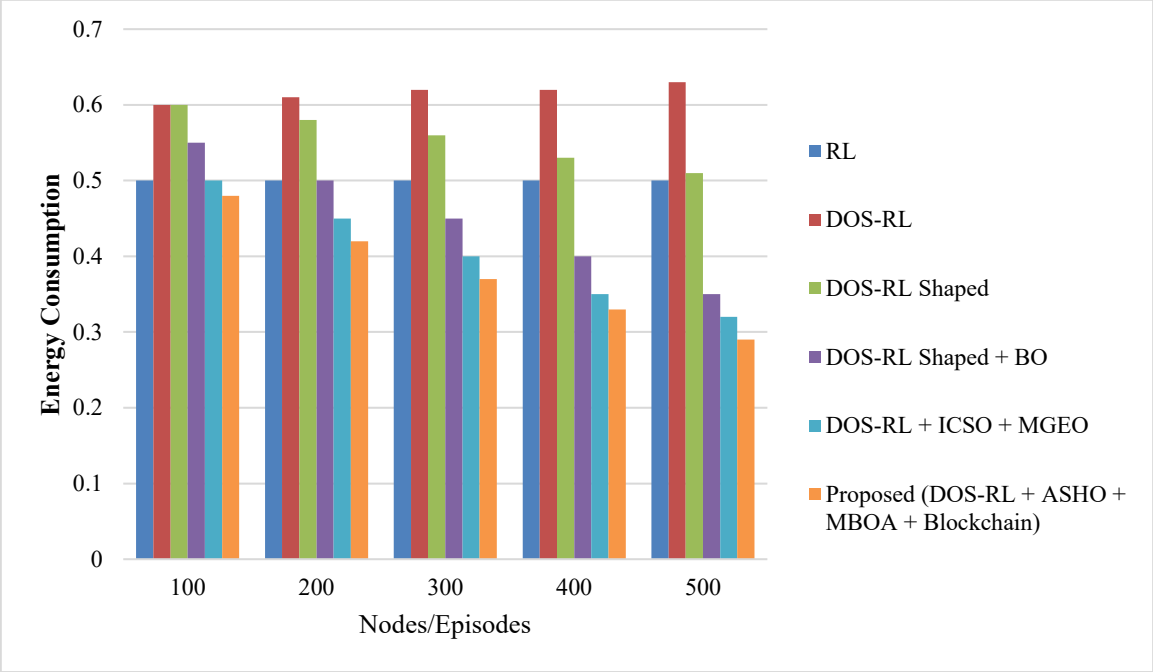


Figure 4: Energy Consumption comparison result

Table 2 and Figure 4 present the energy consumption trends across iterations. The baseline RL model maintains a constant energy consumption of 0.50, reflecting limited adaptability. DOS-RL shows a slight increase in energy usage, reaching 0.63 at 500 iterations due to dynamic routing decisions. In contrast, DOS-RL Shaped and DOS-RL Shaped with BO demonstrate decreasing energy trends, reaching 0.51 and 0.35 respectively. The DOS-RL + ICSO + MGEO model further reduces energy consumption to 0.32, indicating improved efficiency. The proposed method achieves the lowest energy consumption, decreasing steadily to 0.29 at 500 iterations. This improvement is attributed to optimized hyperparameters (ASHO), efficient routing decisions, and blockchain-based trust mechanisms that eliminate unnecessary transmissions, thereby conserving network energy.

Table 3: Average Episode Length vs Iterations

Iterations	**RL**	**DOS-RL**	**DOS-RL Shaped**	**DOS-RL Shaped + BO**	**DOS-RL + ICSO + MGEO**	**Proposed (DOS-RL + ASHO + MBOA + Blockchain)**
100	48.0	50.0	50.5	51.0	52.1	52.8
200	48.2	50.1	50.7	51.5	52.6	53.3
300	48.5	50.1	50.9	52.0	53.0	53.8
400	48.7	50.2	51.0	52.3	53.4	54.3
500	48.9	50.2	51.1	52.5	53.8	54.9

