applied sciences

MDPI

Article

Adaptive Scheduling in Cognitive IoT Sensors for Optimizing Network Performance Using Reinforcement Learning

Muhammad Nawaz Khan [1], Sokjoon Lee [1,*] and Mohsin Shah [2,*]

[1] Department of Smart Security, Gachon University, Seongnam-si 13120, Gyeonggi-do, Republic of Korea; muhammadnawaz@gachon.ac.kr
[2] Department of Computer Engineering, Gachon University, Seongnam-si 13120, Gyeonggi-do, Republic of Korea
* Correspondence: junny@gachon.ac.kr (S.J.); symnshah@gachon.ac.kr (M.S.)

Abstract: Cognitive sensors are embedded in home appliances and other surrounding devices to create a connected, intelligent environment for providing pervasive and ubiquitous services. These sensors frequently create massive amounts of data with many redundant and repeating bit values. Cognitive sensors are always restricted in resources, and if careful strategy is not applied at the time of deployment, the sensors become disconnected, degrading the system's performance in terms of energy, reconfiguration, delay, latency, and packet loss. To address these challenges and to establish a connected network, there is always a need for a system to evaluate the contents of detected data values and dynamically switch sensor states based on their function. Here in this article, we propose a reinforcement learning-based mechanism called "Adaptive Scheduling in Cognitive IoT Sensors for Optimizing Network Performance using Reinforcement Learning (ASC-RL)". For reinforcement learning, the proposed scheme uses three types of parameters: internal parameters (states), environmental parameters (sensing values), and history parameters (energy levels, roles, number of switching states) and derives a function for the state-changing policy. Based on this policy, sensors adjust and adapt to different energy states. These states minimize extensive sensing, reduce costly processing, and lessen frequent communication. The proposed scheme reduces network traffic and optimizes network performance in terms of network energy. The main factors evaluated are joint Gaussian distributions and event correlations, with derived results of signal strengths, noise, prediction accuracy, and energy efficiency with a combined reward score. Through comparative analysis, ASC-RL enhances the overall system's performance by 3.5% in detection and transition probabilities. The false alarm probabilities are reduced to 25.7%, the transmission success rate is increased by 6.25%, and the energy efficiency and reliability threshold are increased by 35%.

Keywords: reinforcement learning; adaptive scheduling; cognative sensors; energy efficiency; Internet of Things; latency; false alarm rate; detection; transmission probabilities

Academic Editor: Christos Bouras

Received: 20 April 2025
Revised: 13 May 2025
Accepted: 14 May 2025
Published: 16 May 2025

Citation: Khan, M.N.; Lee, S.; Shah, M. Adaptive Scheduling in Cognitive IoT Sensors for Optimizing Network Performance Using Reinforcement Learning. *Appl. Sci.* **2025**, *15*, 5573. https://doi.org/10.3390/app15105573

1. Introduction

The Internet of Things (IoT) [1–3] plays a vital role in transforming the device-centric approach to a user-centric approach using seamless connectivity and providing pervasive services according to context and user mode [4,5]. Integrating cognitive sensors with IoT infrastructure has brought about a significant change in real applications such as energy production and transmission, intelligent transportation systems (ITS) [6], healthcare (E-Health and Telemedicine) [7], agriculture (E-Forming) [8,9], smart homes [10], and smart cities [11]. Most of the IoT is based on physical sensors embedded in other devices and home

appliances to create these smart spaces. However, these sensors are resource-restricted and require a careful strategy in deployment because the intensive use of resources can lead these systems to stall and disconnect before performing their intended tasks [12,13].

Cognitive IoT sensors collect data from the environment and disseminate it to other sensors and central authorities for decision making and further processing. A huge volume of data is produced at frequent intervals at central points [14,15]. These frequent communications inside the system not only create congestion and buffer overflow, but also consume a significant amount of energy. The detection of redundant and useless data communications has greatly affected network performance. Suppose that the same data is circulated, and the system remains busy with these unwanted packets. In that case, system resources are used for undesirable data collection, and many useful sensing cycles are wasted [16–19].

With the emergence of artificial intelligence techniques such as deep learning (DL), systems have become more resilient, responsive, interactive, and intelligent [20]. DL trains and learns from available data and responds accordingly to any situation. Typically, it consists of perception and comprehension, learning and judgment, rationality and planning, and finally design and resolution [21–23]. With the integration of cognitive sensors, most IoT devices learn and decide like humans. A cognitive sensor with different controllers collects data from the environment and transmits it to other devices and central points. All of these entities collaboratively manage the situation and respond to any event. Based on an external event with machine learning procedures, cognitive sensors play an important role in data collection and automation [24]. Using these machine learning procedures, the devices can optimize according to the situation and the user mode and can provide services according to the situation based on the external event [25,26]. The main aim is to develop a novel scheme that uses reinforcement learning to optimize the operations of an IoT network, according to traffic conditions and sensor functions. Instead of using the static structure of CAS-IIoT-RL [22], LSTM-RL [27], and AEM-RL [28], ASC-RL has been implemented for a dynamic IoT. These baseline schemes use vector features, time series, and QoS metrics, respectively, while ASC-RL uses sensor states and traffic conditions for the optimization of network operations. In these schemes, no critical infrastructure parameters are found, while in the proposed method, there are three types of parameters. These changes affect all performance parameters, including energy enhancement, sensor reconfiguration, and minimizing delay, latencies, and packet loss.

