International Journal of Safety and Security Engineering
Vol. 15, No. 8, August, 2025, pp. 1745-1754
Journal homepage: http://iieta.org/journals/ijsse

Deep Reinforcement Learning-Based Energy-Aware Intrusion Prevention in IoT Environment

K. S. Rekha[1], Priyadarshini Jainapur[2], K. Manjushree[3], Shashank Dhananjaya[4], S. R. Nandini[5*], G. Nandini[6], R. Sunitha[7]

[1] Department of Computer Science and Engineering, JSS Science and Technology University, Mysuru 570006, India
[2] Department of Electronics and Communication, BMS College of Engineering, Bangalore 560019, India
[3] Department of Computer Science and Engineering, BNM Institute of Technology, Bangalore 560070, India
[4] Department of Information Science and Engineering, The National Institute of Engineering, Mysuru 570008, India
[5] Department of Computer Science and Engineering, Faculty of Engineering and Management, BGSIT, Adichunchanagiri University, Bellur 571448, India
[6] Department of Information Science and Engineering, BNM Institute of Technology, Bangalore 560070, India
[7] Department of Artificial Intelligence and Machine Learning, BNM Institute of Technology, Bangalore 560070, India

Corresponding Author Email: nandinisr593@gmail.com

https://doi.org/10.18280/ijsse.150819

Received: 18 July 2025
Revised: 15 August 2025
Accepted: 19 August 2025
Available online: 31 August 2025

Keywords:
deep learning, intrusion prevention, IoT, energy-aware IDS, deep Q-network

ABSTRACT

The quick growth of the Internet of Things (IoT) has produced severe security issues because of sensor node diversity, scale of deployment, and power limitations. Intrusion detection systems (IDS) are often not flexible and can consume significant energy or computing resources, and are usually not suited for real-time protection in a resource-limited environment. The paper proposes a deep reinforcement learning-based energy-aware intrusion prevention system (DRL-EAIPS). The proposed system is a novel framework to integrate a lightweight quantized convolutional neural network (CNN) module for fast and energy-efficient anomaly detection at the node level, with a deep Q-network (DQN)-based agent that learns to make intrusion prevention decisions that are dynamically decided to learn the whole IoT environment. Thus, the system can extract useful features with minimal computation overhead, but also improves intrusion prevention in an adaptive and energy-aware way using reinforcement learning (RL), which stresses both detection accuracy and energy usage. The reward function is designed with a recognized multi-objective approach, aiming to optimize detection rates while minimizing energy usage, which is crucial to deploy in resource-constrained environments such as IoT systems practically. The extensive simulations done with MATLAB and NS-3 using datasets of NSL-KDD, BoT-IoT, and UNSW-NB15 demonstrated that the DRL-EAIPS outperformed existing methodologies with a good accuracy of 97.21%, a low false positive rate (FPR) of 2.65%, and low energy consumption. Additionally, the routable model expands the network lifetime and lowers latency and transmission expenditure compared to existing DQN-based, CNN-GRU, and trust-aware IDS models. The results demonstrated the scalability, resilience, and real-time utility of the DRL-EAIPS in current IoT paradigms.

1. INTRODUCTION

The Internet of Things (IoT) keeps growing fast, and billions of smart sensors and devices now connect worldwide. Experts think we'll have about 75 billion IoT devices by 2025 [1]. This quick growth makes it easier for cybercriminals to attack. IoT devices often work alone in tough spots, so they can fall victim to many kinds of attacks. Some common threats are denial-of-service (DoS) attacks, spoofing, jamming, eavesdropping, data manipulation, and man-in-the-middle tricks [2]. Also, IoT systems mix many different devices that don't have much CPU, memory, or energy [3]. Their small batteries and weak processing power mean that normal security fixes made for strong servers just don't work well [4].

Making sure IoT security is strong is a big challenge. The usual intrusion detection systems (IDS)/intrusion prevention systems (IPS) systems that use rules or signatures have problems with too many false alarms and can't adapt to fit the always-changing big IoT world. People have suggested using machine learning (ML) and deep learning (DL) a lot to make IoT security better by learning attack patterns on their own [5]. IDS with ML added can spot weird traffic without needing exact signatures. New studies show that smart IDS using ML/DL can find unknown threats and work in real-time, which IoT networks need [6]. But ML/DL models can need a lot of computer power. In IoT devices with limited resources, the energy cost to run complex models is something to think about [7]. In fact, today's ML/DL often needs a lot of CPU

power and drains batteries fast [8].

Reinforcement learning (RL) has become a promising way to tackle these problems. By interacting with their surroundings, RL agents can make security decisions one after another (like when and how to check traffic) and adjust to shifting attack patterns [9]. Deep RL (which combines neural networks with RL) can deal with complex network states and figure out intricate defense strategies. Earlier studies show that RL-based IDS can adapt to new threats and work within resource limits [10]. For example, previous researchers demonstrated that a deep deterministic policy gradient (DDPG) agent can trade off intrusion detection accuracy against devices' energy use, achieving fairly effective intrusion detection but with minimal battery energy consumption. However, while deep RL helps with detection, IoT security requires consideration of energy-aware intrusion prevention [11]. In other words, the device's security defenses must detect IoT intrusions accurately, while optimizing the devices' energy consumption to increase the lifetime of the network [12].

In spite of the steady advancements in the ML and DL-based IDS, there remain various important limitations that make the actual implementation of IDSs in IOT environments unfeasible. First, the vast majority of ML/DL modeling frameworks require extensive computational resources and continuous training on high-volume data and datasets, which is often not possible to meet in the low-power contexts of IoT devices. This results in a trade-off between either compromising the accuracy by using lightweight models or significantly increasing the energy consumption to enable complex models. Second, most traditional IDS approaches are primarily based on the accuracy of detection and either do not have much focus on energy consumption, which impacts the device lifetime, and therefore impacts the network longevity in the IOT domain. Third, most traditional approaches to intrusion detection do not dynamically adapt to the changing network environment and threat landscape, so they remain static and could lead to more false positives or slower responses. Finally, current intrusion detection and intrusion prevention systems differentiate between both and therefore miss out on the decision-making process on how to optimize security efficiency at the same time.

In this paper, we propose a deep RL framework for energy-aware intrusion prevention for IoT devices. We model the intrusion prevention problem as a Markov decision process (MDP), where the agent can take actions that will include a security check or countermeasure, and the reward function that penalizes energy use, packet passing delay, in addition to missed intrusion detections. Our mathematical model, described below, explicitly defines uses energy for monitoring and communication, allowing the RL agent to learn policies that optimize the security-energy tradeoff.