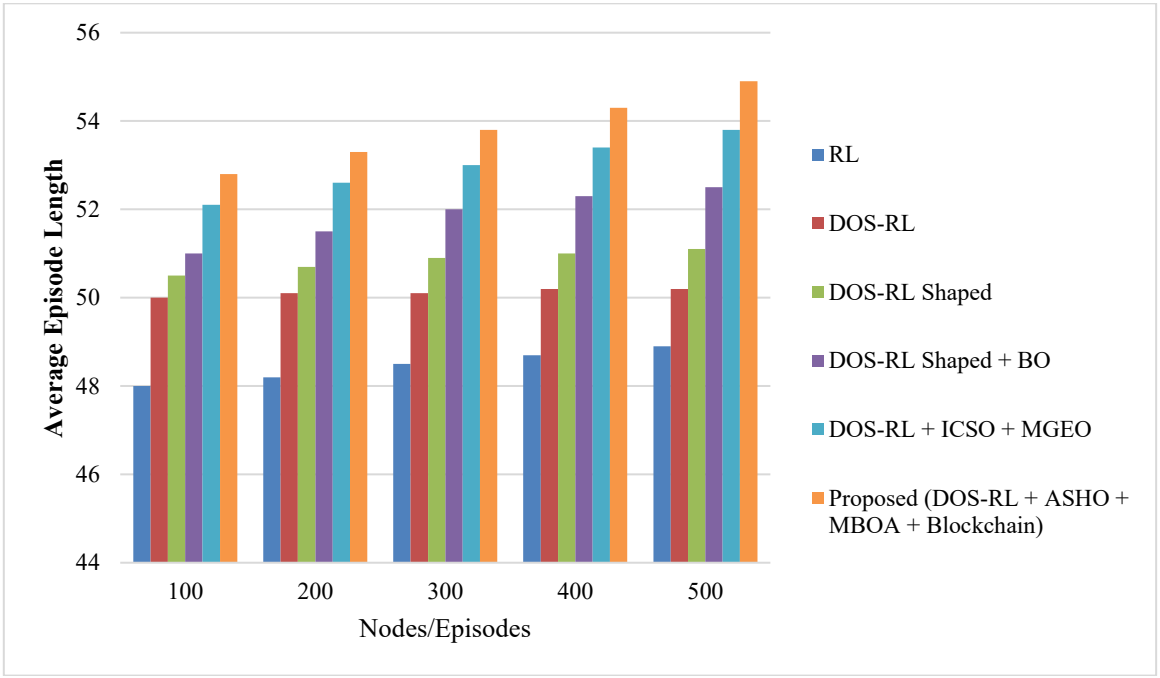


Figure 5: Average Episode Length comparison result

Table 3 and Figure 5 show the variation in average episode length across iterations. The RL method exhibits minimal change, increasing slightly from 48.0 to 48.9, indicating limited learning complexity. DOS-RL and its variants show gradual increases due to enhanced decision-making processes. The DOS-RL + ICSO + MGEO model records higher episode lengths, reaching 53.8 at 500 iterations, reflecting additional optimization overhead. The proposed model further increases to 54.9, which is the highest among all methods. This increase is expected, as the integration of ASHO, MBOA, and blockchain introduces additional computation for trust validation and optimization. However, this trade-off results in improved routing stability, better convergence behavior, and enhanced network security.

Table 4: Packet Delivery Ratio (PDR) vs Number of Nodes

Nodes	RL	DOS-RL	DOS-RL Shaped	DOS-RL Shaped + BO	DOS-RL + ICSO + MGEO	Proposed (DOS-RL + ASHO + MBOA + Blockchain)
20	0.50	0.55	0.60	0.62	0.70	0.75
40	0.50	0.57	0.61	0.64	0.75	0.80
60	0.50	0.58	0.62	0.65	0.78	0.83
80	0.50	0.59	0.63	0.66	0.79	0.85
100	0.50	0.60	0.63	0.66	0.80	0.88