The main contributions of the ASC-RL are as follows.

- Propose a novel scheme for the use of cognitive techniques in IoT sensors with a reinforcement learning procedure to dynamically change the state of the sensors. A sensor state model is established, and each sensor adapts its new state based on three types of parameters. These changes affect all performance parameters, including energy enhancement, sensor reconfiguration, and minimizing delay, latencies, and packet loss.
- Define and utilize three types of parameters to create a reward function in which the states adaptively switch from current to new states, and the agent learns from the traffic condition and plays a vital role in changing these states.
- Implement the proposed ASC-RL in Python and check its applicability with various parameters such as joint Gaussian distributions, event correlations, prediction accuracy, and energy efficiency with a combined reward score. Finally, a comparative analysis was performed with the detection and transition probabilities, false alarm probabilities, and transmission success rate.

Problem Statement and Motivation: Cognitive sensors often create massive amounts of data with redundant and repeated bit values. Sensing this redundant and useless data

and broadcasting not only requires a good amount of energy, but it also affects many other performance parameters such as reconfiguration, delay, latency, and packet loss. To address these challenges, we need a new method to check and adjust the state of cognitive sensors according to the roles and functions within these networks. A reinforcement learning-based approach is useful for dynamically switching states and adjusting sensors. Here, we propose a novel reinforcement learning-based mechanism called "Adaptive Scheduling in Cognitive IoT Sensors for Optimizing Network Performance using Reinforcement Learning". There are three types of parameters: internal parameters (states), environmental parameters (sensing values), and history parameters (energy levels, roles, and number of switching states). The mechanism aims to reduce extensive sensing, minimize costly processing, and control frequent communication, as well as reduce network traffic and optimize network performance in terms of network energy.

The rest of this paper is arranged as follows: Section 2 is the related work in RL-based approaches with cognitive IoT networks; Section 3 is the preliminaries and basics of ASC-RL; Section 4 is the details of the system model of ASC-RL; Section 5 is the simulation parameters and concerned metric values that are discussed; Section 6 is the evaluation part; Section 7 compares the proposed scheme with other schemes; and finally, the paper is concluded in Section 8.

2. Related Work

Combining cognitive awareness with reinforcement learning makes the system more intelligent and responsive, which is very useful in quicker decision-making. The system thoroughly checks the contents of the detected data packets and, based on these parameters, changes the state of the sensors. This switch of state controls regulates communication functions, and an optimized network is established. The following are some of the works that have already been published, and they advocate for the effectiveness of reinforcement learning in cognitive sensing.

Regarding CAS-IIoT-RL [22], various types of applications have been investigated with extensive simulations based on RL. The decision-making process is improved with adaptive and dynamic decision controls in demanding industrial situations. The proposed schemes and results are only applicable and used in industry; they may need further development to implement in a real-time scenario. Another scheme is RL-IoT [29], a routing-based RL. It combines CR-IoT and CRCN to decrease delay and collision. It performs better than AODV-IoT-based schemes and competes with other schemes in using RL methods, including average data rate, throughput, and packet collision. Another RL-based scheme is SCA-RL [30], which is a proactive procedure for selecting how long a channel will be empty, using the Bayesian algorithm. It works on discovering idle channels in descending order with probabilities, reduces the spectrum handover process, and avoids collisions in retransmission.

NOMA [31] is a dynamic Q-learning-based spectrum access scheme used to increase throughput and effective spectrum access. It also learns to use the channels when busy during peak hours. It transmits power in the form of energy in PUS for interference tolerance. It is useful in the access channel, but it needs to disrupt the continuity of the packet flow. Another RL-based scheme is LSTM-RL [27], which learns spatiotemporal patterns in the collected data. It uses a decision-based agent for physical and sensor data to optimize energy while maintaining prediction accuracy. It works in two ways: for larger amounts of data, the deep Q-network-based approach is used, while for smaller amounts of data, MDP is used. It is useful for maintaining network energy with prediction accuracy, but costly in terms of calculating the long-term spatiotemporal correlations. MARL [32] has been proposed to check the state of every cognitive user. It uses the deep recurrent

Q-network and works on a cooperative approach to increase the cognitive radio network. The system has been proven analytically and validated with many inputs.