The contributions of this paper are threefold:

(1) A unique deep RL-based IDS designed specifically for energy-constrained IoT.

(2) A full mathematical formulation of our system states, actions, and reward showing energy level and network metrics.

(3) A simulation-based evaluation showing significant savings in energy consumption, network lifetime, and delay in comparison to benchmarks.

Using RL allows our method to compare to the state-of-the-art, not only to reduce energy-based security overhead on IoT nodes dynamically while still offering a robust threat prevention defense technique, but also fills a gap in the academic research space.

2. RELATED WORK

In recent years, there has been substantial research on IoT security that makes use of AI techniques. Many have pursued DL-based IDS (in an earlier stage). For example, Gyamfi and Jurcut [13] have approached the class imbalance in IoT IDS with a class-imbalance, focusing on focal loss in certain types of DL models in their training, and have shown promising and significant gains in precision and F1-score. Many other DL models, like convolutional neural networks (CNNs), LSTMs, and auto encoders, have been developed for anomaly detection in IoT traffic. And the results have reported high detection accuracies across lots of data sets. Transfer-learning was also explored by Lazzarini et al. [14], have proposed an IDS framework designed for 5G IoT based on transfer-learning principles to reuse knowledge in different domains, and also improve the detection of zero-day attacks. Traditional ML methods like SVM, random forest, and ensemble are still applied in IoT IDS, but deep and transfer models have fared better at wrapping up the complexity of attack patterns.

One of the ideas of Green AI is to decrease the computational costs related to ML models. For example, Deshmukh and Ravulakollu [15] have proposed an EnergyCIDN, which is essentially a collaborative IDS in which the authors integrated an energy-aware trust model to reduce the verification cost required to validate IoT nodes. Research has found considerable battery life savings with the addition of energy considerations: Tharewal et al. [16] have reported up to 35% energy savings when combining energy-aware design with RL, and a different project reported that they gained 21% more energy efficiency in IDS by dynamically scaling model complexity. Similarly, enhancing energy efficiency as a significant area of research, such as towards efficient deep models like model pruning, quantization, and knowledge distillation, are all learning aspects of how to reduce energy without impact on accuracy. Research into energy-efficient models has primarily focused on static ML models while changing the characteristics of the IDS in IoT environments.

RL has the prospect of adaptivity and is being looked at as a means of providing adaptive and flexible IDS. Given the adaptive nature of security in IoT contexts, RL agents will simply know when and where to implement security measures. For example, Taşcı [17] have developed a deep Q-network (DQN)-based IDS named DQN-HIDS for social IoT. The proposed guild of learning for identifying intrusions relies in part on LSTM-DQN that incrementally improves the correctness of labelled intrusions, with fewer samples needed. In the results, the DQN-HIDS shows a high level of classification accuracy with fewer training samples than purely supervised approaches. Recent studies in wireless networks and IoT-based networks have researched many possible approaches to assist with intrusion detection and to improve resilience to cyberattacks. AGR et al. [18] have addressed one component of wireless networks, the distributed denial-of-service (DDoS) flooding attack. They proposed a new mechanism that uses dynamic path identifiers for resilience by limiting the probability of exploitation of a single route, which led to improved detection and reduced false positives. While their strategy effectively countered flooding-based DDoS, it has limited scalability when extended to IoT

on a large scale. The dynamic paths they had to manage created overload, and their mechanism was insufficient to counter multi-vector or application-layer attacks.

Moving towards DL solutions, Yaras and Dener [19] have developed an IoT-based IDS with a hybrid DL model consisting of a combination of convolutional and recurrent neural networks. This system captured both spatial and temporal features of network traffic, improving accuracy against many different types of IoT attacks. However, the model's computational cost and complexity in training made it less appropriate for the resource-limited design of IoT nodes. Additionally, it relied on labeled datasets, which limited real-time adaptation to zero-day threats without huge retraining.

To tackle the data imbalance problem in intrusion detection, Dener et al. [20] have developed the STLGBM-DDS framework which was a combination of synthetic minority oversampling technique (SMOTE) and light gradient boosting machine (LGBM), which markedly improved detection of the minority attack classes and were scalable to big data solutions, but its incorporating of set preprocessing and balancing step caused latency, which impacted the real-time application of the model. Similar to this, using pre-engineered features reduced its usability in responding to new and evolving attack types.

Sunitha and Chandrika [21] have examined similar undergraduate issues that persist in wireless sensor networks (WSNs), such as reliable routing, fault tolerance, and anomaly detection. They proposed that data mining and soft computing techniques could optimize sensor operation to improve anomaly detection. This is informative based on their consideration of computational intelligence as a management strategy for WSNs, but the study was mostly conceptual and lacked experimental support. While the study did represent new thinking when developed, many of the challenges needed for WSNs related to the continuous evolution of IoT technologies, and concerns related to an active attack surface and data-centric issues were not encompassed in their framework.

Kaur et al. [22] have proposed P2ADF, a privacy-preserving attack detection framework for fog-IoT environments. The authors proposed using lightweight cryptographic mechanisms, along with distributed anomaly detection, to maintain data confidentiality and security in IoT ecosystems. P2ADF provided a trade-off between privacy and detection, but it did add computational overheads that increased energy consumption; this was problematic for battery-constrained IoT devices. Moreover, while the system put security mechanisms in place during transmission, limited mechanisms existed for protecting data-at-rest at fog nodes.

This study progresses the field by proposing a novel DRL-EAIPS framework, consisting of an optimized, quantized CNN scoring module for node-level anomaly detection tightly integrated with a DQN agent learning energy-aware prevention policies. This structure reduces both computational overhead and energy consumption while producing reliable and accurate detection. In addition, modeling the intrusion prevention problem as an MDP with a multi-objective reward function addressing both detection rate and energy efficiency provides an adaptive and real-time intrusion prevention process to the energy-constrained IoT networks. Our extensive simulation results demonstrate a longer network lifetime, lower latency, and minimal false positives compared to the benchmark models, and address key limitations in existing methods.

3. PROPOSED MODEL

3.1 System model and assumptions

The system model depicts a multi-hop WSN of NNN distributed sensor nodes deployed to support data sensing and communication in an IoT context. Each node was initialized with a fixed energy $E_i(0)$ amount and acts under limited power resources. These nodes send sensed data to a central base station or sink node via selected intermediate nodes while utilizing a multi-hop infrastructure. At any given time, communication paths can be established, changed, or terminated based on link quality and viability of the path through the present energy level and cost. Figure 1 uses arrows to show data forwarding paths that change over time based on node failures, detection of an attack, etc., or the power level depleted from the nodes.