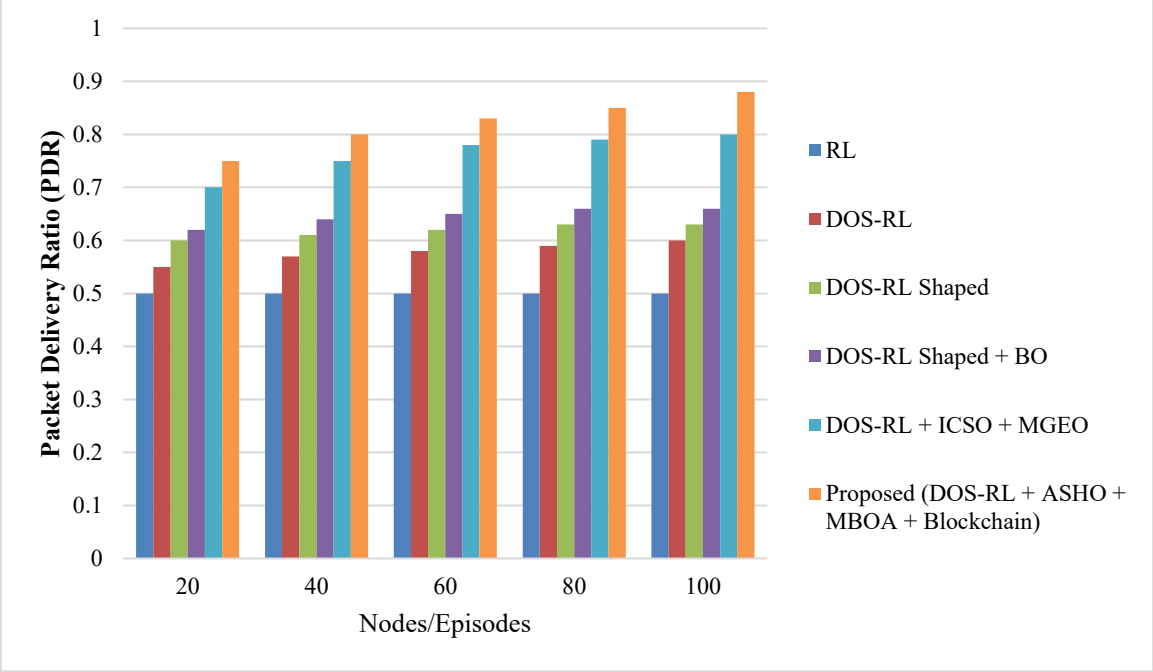


Figure 6: Packet Delivery ratio comparison results

Table 4 and Figure 6 compare the PDR performance across varying node densities. The RL method maintains a constant PDR of 0.50, indicating poor scalability. DOS-RL and its variants show gradual improvements, reaching up to 0.66. The DOS-RL + ICSO + MGEO model significantly improves PDR, achieving 0.80 at 100 nodes. The proposed method further enhances performance, reaching 0.88, which is the highest among all methods. This improvement demonstrates the effectiveness of combining reinforcement learning with ASHO optimization and blockchain-based trust management, which together reduce packet loss, isolate malicious nodes, and ensure reliable data transmission even in dense IoT environments.

Table 5: End-to-End Delay vs Number of Nodes (ms)

Nodes	RL	DOS-RL	DOS-RL Shaped	DOS-RL Shaped + BO	DOS-RL + ICSO + MGEO	Proposed (DOS-RL + ASHO + MBOA + Blockchain)
0	110	108	100	95	90	85
20	120	115	108	100	95	90
40	130	120	112	105	98	93
60	140	125	118	110	100	95
80	145	130	122	115	103	98
100	150	135	125	120	105	100