Another RL-based scheme, RL-IoT [33], has been proposed as a routing technique to minimize EED and avoid data collision due to its decision capacity. In validation, it utilizes the AODV-IoT and ML-based procedures in sensor-based interaction. MA-DCSS [34] is an NP-hard stochastic sequential optimization process designed for detection accuracy. The main problem has to be converted to a Dec-POMDP for solving in a distributed form. It uses a multi-agent deep deterministic policy gradient technique for finding the optimal control based on conditional probabilities. Mostly focused on CTDE, which is a centralized approach, it fails in distributed systems. MICRC [35] is used in heterogeneous sensor data collection using the multi-objective intelligent cluster routing procedure. RL-based routing in IoT-based WSNs has been applied to current traffic conditions, and a new design has been suggested: to divide the entire network into many unequal groups. It works better in energy enhancement, but is slow in data processing, and communication overhead is created inside the network. CIRM [36] is based on the brain's working principle and focuses mainly on the manufacturing process. The connection between robotics and cognition is simplified using a sophisticated continual learning method based on an ANN. Informally, it processes information in parallel with a counter; for unforeseen situations, the movement of the robot is adjusted.

MRL-CSS [28] has been proposed to enhance the spectrum sharing in CRNs, using the cooperative spectrum sensing (CSS) method of multi-agent deep reinforcement learning. It is based on Adaptive Partner CSS and multi-agent deep deterministic policy gradient. Its main aim is to reduce sensing accuracy and lower the communication overhead. This scheme has obtained better results in sensing accuracy, but due to greater communication costs, it is not recommended for large-scale networks. CMRL-DG [37] is based on MARL and is used in a learning-based strategy for EH-WSNs. It achieves high network performance by efficiently communicating sensor data. Its agent continuously learns and effectively utilizes the resources, even when sensor failures are encountered by other nodes. The scheme works better in EH-WSN, but it is more specific in its structure, and its performance may decline in delay-tolerant networks. SDTVA [38] is a smart city governance integration system for analyzing big data analytics and cloud computing. It uses sustainable development data from 2018 to 2024 using the Shiny app and PRISMA. It further uses some statistical and analytical results to prove the scheme's legitimacy and working procedures on big data. Specifically, it focuses on evidence mapping, machine learning tools, and bibliometric visualization in data analysis and processing in various procedures. CPSL-CM [39] is based on a safe decision-making model that combines blockchain technology with RL. It focuses mainly on communication security, resource constraints, and routing disturbances. It gets rid of ambiguities and malevolent threats in IoT with flexibility and better efficiency. It works better in fault detection.

LSTM-DQN [40] is a short-term memory deep Q-network RL-based technique used to improve energy efficiency in IoT-based target tracking systems. Dynamically selects the most energy-efficient sensor based on a minimum distance state function. This scheme is useful in maintaining efficient target tracking with lower energy consumption. It is better in energy utilization, but requires more calculation in the RL strategy. MOSA-RL [41] is the combination of multihead self-attention and multiple agents. Channel selection and energy efficiency benefit from centralized training and a decentralized execution architecture. The rewards are continuously updated with a dynamic and multi-constraint proportional function. It also helps in distributing attention using a multihead self-attention mechanism. It improves throughput, convergence, and flexibility, but experience more delay and latency in communication. RL-ORI [42] is mainly used to improve individualized healthcare

services. Due to the use of extensive sensors with RL strategies, it dynamically modifies the activities based on individual reactions from real-time data to update its position adaptively. It improves decision-making in the medical processes and increases efficiency. IDR-FRL [43] has ensured high-speed data routing with frequent node relocations, scalability, and energy efficiency. With Federated RL, it manages the load balance and localizes with the routing cost. It has obtained good results in terms of lowering latency and packet loss but has failed in the large-scale scalability problem.

MURPPO [44] is a multi-UAV reconnaissance proximal policy scheme to identify and locate radiation sources in urban areas. It is based on distributed RL and on a dual-branch actor structure for controlling and finalizing decision-making. The reward function and the agent combined use a task-specific approach. It ensures better results in localizations and completely ignores the authentication error. DRL-CRS [45] is RL-based for the efficient use of pulse-agile radar systems in crowded areas of spectrum use. The agent updates the waveform with different parameters such as network distortion, bandwidth consumption, and collision avoidance. Improves high-resolution operations due to its careful editing. It is useful in low operations, changing spectrum-sharing situations, but it is slow due to its complexity. MDRL-RA [46] was proposed to improve QoS, which includes low latency and high throughput. It manages all things in the centralized training and decentralized execution of a multi-agent proximal policy optimization technique. LSTM layers are utilized to detect errors. It further improves the transmission success, capacity, and payload delivery, but due to its complex structure, it may cause delay and overhearing. MRL-IPP [47] is based on Q-RTS with a multi-agent procedure that has been used for robotic applications. With the increase in agents, the convergence time decreases, allowing for limited training iterations. It is more scalable and reliable, but more specific, and it needs to be generalized to other applications.

All of the above schemes have worked in cognitive networks with RL, but most of them have been applied as a trade-off between different parameters. They implement specific domains and most of the time implement a passive approach, while we need an active approach that works under real-time traffic conditions with other environmental and internal parameters. Table 1 provides an overview of various systems, including their fundamental ideas, advantages, and limitations.