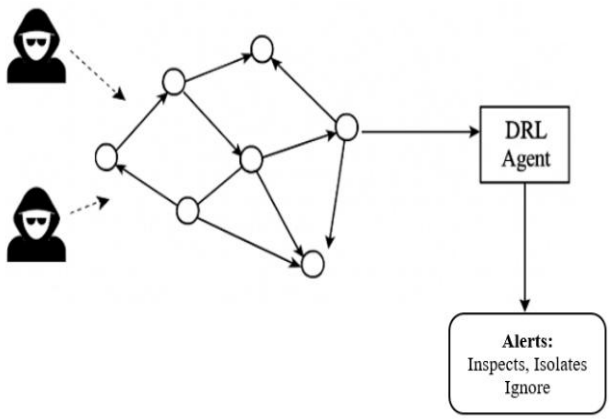

Figure 1. System model and network assumptions

Accordingly, this non-threatened environment will be at the mercy of stochastic and unpredictable attempts to gain access and compromise the overall reliability of the network. These include malicious packet injections, DoS attempts, or routing failures, and all could be directed towards either specific or arbitrary nodes. Attack surfaces will be treated as external entities that are attempting to compromise the reliability and security of the network. An attack that involves a targeted malicious node can leave itself open to packet injections, hijacked routes, or deterioration of energy usage, before critical quality-of-service and network lifetimes are affected. Therefore, a centralized DRL agent is to be a part of the overall architecture.

The DRL agent has both the option of periodically polling the state of the network continuously through state information from sensor nodes and/or edge gateways, and an event-driven operation that allows it to only respond when lightweight anomaly detectors i.e., CNN modules, are raising alerts. The state information will consist of state variables such as E_t^i, residual energy; D_t^i, delay; T_t^i, total cost of transmission; and A_t^i, the anomaly score. The DRL agent takes in these state parameters to derive an action that attempts to balance the principles of energy efficiency with threat mitigation.

The DRL agent can derive an action that results in one of these three basic types of actions: Inspect, Isolate, and Ignore. The Inspect action will initiate some additional packet-level or behavioral analysis and will incur some additional sensing or processing overhead, but will produce greater detection rates. The Isolate action is used when a suspicious node needs to be

temporarily blocked or quarantined, mostly to help contain the spread of malicious activity, but also to use up energy on rerouting and updates. The agent is also going to want to conserve energy, especially when protecting critical paths, when it selects the Ignore action, especially if the threat level is low or the energy cost for counteraction might exceed damage levels. Action types also include energy cost and level of impact on communication quality, both of which are explicitly represented in the DRL reward function.

This model operates under a time-slotted paradigm where decisions, detections, and data transmissions are made at discrete time frames. All actions, detections, and transmissions are made within those time slots, while accommodating the level of coordination of the actions to ensure operations are scheduled efficiently. It is important to note that all nodes should have the capability to assess their local conditions and transmit those state-relevant metrics without excessive overhead. As the DRL agent will be creating a model of what actions have the most impact on security and energy sustainability over time, it needs only to adapt its model due to changes in the attack profiles and constraints associated with the available resources.

This system model creates a malleable, yet representative model that highlights the primary complexities of IoT operation with respect to security: energy awareness in decision-making, time-sensitive intrusion prevention, and scalable learning. With this model, an RL-based DRL-EAIPS framework is described to enable optimization of the trade-off between the effectiveness of protection and the preservation of available resources, in constrained IoT networks.

3.2 DRL-EAIPS

In the contemporary IoT environment, energy-constrained sensor nodes face increased vulnerabilities stemming from more complex cyber-attacks such as DDoS attacks, spoofing, and wormhole attacks. Traditional IDSs are generally static, non-adaptive, and energy-draining. Our proposed framework is a DRL-EAIPS. This IDS performs real-time detection and adaptive executions based on environmental feedback to make decisions. It contains a feature extraction layer that can implement the learning on low-resource IoT nodes, a DQN agent that learns and updates optimal defense policies from historical and real-time stored information, a limited energy monitor and threat profiler that keeps track of health within the network layer, and an object that executes response action. The distributed architecture of DRL-EAIPS enables adaptive, intelligent, and autonomous intrusion response while limiting energy usage in IoT systems. Moreover, energy usage generates accurate QoS.

The proposed DRL-EAIPS consists of four primary elements: the observation space for state representation, the action space, the reward function, and the policy learning component. Each of these elements is essential to the system's ability to control intrusion response effectiveness in a way that optimizes both energy efficiency and its QoS properties with respect to the dynamically changing IoT environment in which the DRL-EAIPS operates.

3.2.1 State representation

The observation space, i.e., the system's state, uniquely represents the contextual features in an IoT environment at each action moment. The system must observe and cumulate the residual energy of all the IoT nodes, the observed latency as delay due to the actual data transmission, the determined cost of transmission given factors such as hop count, bandwidth available, and channel condition, and then at the edge, the decision-making agent will compute an anomaly score using the lightweight intrusion detection threat detection mechanism. Addressing all of these metrics collectively provides a physical view of the health of the network's operations and threat exposure. The observation space is a multi-dimensional state vector that continuously updates and represents the data transmitted.

This continuous multi-dimensional state vector will allow the decision-making agent to feed information into its situational understanding to learn useful actions based on previous action states taken. Each current state at a selected time step t's states the operational/physical condition of each respective node. The state vector contains decision-making agent information regarding four main metrics, are shown in Eq. (1). These observations form the state vector s_t, which is used as input for the policy network.

$$S_t^i = \{E_t^i, D_t^i, T_t^i, A_t^i\} \tag{1}$$

where, E_t^i represents the node i's remaining energy, D_t^i is an average packet delay, T_t^i is the cost to transmit one unit of data based on hop count and channel quality, A_t^i specifies an anomaly score representing evidence of malicious behavior from local IDS. Each observation has its place in the state vector s_t, which is used as input to the policy network. This vector is sent to a local edge gateway or fog node, which merges data from all nodes into a global state as shown in Eq. (2):

$$S_t = \{s_t^1, s_t^2, \dots, s_t^N\} \tag{2}$$

This state represents the health and security context of the network overall, at time *t*. The Anomaly Score A_t^i uses a lightweight CNN at the node level. The model has been trained offline and packaged in quantized form for edge inference as localized classification. It examined packet characteristics, including header patterns, frequency of bytes, and timing anomalies. The convolutional filter passes along the input, extracting local spatial/temporal correlations as shown in Eq. (3):

$$z_{i,j}^1 = \left(x_t^i * W^1\right)_{i,j} + b^1 \tag{3}$$

where, * is the convolution operation, W^1 is the filter/kernel of size m × k, b^1 is the bias term, $z_{i,j}^1$ is in the image convolved at location (*i,j*). In general, the convolution captures patterns such as port scanning, repeated packet sizes, and multiple combinations of anomalous header flags are shown in Eq. (4):

$$a_{i,j}^1 = \mathrm{ReLU}\left(z_{i,j}^1\right) = \max\left(0, z_{i,j}^1\right) \tag{4}$$