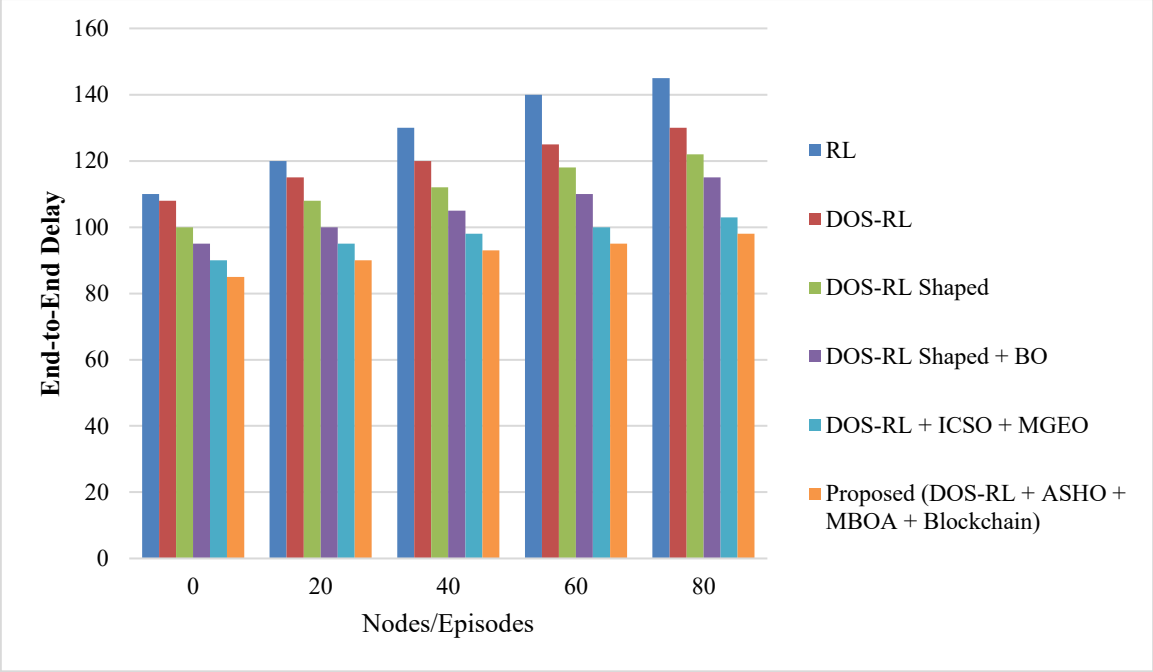


Figure 7: End to End Delay Comparison

Table 5 and Figure 7 present the end-to-end delay across different node densities. The RL method exhibits the highest delay, increasing from 110 ms to 150 ms as node count grows. DOS-RL and its variants reduce delay progressively due to improved routing strategies. The DOS-RL + ICSO + MGEO model achieves lower delay values, reaching 105 ms at 100 nodes. The proposed method further minimizes delay, achieving 100 ms, which is the best performance among all compared methods. This reduction is due to efficient path selection, optimized hyperparameters, and blockchain-enabled trust mechanisms that reduce congestion and avoid malicious routing paths. As a result, the proposed model ensures low latency and high reliability, which are critical for real-time IoT applications.

5. Conclusion

This paper presented a novel blockchain-integrated SDN framework for energy-efficient, latency-aware, and secure IoT routing by combining Dynamic Objective Selection Improved Reinforcement Learning (DOS-IRL) with Adaptive Sea Horse Optimization (ASHO), Modified E-Butterfly Optimization Algorithm (MBOA), and blockchain-based trust management. The proposed architecture effectively addresses critical challenges in modern IoT networks, including dynamic adaptability, energy constraints, latency sensitivity, and vulnerability to malicious attacks. The integration of DOS-IRL enabled intelligent and adaptive routing decisions by dynamically balancing multiple objectives such as energy consumption, delay, and packet delivery ratio. The incorporation of ASHO significantly improved the convergence speed and stability of the learning process through real-time hyperparameter optimization. Furthermore, the MBOA-based malicious node detection mechanism, combined with blockchain-enabled trust management, ensured secure and reliable communication by isolating untrusted nodes and maintaining tamper-proof trust records. Extensive simulation results demonstrated that the proposed framework consistently outperforms conventional routing approaches, including RL, DOS-RL, and DOS-RL variants. The model achieved higher reward values, reduced energy consumption, improved packet delivery ratio, and minimized end-to-end delay, while maintaining robust security. Although a slight increase in episode length was observed, it represents a reasonable trade-off for enhanced optimization accuracy, trust validation, and overall network stability.