Table 1. Summary of the literature of RL-based schemes and cognitive IoT networks.

Scheme	Parameters Used	Advantages	Drawback (s)
CAS-IIoT-RL [22]	Adaptive, dynamic decision controls	Enhanced decision making in industrial settings	Limited to simulations, lacks real-time validation
RL-IoT [29]	Routing, CR-IoT, CRCN, retransmission	Decreased delay and collisions, improved throughput	Specific to routing, generalization may be limited
SCA-RL [30]	Bayesian algorithm, idle channel prediction	Reduces spectrum handover, avoids retransmission collisions	Limited adaptability in dynamic environments
NOMA [31]	Dynamic Q-learning, spectrum access	Increased throughput and spectrum utilization	Disrupts continuity of packet flow
LSTM-RL [27]	Spatiotemporal patterns, DQN, MDP	Optimizes energy, maintains prediction accuracy	High computational cost for long-term prediction
MARL [32]	Deep recurrent Q-network, cooperative approach	Validated improvement in cognitive radio networks	Requires high-level coordination
RL-IoT [33]	EED minimization, AODV-IoT, ML techniques	Avoids data collision, improved routing	Dependent on specific validation scenarios
MA-DCSS [34]	Dec-POMDP, CTDE, conditional probabilities	High detection accuracy, optimal control	Fails in distributed systems
MICRC [35]	Multi-objective clustering, RL routing	Better energy enhancement, new network design	High data processing and communication overhead
CIRM [36]	ANN, continual learning, brain-based model	Adaptive robotic movement in manufacturing	Informal structure may lack robustness
MRL-CSS [28]	Multi-agent DDPG, adaptive CSS	Improved sensing accuracy, cooperative sensing	Not scalable due to high communication cost

Table 1. *Cont.*

Scheme	Parameters Used	Advantages	Drawback (s)
CMRL-DG [37]	EH-WSNs, MARL, adaptive learning	High performance despite sensor failures	Structure-specific, poor in delay-tolerant networks
SDTVA [38]	Big data, PRISMA, Shiny app	Smart city data analytics and evidence mapping	Complex architecture, requires real-time support
CPSL-CM [39]	Blockchain, secure routing, RL	Effective fault detection and secure communication	High computational demand
LSTM-DQN [40]	Short-term memory, minimum distance function	Energy-efficient target tracking	High RL computation needed
MOSA-RL [41]	Multi-head attention, multi-agent RL	Flexible, improves throughput and convergence	Delay and latency in communication
RL-ORI [42]	Sensor-driven decisions, healthcare monitoring	Improved decision making in medical processes	Requires extensive real-time sensor data
IDR-FRL [43]	Federated RL, node relocation, load balancing	Reduces latency, packet loss	Not scalable to large-scale networks
MURPPO [44]	Dual-actor structure, task-specific rewards	Effective urban radiation localization	Ignores authentication errors
DRL-CRS [45]	Pulse-agile radar, waveform updates	Efficient in changing spectrum scenarios	Computationally complex and slow
MDRL-RA [46]	LSTM, multi-agent PPO, QoS parameters	Improved payload delivery and sensing	Delay and overhearing due to complexity
MRL-IPP [47]	Q-RTS, multi-agent scalability	Reliable, decreases convergence time	Needs generalization for wider use

3. Preliminaries

Massive volumes of data with several repetitive and repeating bit values are frequently produced by cognitive sensors. In addition to consuming a significant amount of energy in broadcasting, this redundant and useless data also has an impact on other performance metrics, including reconfiguration, latency, delay, and packet loss. We require a fresh approach to monitoring and modifying the condition of cognitive sensors based on the roles and functions inside these networks to overcome these difficulties. We need a method based on reinforcement learning that helps dynamically change the states and modify the structure of the network. Here, a novel scheme based on reinforcement learning is implemented to schedule IoT sensors. The proposed scheme (ASC-RL) must move memorylessly from one state to the next, meaning that the subsequent state is solely dependent on the current state and not on past events. In the following subsections, the basics of ASC-RL are explained in detail. The system works in a way that checks the traffic conditions with some other parameters, which include internal parameters, environmental parameters, and historical parameters. The state of the cognitive sensor will change due to changing parameters, while the RL algorithm applies all these parameters, and a reward function is created to check the rate of change. A good/bad value decides the change in the network structure and adjusts the sensor to different states. The following are some of the basic preliminaries of the proposed system. The basic notations and their meanings are shown in Table 2.

Table 2. Symbols and their meaning.