This provides the model with non-linearity to learn more complicated patterns. The dimensionality reduction and retention of significant features are displayed in Eq. (5):

$$p_{i,j}^1 = \max_{(u,v)\epsilon pool} a_{u,v}^1 \tag{5}$$

This step gives a form of translation invariance, resulting in a smaller memory footprint. We flatten the pooled output or states and perform a dot product with a matrix of weights (W^2)

as shown in Eq. (6):

$$Z^2 = W^2 \cdot \text{flatten}(p^1) + b^2 \quad (6)$$

where, $W^2 \in R^{d \times f}$ represents the weights for the fully connected layer, d is the number of hidden units, and f is the total number of features in the flattened input. The final output is a scalar anomaly score $A_t^i \in [0,1]$ as shown in Eq. (7):

$$A_t^i = \sigma(z^2) = \frac{1}{1+e^{-z^2}} \quad (7)$$

The output will be a sigmoid function, which is structured so that $A_t^i \approx 0$ for normal or benign traffic and $A_t^i \approx 1$ for highly suspicious or malicious traffic. The convolution unveils patterns such as port scanning, repeat packet sizes, and impossible header flag combinations. Where, A_t^i : feature vector of traffic at the node I, and θ is the quantized CNN parameters. The anomaly score is normalized in [0,1], so that if the score is above 0.5, this indicates suspicious behavior. This preliminary filtering can be executed on upstream devices and saves potential collision overhead that invokes DRL, which is usually a heavy portion of the algorithm. The full state S_t is provided as the input to a unique DQN that estimates Q-values for all possible actions as shown in Eq. (8):

$$Q(S_t, a;\ \theta) \rightarrow Er \quad (8)$$

The agent selects which action a_t to take using a ε-greedy policy as shown in Eq. (9):

$$a_t = \begin{cases} \text{rand(a)} & \text{with probability } \varepsilon \\ \text{argmax}_a Q(S_t, a;\ \theta) & \text{with probability}(1\ -\ \varepsilon) \end{cases} \quad (9)$$

3.2.2 Action space

The action space is the set of possible actions that the system can take if there is a detection of a threat or anomaly. The traditional way of decision-making by binary values (allow/deny) is replaced by a fine-grained set of actions: Allow, Drop, Quarantine, and Reroute. The Allow action is taken to forward the packet normally when the traffic is deemed safe. The Drop action is taken to drop the packet(s) that have been scored as magic/suspicious. The Quarantine action is taken to temporarily isolate a node's communication so that it can be observed before a major action occurs, allowing the neutrality of the traffic and preventing an analyst from acting immediately due to a false positive. The Reroute action is taken to divert traffic to a different communications path if the node or path has potentially been compromised. These fine-grained action outputs enable the system to make decisions and reduce the effect of threats on the network flow. The actions are chosen based on long-term expected reward, based on the energy and security constraints as shown in Eq. (10):

$$A = \{\text{Allow}, \text{Drop}, \text{Quarantine}, \text{Reroute}\} \quad (10)$$

The action space is the set of possible responses the IDS agent can employ: Allow means forward packets as normal (benign traffic), Dropped means discard packets that have been flagged as malicious (low threat score), Quarantine means temporarily isolate a node to observe its behavior, and Reroute means temporarily divert traffic around suspicious and congested paths.

3.2.3 Reward function

A reward function in RL is a critical component. His reward function gives feedback in the form of rewards based on the agent's actions to facilitate learning. The proposed structural framework fosters a multi-objective reward function to reward security actions, but still accounts for efficient resource usage in choosing a security preference. Some positive feedback is provided for a correct threat mitigation measured by some score of security, and I decide to penalize actions with a lot of energy usage, high latency, and cost of transmission. I penalize for excessive consumption in a weighted manner so that an agent will also try to learn to minimize extreme energy consumption and also excessive rerouting. Instead, I want the agent to develop policies to keep the network functioning in the long term for addressing the security threat. To the agent, functions give gradients of feedback as a reward in the form of the following Eq. (11):

$$r_t = \lambda_1 S_t - \lambda_2 E_t - \lambda_3 D_t - \lambda_4 C_t \quad (11)$$

where, S_t is the threat mitigation, or score, which gives a higher reward for correctly blocking attacks, E_t is the energy consumed to take action a_t, D_t is the delay created by taking an action, for example, rerouting can create delay, C_t is the communication cost based on forwarding, isolation, or route discovery. The reward function encourages security actions but still considers arriving at the most resource-conservative option. For the coefficients λ_1 to λ_4, they can be tuned to account for specific network behaviors or essentially how important user priorities play into the reward, for example, if a network was security-critical, λ_1 would be significantly higher.

3.2.4 Policy learning

The final element is the policy learning mechanism, which is a DQN that learns the optimal mapping from states to actions. The DQN approximates the action-value function $Q(s, a)$, which is the expected cumulative reward received for taking action a in state s. The DQN learns by iterative training with experience replay and updating its predictions with the target network. The learning mechanism works on the basis of experience from actions taken. As the agent receives feedback from its actions, its internal model is updated in a way that maximizes the anticipated reward from its next decisions. Experience from the past, in the form of cumulative total rewards, is retained and allows the DRL-EAIPS agent to dynamically adapt to changing network-state conditions and unknown attack patterns in real time without additional re-configuration by the user. The DRL agent updates its Q-values with experience replay and the Bellman Eq. (12):

$$Q(S_t, a_t) \leftarrow Q(S_t, a_t) + \alpha \left[r_t + \gamma \max_{a'} Q(s_{t+1}, a') - Q(S_t, a_t) \right] \quad (12)$$