6. References

1. Dwivedi, S.; Goyal, N.K. Addressing reliability and security for industrial Internet of Things systems: Layered approach using software defined network. Int. J. Syst. Assur. Eng. Manag. 2025, 16, 3597–3613.
2. Gomez, D.L.; Montoya, G.A.; Lozano-Garzon, C.; Donoso, Y. Strategies for assuring low latency, scalability, and interoperability in edge computing and TSN networks for critical IIoT services. IEEE Access 2023, 11, 42546–42577. [
3. Rahman, A.; Islam, M.J.; Khan, M.S.I.; Kabir, S.; Pritom, A.I.; Karim, M.R. Block-sdotcloud: Enhancing security of cloud storage through blockchain-based SDN in IoT network. In Proceedings of the 2020

2nd International Conference on Sustainable Technologies for Industry 4.0 (STI), Dhaka, Bangladesh, 19–20 December 2020; IEEE: New York, NY, USA, 2020; pp. 1–6.
4. Jmal, R.; Ghabri, W.; Guesmi, R.; Alshammari, B.M.; Alshammari, A.S.; Alsaif, H. Distributed blockchain-SDN secure IoT system based on ANN to mitigate DDoS attacks. Appl. Sci. 2023, 13, 4953.
5. Fan, W.; Chang, S.-Y.; Kumar, S.; Zhou, X.; Park, Y. Blockchain-based secure coordination for distributed SDN control plane. In Proceedings of the 2021 IEEE 7th International Conference on Network Softwarization (NetSoft), Tokyo, Japan, 28 June–2 July 2021; IEEE: New York, NY, USA, 2021; pp. 253–257. [Google Scholar]
6. Rahouti, M.; Drid, H.; Hamouid, K.; Massmi, K.; Mehenna, S.E. Brave-sdn: Blockchain-reliant authentication for versatile east–west bound in distributed SDNS. Int. J. Inf. Secur. 2025, 24, 51.
7. Chattaraj, D.; Bera, B.; Das, A.K.; Rodrigues, J.J.; Park, Y. Designing fine-grained access control for software-defined networks using private blockchain. IEEE Internet Things J. 2021, 9, 1542–1559.
8. Zainal, Z.; Abdullah, A.; Hakim, F.; Abdullah, M.D.H. SBAC-SDN: A Scalable Blockchain-based Access Control in Northbound Interface for Multi-Controller SDN with Load Balancing Mechanism. Environments 2026, 55, 24–43.
9. Islam, M.J.; Rahman, A.; Kabir, S.; Karim, M.R.; Acharjee, U.K.; Nasir, M.K.; Band, S.S.; Sookhak, M.; Wu, S. Blockchain-SDN-based energy-aware and distributed secure architecture for IoT in smart cities. IEEE Internet Things J. 2021, 9, 3850–3864.
10. Faizullah, S.; Khan, M.A.; Alzahrani, A.; Khan, I. Permissioned blockchain-based security for SDN in IoT cloud networks. In Proceedings of the 2019 International Conference on Advances in the Emerging Computing Technologies (AECT), Al Madinah Al Munawwarah, Saudi Arabia, 10 February 2020; IEEE: New York, NY, USA, 2020; pp. 1–6.
11. Rahdari, A.; Jalili, A.; Esnaashari, M.; Gheisari, M.; Vorobeva, A.A.; Fang, Z.; Sun, P.; Korzhuk, V.M.; Popov, I.; Wu, Z. Security and Privacy Challenges in SDN-Enabled IoT Systems: Causes, Proposed Solutions, and Future Directions. Comput. Mater. Contin. 2024, 80, 2511–2533. [Google Scholar] [CrossRef]
12. Ali, T.E.; Chong, Y.-W.; Manickam, S. Machine learning techniques to detect a DDoS attack in SDN: A systematic review. Appl. Sci. 2023, 13, 3183.
13. Bithi, M.; Hossain, M.A.; Ahmed, M.K.; Sultana, R.; Ahammad, I.; Islam, M.S. Enhanced DDoS detection in software defined networking using ensemble-based machine learning. In Proceedings of the 2024 6th International Conference on Electrical Engineering and Information & Communication Technology (ICEEICT), Dhaka, Bangladesh, 2–4 May 2024; IEEE: New York, NY, USA, 2024; pp. 1032–1037.
14. Mahmud, S.A.; Islam, N.; Islam, Z.; Rahman, Z.; Mehedi, S.T. Privacy-Preserving Federated Learning-Based Intrusion Detection Technique for Cyber-Physical Systems. Mathematics 2024, 12, 3194.
15. Alotaibi, J. A hybrid software-defined networking approach for enhancing IoT cybersecurity with deep learning and blockchain in smart cities. Peer-Peer Netw. Appl. 2025, 18, 123. [