Symbols	Meaning	Symbols	Meaning
μ_n	Sensing Data	S_n	Generic State
S_{Act}	Active State	S_{Wt}	Wait State
S_{Rot}	Route State	MP_n	Microprocessing unit
SM_n	Sensing Module	RM_n	Radio Module
Ac_n	Actions	$Pr(S_t)$	Probability of stochastic process
I_{Total}	Total Current	I_{RM}	Current in Radio Module
I_{SM}	Current in Radio Link	I_{MP}	Current in Microprocessor
R_t	Reward	I_{RM}	Current in Radio Module
I_{SM}	Current in Radio Link	I_{MP}	Current in Microprocessor
RL	Reinforcement Learning	PA	Prediction Accuracy
CR	Combined Reward Score	EE	Energy Efficiency

3.1. System Model for ASC-RL

A cognitive sensor can be adjusted in various states according to its intrinsic components. A sensor consists mainly of three primary parts that participate in energy consumption. These are a microprocessor (MP_n), sensing module (SM_n), and radio module (RM_n) for communication. Consider a cognitive sensor already existing in any state (S_n) and an action (AC_n) needed to switch to another state. To represent this state and action, a real number vector is used as $F(S_n, Ac_n)$. The agent, in the form of $F(S_n, Ac_n)$, will provide a reward. The reward is responsible for changing S_n depending on the internal, environmental, and historical parameters. The Markov Decision Process is used for timely decision making in changing states from one state to another.

3.2. Four State Model

A four-state transition model based on sensor internal components was developed to ensure that no events are missed while the states change dynamically. Combining these modules generates a total of sixteen possible states. However, in this case, only four states are employed to increase energy efficiency and avoid missing any important events. In other words, when the four-state model is used, the sensor never misses an event and may change states without compromising network performance. After calculating a reward function $F(S_n, Ac_n)$, the state can change to another based on the agent's values. The sensing data slices are marked as μ; using the Markov process of changing states, this can be represented as follows: A state will remain the same if the detected data slice μ remains the same. The values of each state will change if the change is based on the number of times it is shown as $1 - \mu$. The four states with symbols and transitions are shown in Figure 1. This figure also shows the distributions of the states that change from one state to another based on the detected data packets μ.

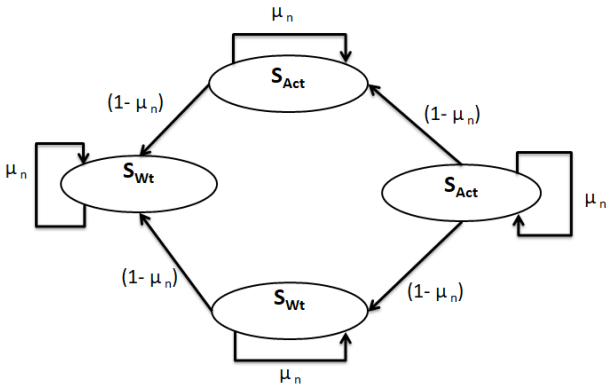

Figure 1. Four state model.

The Markov Decision Process is used to calculate the state-changing policy due to the stochastic nature of dynamic state change using a four-state model. Using the above, we have the four-state model with different probabilities because ASC-RL moves from one state to another, and these transitions may occur simultaneously or asynchronously among multiple nodes. With the above state transition values, the probabilities of each row in ASC-RL, the states changing, can be modeled as

$$Pr(S_t) = \begin{bmatrix} \mu & \frac{1-\mu}{3} & \frac{1-\mu}{3} & \frac{1-\mu}{3} \\ \frac{1-\mu}{3} & \mu & \frac{1-\mu}{3} & \frac{1-\mu}{3} \\ \frac{1-\mu}{3} & \frac{1-\mu}{3} & \mu & \frac{1-\mu}{3} \\ \frac{1-\mu}{3} & \frac{1-\mu}{3} & \frac{1-\mu}{3} & \mu \end{bmatrix} \quad (1)$$

Based on the probability distribution over these four states, the system must move somewhere; the total probability of all transitions from any given state must equal one. In Equation (1), the valid value of the stochastic matrix is 1 in each row; otherwise, the probability is marked as erroneous. For this matrix, the sums are 1.

$$\mu + 3 \cdot \frac{1-\mu}{3} = \mu + (1-\mu) = 1 \tag{2}$$

For each cognitive sensor, it measures the surroundings and processes the data using its RM_n, and based on these values, it will adjust the state of the sensor. These values of μ in the n-th slot can be written as

$$\mu(n)_{St} = \mu(1)_{St1}, \mu(2)_{St2}, ..., \mu(n-1)_{Stn-1}, ...\mu(n+1)_{Stn+1} \in \{0,1\})^{i \times j} \tag{3}$$

where $\mu(n)_{St} = \mu(1)_{St1}, \mu(2)_{St2}, ..., \mu(n-1)_{Stn-1}, ...\mu(n+1)_{Stn+1}$ are the values sensed by different sensors at different times.

3.3. Component-Based Cognitive Sensor

A typical sensor consists of five main components: microprocesssor (MP_n), sensing module (SM_n), radio-link module (RM_n), memory, and built-in power source. Here, in ASC-RL, the three components are used in the energy calculations, while the other two are assumed to be ignored. The other three components are used to calculate the energy levels in any transaction in dynamic scheduling. The current flow inside the sensor is the amount of energy that these three components use. The total current in the form of energy is shown in Figure 2, where Equation (8) is the total charge with internal components.