where, α represents the learning rate, γ represents the discount factor for future rewards, and transitions (S_t, a_t, r_t, S_{t+1}) are stored in a replay buffer and sampled randomly to break the correlation between the samples. A target network Q' is updated every K steps to make the learning process more stable. The work learns an optimal mapping of states to actions through a DQN as shown in Eq. (13):

$$Q(s_t, a_t;\ \theta) \approx E\left[r_t + \gamma \max_{a'} Q(s_{t+1}, a'; \theta')\right] \quad (13)$$

where, Q represents an estimated cumulative reward of state-action pairs, θ represents the DNN weights, γ represents the discount factor of future rewards, and θ' represents the target network weights for stable learning. The DQN was used with experience replay for storage, and sample past interactions and target network separation made learning unstable. The thing is to maximize the cumulative reward as shown in Eq. (14):

$$\max_{\pi} E[\sum_{t=0}^{T} \gamma^T r_t] \quad (14)$$

Subject to, Energy constraints is $E_i \geq E_{min}$oS constraints is $D_i \leq D_{max}$ and Correct intrusion mitigation is $S_t \geq \delta$. This format helps ensure the agent learns a sustainable defensive policy in a stochastic, partially observable IoT environment.

4. RESULTS AND DISCUSSION

To provide a thorough evaluation of the proposed DRL-EAIPS, three benchmark datasets with varying IoT threat profiles were selected: NSL-KDD, UNSW-NB15, and BoT-IoT. Each of these datasets provides a wide array of network traffic patterns and types of attack that ultimately allow for the ability to verify the generalizability of the model.

- NSL-KDD: A cleaned version of the KDD Cup 1999 dataset, NSL-KDD addresses some of the problems with KDD, such as redundant records and unbalanced classes. It has around 125,973 training records and 22,544 testing records that are labeled into five classes: Normal, DoS, Probe, R2L, and U2R attacks. Each class contains a balanced number of samples, allowing for more reliable training and testing.
- UNSW-NB15: This dataset was created under real, modern network traffic and contains 2.54 million records with nine different attack types, including Fuzzers, Analysis, Backdoors, DoS, Exploits, and Worms. The attacks and evasion attempts present in this dataset exhibit a considerable amount of variation and are designed to imitate new styles of attacks in a live and modern IoT network environment that contains evasive actions from highly sophisticated attackers.
- BoT-IoT: The focus of this dataset is threats that are based on IoT, focusing on the botnet traffic with a lot of class imbalance. It contains records of attacks with records that capture more than 72 million records with multiple attack types like DDoS, DoS, reconnaissance, or keylogging.

The dataset's imbalance was managed by utilizing data augmentation to create better data balance in conjunction with stratified sampling during training in order to reduce bias, which could affect performance, caused by strongly imbalanced majority classes. For a realistic and dynamic attack simulation with NS-3, the following attack scenarios were implemented, each encompassing the following forms of attacks:

- DoS Attacks: There are a few commonly defined forms of `DoS`, such as flooding and jamming, in which an attacker will use a number of nodes to generate and send excessive amounts of traffic to overwhelm legitimate routing of necessary communications. In our simulations, we used passive and stochastic traffic generation methods to produce traffic bursts periodically over the attacker nodes, targeting key nodes of the network.
- Malicious Packet Injection: The attacker node(s) attempt to inject malformed packets or spoofed packets into the network, victimizing the legitimate nodes to cause a route failure or gain access to a node's integrity.
- Sinkhole Attacks: The attacker node(s) will advertise to the network that they are the best route for legitimate packets while dropping packets or otherwise manipulating legitimate packets. In our simulations, we used dynamic route manipulation through AODV to simulate attack nodes.
- Replay Attacks: The attacker node(s) will serialize packets in order to replay the packets, in hopes of confusing a specific protocol stack or creating false positives, which would generate alarms on a wiretap node.

The attacker nodes were selected randomly at the beginning of each simulation to give variability to the simulation. The attack nodes' behavior was controlled by probabilistic timers, each simulating that the attack nodes were capable of becoming active in order of regular and sporadic attack bursts. This method evaluated the survivability of the system against predictable and unpredictable threat events. The simulations ran on a testbed consisting of an Intel Core i7 processor, 16 GB of RAM, and running with Ubuntu 22.04 LTS. The adopted simulated model was time-slotted, where each time slot represented one second of network activity, allowing the observers to effectively sync the detection, decision, and communication processes. The key metrics collected by the simulations were energy consumption, network lifetime, end-to-end delay, and throughput.

The simulated WSN consisted of randomly deployed nodes, either 50, 100, or 150 nodes, inside a 500m × 500m movement area. Each sensor node was initialized with 2.0 Joules of energy and was capable of communicating with a ZigBee compatible radio model based upon IEEE 802.15.4 within an 80m range, i.e., AODV to enable multi-hop routing of communication. The communication among sensor nodes relied upon the routing model based on the AODV protocol that has been previously modified, e.g., route re-routing and isolating actions that were initiated by the DRL agent. The CBR traffic of the sensor nodes was one packet every second sent to another source node, with each packet a size of 64 bytes. The simulation was capped at 5000 time-slots, or until 20% of the nodes used up their energy.

To simulate potential real-world threats, several attack models were implemented, such as DoS attacks, e.g., flooding and jamming, malicious packet injection, sinkhole routing replication, and replay attacks. Attackers were homed on random nodes that then utilized either periodic or stochastic behaviors. This variable characteristic permits the testing of the DRL-EAIPS under both periodic and random attack conditions to simulate a dynamic or unpredictable IoT condition. The foundation of the DRL-EAIPS is the DQN that takes the state of the environment as input and maps it to the appropriate defense action. The DQN consisted of two hidden layers of 128 and 64 neurons with ReLU activation. A replay buffer of size 10,000 was implemented to store past experiences, with training carried out in mini-batches of size 64. The learning rate and discount factor γ were set to 0.001 and 0.95, respectively. A target network is synchronized once every 100 training steps to stabilize learning. An epsilon-

greedy exploration strategy was implemented, in which ε decayed from 1.0 to 0.05 shown in Table 1, to allow the trading off of exploration and exploitation.

Table 1. Parameters and values

Parameter	Value / Description
Number of Sensor Nodes	50-150
Deployment Area	500 × 500 meters
Communication Range	80 meters
Initial Node Energy $E_{i(0)}$	2.0 Joules
Packet Generation Rate	1 packet/sec (CBR)
Packet Size	64 bytes
Routing Protocol	AODV
MAC/PHY Protocol	IEEE 802.15.4 / CSMA-CA
Base Station Position	Center of field

Each IoT node comprised a lightweight CNN module that enabled real-time anomaly detection. Each IoT node received a 2D feature matrix (of 16 features such as TCP flags, byte entropy, inter-arrival times) per packet, which was trained offline using both the NSL-KDD and BoT-IoT datasets. The CNN architecture comprised of a Conv1D layer with 32 filters of size 3, followed by a ReLU activation, max-pooling layer, and a final dense fully connected layer with a sigmoid output producing $A_t^i \in [0,1]$ anomaly score. The model was quantized for edge device deployment using TensorFlow Lite to minimize memory use and processing overhead. The evaluation involved three benchmark datasets: NSL-KDD, UNSW-NB15, and BoT-IoT, each with different threat profiles: NSL-KDD for simple classification, UNSW-NB15 for recent network attack vectors, and BoT-IoT for botnet traffic, which is significantly imbalanced. These datasets were employed within the traffic generation module of NS-3 to reflect realistic attack behaviour during runtime. The anomaly scores generated by the CNN were combined with energy, delays, and link costs to form the state inputs provided to the DRL agent.

Table 2 and Figures 2 to 6 show a low-density deployment in the range of 50 nodes, typically seen in agricultural or remote environmental monitoring circumstances. The proposed DRL-EAIPS was shown to clearly perform better overall. The DRL-EAIPS achieved the best detection accuracy of 94.86% with the lowest false positive rate (FPR) of 3.42% over the other models. The total energy consumed per node was also lowest across all models at 0.129 J, resulting in an increased network lifetime of 3875 rounds, which is important when considering batteries for sensor deployments. The average communication delay was only 11.3 ms, and the transmission cost was also low due to selective inspections and low control overhead.

Table 3 shows that the deployment of 100 nodes for DRL-EAIPS exhibited superior performance. DRL-EAIPS reached the best accuracy rate at 96.87%, the lowest FPR at 2.91% and energy consumption per node of 0.143 J, leading to a prolonged network lifetime of 4321 rounds. The average delay was reduced to 12.4 ms, and the transmission costs were low as a result of the DRL agent's energy-aware conditions to either inspect or ignore packets based on its RL experience. Other models, such as CNN-GRU Hybrid IDS, could not achieve DRL-EAIPS's accuracy of 94.23% or delay of 14.2 ms, but they consumed more energy of 0.178 J, thereby impacting lifetime compared to DRL-EAIPS. Even though DQN-Based IDS was adaptable, it did not include any energy-constrained aspects in its reward model, so it suffered with respect to energy and lifetime performance. These results demonstrate that inclusion of both energy and anomaly context when applying RL leads to balanced and ecologically valid intrusion prevention behavior.

Table 2. Comprehensive comparative analysis (50 nodes, 10,000 iterations)

Model	Accuracy (%)	FPR (%)	Energy Consumption (J)	Network Lifetime (Rounds)	Average Delay (ms)
Proposed DRL-EAIPS	94.86	3.42	0.129	3875	11.3
DQN-Based IDS	90.45	5.91	0.168	2950	13.9
Energy-Aware SVM	85.62	6.98	0.183	2694	16.4
CNN-GRU Hybrid IDS	92.17	4.45	0.159	3120	12.7
TDRL IDS (Trust-Based)	89.88	6.11	0.172	2804	14.8