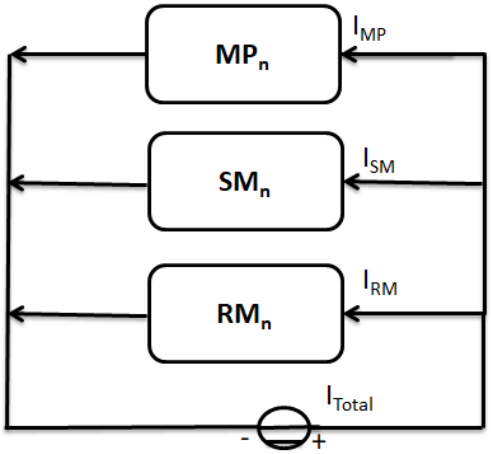

Figure 2. State changing in the ASC-RL function.

$$I_{Total} = I_{MP} + I_{SM} + I_{RM} \tag{4}$$

Here, "I" represents the current flow inside the cognitive sensor, and each module consumes an amount of energy in the system. This energy in the form of power can be discretized into many power levels that exactly match the four states. These states use different amounts of energy. It can be expressed for all sensors with states as

$$I_{Total} = f(q^{th} I_q^{t-1} / q^{th}) \tag{5}$$

Here, function $f(X_n)$ represents the discrete power levels that depend on the state of the sensor. The function can be expressed as $f(x_n)$ if I_{Total} is closest to x_n.

3.4. States in ASC-RL

The states are the energy levels that the cognitive sensor applies during working conditions. These four states can be represented by S_n; these are (S_{Act}), (S_{Wt}), (S_{Rot}), and (S_{Slp}). With the combination of the internal components, many other states may be possible, but due to the structure and functionality of these networks, in ASC-RL, only four states are used and implemented. To decide on the change in the cognitive network, many parameters are needed, but the state of the sensor is one of them. The states are the energy levels that any sensor can use depending on its current state and the contents of the detected data packets. This finalizes the relevant information that the agent needs to make a decision.

$$S_{ASC\text{-}RL} = (S_{Act}) + (S_{Wt}) + (S_{Rot}) + (S_{Slp}) \tag{6}$$

3.5. Actions in ASC-RL

The states change with actions that meet the conditions of the reward function. These actions affect the status, and they are all set up by traffic conditions and other parameters (internal, environmental, and historical). Actions can be represented with $Ac_n = Ac_{n1}, Ac_2, \ldots, Ac_{n+1}$. The action space is the combination of these states for these sensors.

$$Ac_n = Ac_{n1} + Ac_2 + \ldots, + Ac_{n+1} \tag{7}$$

3.6. Rewards in ASC-RL

The agent receives feedback for an Ac_n, based on the scope of the performance of Ac_n. It may be either good/better or bad/worse in that specific scenario. Its main aim is to learn about the state change and other parameters to maximize the cumulative reward. The feedback works like a reward function and depends on many parameters.

$$R_t = R(S_s) + R_{(\beta)} + RE_{(P)} + Ac_{(n)} \tag{8}$$

Equation (8) is the combination of five tuples for a reward function R_t. These values influence the final reward value. RS_s is the environment state space (states: S_n) for event detection and obtaining μ, R_β is the observed space, $RE_{(P)}$ is the action space for the same event, and $Ac_{(n)}$ is explained in Equation (7). After the preliminaries above, in the subsequent section, we will present a detailed model of ASC-RL.

4. Adaptive Scheduling in Cognitive IoT Sensors for Optimizing Network Performance Using Reinforcement Learning (ASC-RL)

RL-based scheduling in cognitive sensors is a promising idea, implementing different types of parameters, including internal, environmental, and historical. These parameters mainly constitute the policy of state change and create a dynamic and adaptive sensor state system applied in real-world IoT networks.

4.1. Working Procedure of the ASC-RL

The dynamic state change process for optimizing network performance is divided into two phases: the training phase and the deployment phase. In the training phase, the system is trained using sensor data (μ_n), which is a real-time collection of data, while three other parameters are used in this phase. These are internal parameters, including the state of the sensors (S_n) and historical parameters such as energy level, remaining energy, and role in detection.

The environmental parameters are similar to the sensor values. During the deployment phase, the trained agent (A_{nt}) is deployed and uses a reward function $F(s, n)$ to generate various actions (A_{cn}). The sensors are set to different states based on A_{cn}. These S_n will

be sent back to $F(s,n)$ to be A_{nt} reconfigured in the form of a circle. Figure 3 illustrates the main phases of training and deployment, including the parameters and transitions between them.