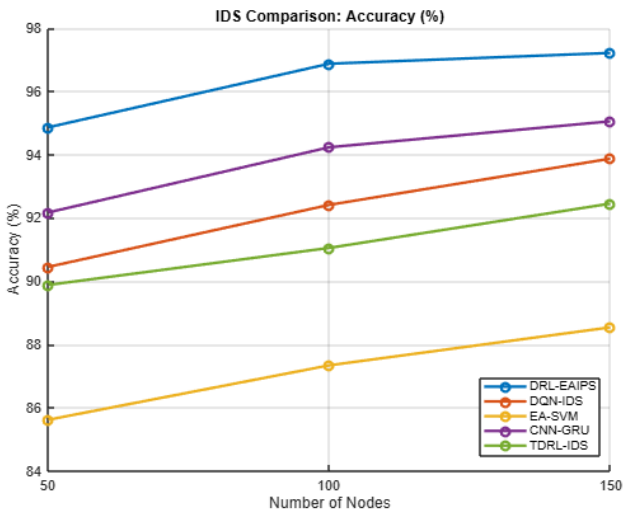

Figure 2. Accuracy vs. number of nodes

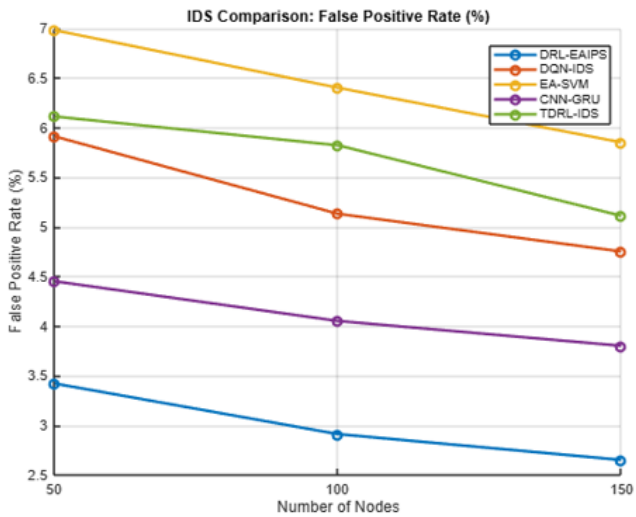

Figure 3. FPR vs. number of nodes

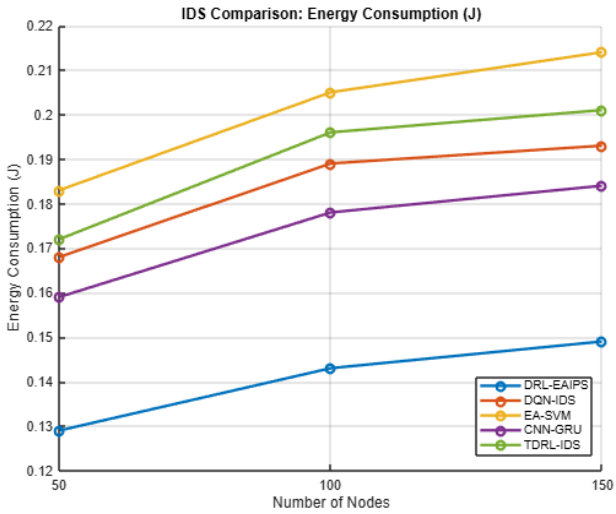

Figure 4. Energy consumption vs. number of nodes

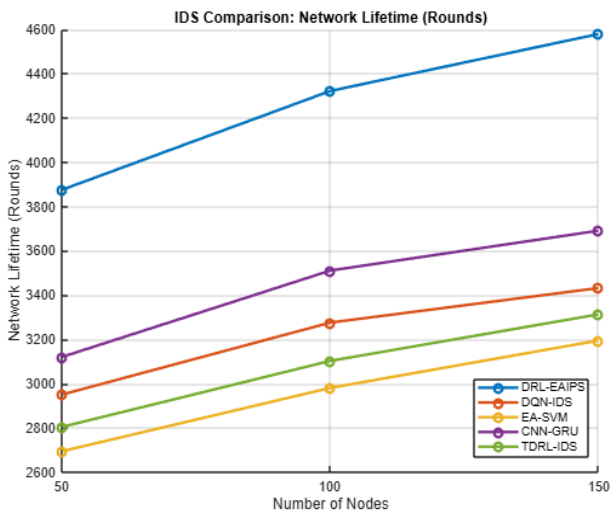

Figure 5. Network lifetime vs. number of nodes

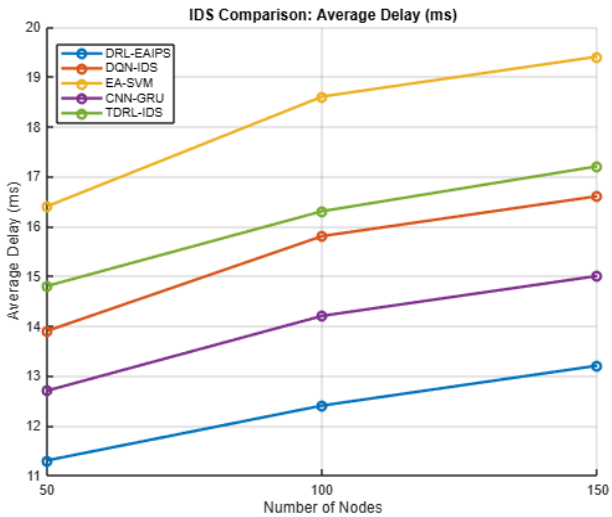

Figure 6. Average delay vs. number of nodes

Table 3. Comprehensive comparative analysis (100 nodes, 10,000 iterations)

Model	Accuracy (%)	FPR (%)	Energy Consumption (J)	Network Lifetime (Rounds)	Average Delay (ms)
Proposed DRL-EAIPS	96.87	2.91	0.143	4321	12.4
DQN-Based IDS	92.41	5.13	0.189	3275	15.8
Energy-Aware SVM	87.34	6.40	0.205	2980	18.6
CNN-GRU Hybrid IDS	94.23	4.05	0.178	3510	14.2
TDRL IDS (Trust-Based)	91.05	5.82	0.196	3102	16.3

Table 4. Comprehensive comparative analysis (150 nodes, 10,000 iterations)

Model	Accuracy (%)	FPR (%)	Energy Consumption (J)	Network Lifetime (Rounds)	Average Delay (ms)
Proposed DRL-EAIPS	97.21	2.65	0.149	4578	13.2
DQN-Based IDS	93.87	4.75	0.193	3431	16.6
Energy-Aware SVM	88.54	5.85	0.214	3194	19.4
CNN-GRU Hybrid IDS	95.05	3.80	0.184	3690	15.0
TDRL IDS (Trust-Based)	92.45	5.11	0.201	3312	17.2

Misclassification Rate: The main misclassification problem we dealt with was FPR, which was benign traffic that was misidentified as malicious. Even though our average FPR was 2.65% to 3.42%, if we look closely, we can see that false positives generally occurred most frequently during times of network congestion, as well as in rapidly changing topologies,

especially in the denser (150-node) deployments. The high variability of legitimate traffic patterns and transient delays that were sometimes observable in the denser deployments may have resulted in false spikes in CNN anomaly scores. Likewise, amongst the missed detections (i.e., false negatives) were sophisticated, low-rate attacks, typically stealthy sinkhole and replay attacks, attacks that look similar to normal traffic patterns. Although DRL-EAIPS reduced missed detections from the baseline models, detecting these subtle threats will continue to be difficult due to the low-profile nature of these attacks.

Table 4 shows the deployment with 150 nodes, where both the routing complexity is higher, and energy contention is also elevated; the DRL-EAIPS maintained its superiority and achieved an impressive accuracy of 97.21% (FPR 2.65%), topping all competing models. Notably, despite the network complexity, energy consumption per node was maintained at 0.149 J, and a network lifetime of 4578 rounds was allowed. Delay was a tad higher at 13.2 ms, which is reasonable considering the heightened packet routing among nodes from the dense network, and the transmission cost was low due to the DRL agent's context-aware action selection. Comparatively, the CNN-GRU performed well at 95.05% accuracy but had a slightly higher delay of 15.0 ms, too, and used more energy of 0.184 J. Other models, Energy-Aware SVM and Trust-Based IDS, have even lower accuracy, higher false positives, and shorter lifetimes, indicating that DRL-EAIPS is highly scalable and demonstrates robustness in complex, high-volume IoT infrastructures.

The DRL-EAIPS approach performed better than in all three deployment changes, 50, 100, and 150 nodes in overall detection accuracy while consuming less energy, having lower delay, and increasing the lifetime of the network. These results indicate the benefits of combining deep RL with a lightweight CNN-based anomaly scoring model, along with an energy-aware reward model to create a scalable, real-time, resource-efficient IDS applicable to any IoT environment. The DRL-EAIPS successfully adapts to different node densities and threat conditions when compared to both standard ML models and more recent hybrid IDS models.

5. CONCLUSIONS

This paper provides a framework for using a deep RL agent for intrusion prevention in IoT networks. The approach seeks to balance energy use and energy efficiency. The model considers the challenge of protecting IoT resources from unpredictable cyberattacks or threats. Here, we can view intrusion prevention as an energy-aware RL problem. Our approach blends the strengths of deep neural networks for feature extraction with the advantages of an RL agent that has the ability to plan security checks and actions to limit threats while minimizing energy use. We develop a model, configure state and action spaces, and define a reward function that can minimize the energy use while maximizing threat prevention. We test our approach in a simulated NS-3 IoT environment. Our findings highlight that the proposed deep RL intrusion prevention (DRL-IP) approach can maximize time to network failure, minimize energy use, and reduce communication delays compared to standard IDS/IPS approaches.

DRL-IP demonstrates connections that reduce end-to-end delay, improve the accuracy of 97.21% and outperform a few existing mechanisms. These results suggest that deep RL could provide a flexible intrusion prevention approach that minimizes battery life for device connections while maintaining robust security. For future work, there are a number of distinct paths that can help to advance this work, including, but not limited to, Advancing Energy Models, Multi-layer IoT Networks and Heterogeneous Networks, Adversarial Robustness, and Real-world Deployment and Validation.

REFERENCES

[1] Cevallos Moreno, J.F., Rizzardi, A., Sicari, S., Coen-Porisini, A. (2023). Deep reinforcement learning for intrusion detection in Internet of Things: Best practices, lessons learnt, and open challenges. Computer Networks, 236: 110016. https://doi.org/10.1016/j.comnet.2023.110016

[2] Jamshidi, S., Nafi, K.W., Nikanjam, A., Khomh, F. (2025). Evaluating machine learning-driven intrusion detection systems in IoT: Performance and energy consumption. Computers & Industrial Engineering, 204: 111103. https://doi.org/10.1016/j.cie.2025.111103

[3] Umar, H.G.A., Yasmeen, I., Aoun, M., Mazhar, T., et al. (2025). Energy-efficient deep learning-based intrusion detection system for edge computing: A novel DNN-KDQ model. Journal of Cloud Computing: Advances, Systems and Applications, 14: 32. https://doi.org/10.1186/s13677-025-00762-9

[4] Alsubaei, F.S. (2025). Smart deep learning model for enhanced IoT intrusion detection. Scientific Reports, 15: 20577. https://doi.org/10.1038/s41598-025-06363-5

[5] Kalnoor, G., Gowrishankar, S. (2021). Markov decision process based model for performance analysis of an intrusion detection system in IoT networks. Journal of Telecommunications and Information Technology, 85(3): 42-49. https://doi.org/10.26636/jtit.2021.151221

[6] Shaikh, J.A., Wang, C., Us Sima, M.W., Arshad, M., et al. (2025). A deep reinforcement learning-based robust intrusion detection system for securing IoMT healthcare networks. Frontiers in Medicine, 12: 1524286. https://doi.org/10.3389/fmed.2025.1524286

[7] Rahman, M.M., Shakil, S.A., Mustakim, M.R. (2024). A survey on intrusion detection system in IoT networks. Cyber Security and Applications, 3: 100082. https://doi.org/10.1016/j.csa.2024.100082

[8] Olanrewaju-George, B., Pranggono, B. (2025). Federated learning-based intrusion detection system for the Internet of Things using unsupervised and supervised deep learning models. Cyber Security and Applications, 3: 100068. https://doi.org/10.1016/j.csa.2024.100068

[9] Li, W., Rosenberg, P., Glisby, M., Han, M. (2023). EnergyCIDN: Enhanced energy-aware challenge-based collaborative intrusion detection in Internet of Things. In Algorithms and Architectures for Parallel Processing, pp. 293-312. https://doi.org/10.1007/978-3-031-22677-9_16

[10] Aruchamy, P., Gnanaselvi, S., Sowndarya, D., Naveenkumar, P. (2023). An artificial intelligence approach for energy-aware intrusion detection and secure routing in Internet of Things-enabled wireless sensor networks. Concurrency and Computation: Practice and Experience, 35(23): e7818. https://doi.org/10.1002/cpe.7818

[11] Otoum, Y., Nayak, A. (2021). AS-IDS: Anomaly and

signature based IDS for the Internet of Things. Journal of Network and Systems Management, 29: 23. https://doi.org/10.1007/s10922-021-09589-6

[12] Shalabi, K., Abu Al-Haija, Q., Al-Fayoumi, M.A. (2024). A blockchain-based intrusion detection/prevention system in IoT network: A systematic review. Procedia Computer Science, 236: 410-419. https://doi.org/10.1016/j.procs.2024.05.048

[13] Gyamfi, E., Jurcut, A. (2022). Intrusion detection in Internet of Things systems: A review on design approaches leveraging multi-access edge computing, machine learning, and datasets. Sensors, 22(10): 3744. https://doi.org/10.3390/s22103744

[14] Lazzarini, R., Tianfield, H., Charissis, V. (2023). Federated learning for IoT intrusion detection. AI, 4(3): 509-530. https://doi.org/10.3390/ai4030028

[15] Deshmukh, A., Ravulakollu, K. (2024). An efficient CNN-based intrusion detection system for IoT: Use case towards cybersecurity. Technologies, 12(10): 203. https://doi.org/10.3390/technologies12100203

[16] Tharewal, S., Ashfaque, M.W., Banu, S.S., Uma, P., Hassen, S.M., Shabaz, M. (2022). Intrusion detection system for industrial Internet of Things based on deep reinforcement learning. Wireless Communications and Mobile Computing, 2022(1): 9023719. https://doi.org/10.1155/2022/9023719

[17] Taşcı, B. (2024). Deep-learning-based approach for IoT attack and malware detection. Applied Sciences, 14(18): 8505. https://doi.org/10.3390/app14188505

[18] AGR, R.R., Sunitha, R., Prasad, H.B. (2020). Mitigating DDoS flooding attacks with dynamic path identifiers in wireless network. In 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, pp. 869-874. https://doi.org/10.1109/ICIRCA48905.2020.9182867

[19] Yaras, S., Dener, M. (2023). IoT-based intrusion detection system using new hybrid deep learning algorithm. Electronics, 13(6): 1053. https://doi.org/10.3390/electronics13061053

[20] Dener, M., Al, S., Orman, A. (2022). STLGBM-DDS: An efficient data balanced DoS detection system for wireless sensor networks on big data environment. IEEE Access, 10: 92931-92945. https://doi.org/10.1109/ACCESS.2022.3202807

[21] Sunitha, R., Chandrika, J. (2016). A study on detecting and resolving major issues in wireless sensor network by using data mining and soft computing techniques. In 2016 International Conference on Emerging Trends in Engineering, Technology and Science (ICETETS), Pudukkottai, India, pp. 1-6. https://doi.org/10.1109/ICETETS.2016.7602979

[22] Kaur, J., Agrawal, A., Khan, R.A. (2023). P2ADF: A privacy-preserving attack detection framework in fog-IoT environment. International Journal of Information Security, 22(4): 749-762. https://doi.org/10.1007/s10207-023-00661-7