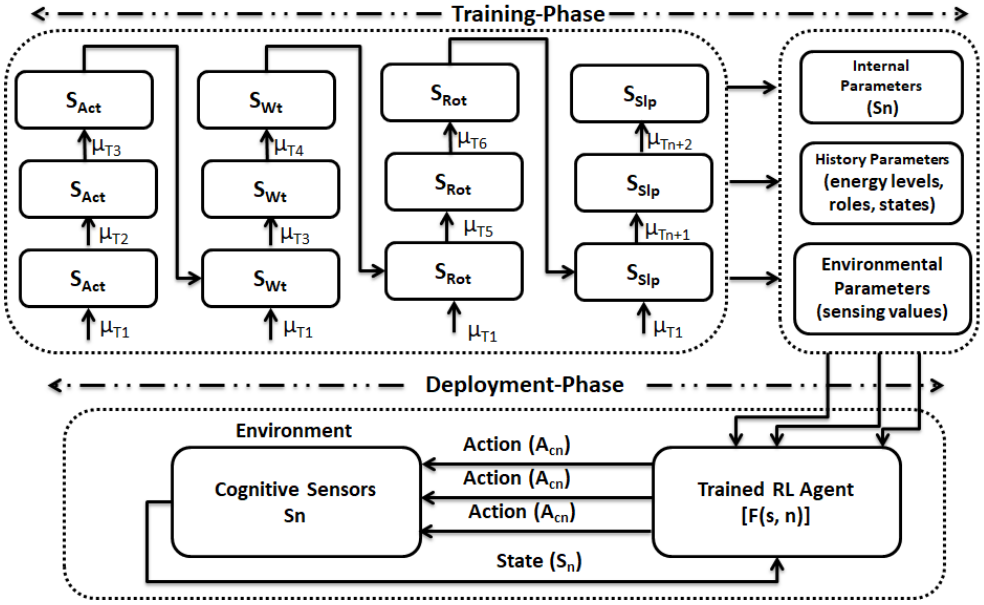

Figure 3. Generic model of ASC-RL.

4.2. Sensor Data Collection

Cognitive sensors are deployed for the data collection from any environment. Let (i, j) be two variable values, where $i = i_1, i_2, i_3, \ldots i_n$ and $j = j_1, j_2, j_3, \ldots j_n$. These are vector values of length n. i_n and j_n are sensor readings at any time T_n. The prediction probability of reading values from cognitive sensor $Pr(I/J)$ is defined as

$$Pr(I/J) = Pr(I/j_1, j_2, j_3, .., j_n) \tag{9}$$

$$Pr(I/J) = \sum_{n=1}^{n+1} Pr(j_n/j_1, j_2, j_3, .., j_n; i_1, i_2, i_3, .., i_n) \tag{10}$$

These values are random and are obtained from the environmental parameters. These values depend on the state, as shown in Figure 3. They can be written for each changing state as

$$Pr(I/J) = \sum_{n=1}^{n+1} Pr(j_n/j_1, j_2, j_3, .., j_n; i_1) \tag{11}$$

4.3. Problem Formulation

Using the MDP, the problem is derived based on the values collected from different resources, including environmental, internal, and system history. Based on these parameters, the states and actions are defined and used. The states are defined as $s \in S$, while actions are denoted as $n \in N_i$. The state-changing probabilities are defined by $Pr(s'/s, n) = Pr(S_{t+1}) = s, A_t = n$. Based on these values, an expected reward can be obtained as $r(S, s, n)$. The required reward depends on the state that changes from s to S on the expected action n. All these values are provided to the agent for persistent action. The reward R_{t+1} is in R, and this reward is the indication of a better or worse prediction. The reward is the amount of energy that is consumed by the sensor in any transaction. All sensors are in S_{Act}, and after some cycles, the network adjusts itself using traffic conditions. Other parameters can also help calculate these values. These values are defined over a range $1 - N_i$ for each sensor. Other factors are the accuracy parameter (A_P) and the energy

parameter (E_P) for each sensor. Both are used to balance accuracy and energy. When a sensor is in S_{Act}, the agent gains a level of accuracy in the form of an event detection ratio. Four states with their reward functions are depicted in the equations below:

$$R_t = \left\{ A_P \sum_{n=1}^{n+1} k_i^{<i>} - NE_P, :: IfS_n = S_{Act} \right. \tag{12}$$

$$R_t = \left\{ N.E_P + A_P. \sum_{n=1}^{n+1} (k_i^{<i>} \times k_i'^{<i>}) - A_P. \sum_{n=1}^{n+1} (k_i^{<i>} - k_i'^{<i>}) :: IfS_n = S_{Wt} \right. \tag{13}$$

$$R_t = \left\{ N.E_P + A_P. \sum_{n=1}^{n+1} (k_i^{<i>} \times k_i'^{<i>}) - A_P. \sum_{n=1}^{n+1} (k_i^{<i>} + k_i'^{<i>}) :: IfS_n = S_{Rot} \right. \tag{14}$$

$$R_t = \left\{ N.E_P + A_P. \sum_{n=1}^{n+1} (k_i^{<i>} \times k_i'^{<i>}) - A_P. \sum_{n=1}^{n+1} (k_i^{<i>} \times k_i'^{<i>}) :: IfS_n = S_{Slp} \right. \tag{15}$$

In Equations (12)–(15), the agent obtains accuracy based on the sum of the detected values, μ, which starts from any value $n = 1$ and can end at $n + 1$. For $k_i^{<i>} = 1$, $k_i^{<i>}$ is the ith element of vector k^i. Here, N is the total number of sensors at the time of state transition in a specific state S_n. When all sensors are in the S_{Slp} state, the agent receives all the gains in E_p. But in other cases, the gain in precision depends on the prediction of the μ terms of events, which is $k'^{<i>}$, in contrast to $k^i = 1$. The states are $S_n = S_{Act}, S_{Wt}, S_{Rot}$, and S_{Slp}. These equations ensure that the four states are based on the different parameters that are used in the state-changing policy. The three S_{Act}, S_{Rot}, and S_{Wt} ensure that the probability is 1, while in the case of S_{SlP}, $Pr(s/S, n)$, it will be 0 as given in the following equation:

$$Pr(s/S, n) = \begin{cases} 1 & \text{if } Pr(j_i/j_{i-1}; h_i) :: S_n = S_{Act} \\ 1 & \text{if s = } S_{st} \text{ and } S_n = S_{Rot} \\ 1 & \text{if } Pr(k_i/k_{i-1}; h_i) :: S_n = S_{Wt} \\ 0 & \text{otherwise} \end{cases} \tag{16}$$

4.4. Reinforcement Learning-Based Optimum Solutions

For dynamic state change in an IoT-based network, an optimum action results in several states that ensure two main requirements: (1) never missing an event and (2) having optimum number of S_{Act}, S_{Wt}, S_{Rot}, and S_{Slp} sensors adaptively in the whole network. For the uniform and adaptive strategy, an optimal policy $(\phi*)$ is defined. $\phi*$ is based on the Ac_n taken by A_g. The latter is a mapping from the current state to the action to be taken by the A_{nt}. This A_{nt} is responsible for maximizing the discounted cumulative reward (R_t). The discount rate is always marked as $D_{rate} \in [0.1]$.

The new states will be decided soon on the basis of this cumulative reward. $f(S_n, Ac_n)$ has already been defined, from which an output is returned, Ac_n, for the next state. The $(\phi*)$ can be defined as

$$\phi* = F(\phi*)[S_n, Ac_n] \tag{17}$$

This function operates iteratively and updates with increasing epochs and can converge to the optimal action value.

$$\phi* = F_{n+1}(\phi*)[S'_{n+1}, Ac_{,n+1}] \tag{18}$$

For $\phi_i \rightarrow \phi_i*$ and i $\rightarrow \infty$,

$$\phi_{St} = F(\phi*)[S_t, Ac_t] \tag{19}$$

$$\phi_{St} = \sum_{s'}^{S} Pr(s'/S, Ac_n)[R_t + D_{Rate} + \phi(st)] \quad (20)$$

Equation (20) is the function that manages the sensor state under policy $\phi*_n$. The outcomes from A_g are based on Ac_n and S_n, and all these make up the policy for changing the state of the sensor dynamically.

4.5. Derivation in Baseline for ASC-RL Agent

The energy consumption limits in the baselines are defined with some precision. Let any value $x_n \in X_n$ be a binary value for an optimal decision at any time t_n. Then, $x_n = 1$, which means that the sensor is in any of three states, S_{Act}, S_{Wt}, or S_{Rot}, for any slot t_n. The optimization is derived based on Equation (20). Some prediction errors can be expressed as

$$\sum_{n=1}^{n+1}(1 - x_n)\sum_{n=1}^{n+1}(k_i^{<i>} - k_i^{<i>})^2 \leq T, :: i_n \in \{0, 1\} \quad (21)$$

where "n" is any random/increasing order value, starting from 3, ..., n, and "i" is the prediction error, either 0 or 1. If "i" is 1, the sensor is in a firing state (S_{Act}, S_{Wt}, or S_{Rot}), and when "i" is 0, the sensor is in a sleep state S_{Slp}. This equation expresses the overall prediction error as a weighted binary vector. Exact value prediction is feasible in three states, except for S_{Slp}, where no communication happens.

5. Experimental Setup and Performance Metrics

ASC-RL was implemented in a particular setting using the PPO algorithm [48] and various metric values. The system was established with the right operating methods, principles, and procedures. In Table 3, different parameters, network topology, sensor count, and data generation processes are listed.

Python was used in the development of ASC-RL, using the basic concept of the PPO algorithm for reinforcement learning. It is useful because it supports the adaptive scheduling solution for IoT cognitive sensors. The *Gym* library was used to model the IoT scheduling environment, and *PyTorch* and $stable - baselines3$ were used for different reinforcement learning modules. The *NumPy* and *pandas* libraries have support for numerical calculations, and we used them for sensor data handling, and *matplotlib* was used to visualize performance.

Table 3. Simulation parameters, symbols, and metric values.

Parameter	Symbol	Matric Value
RL-Algorithm	$AlRL$	PPO "(Proximal Policy Optimization)"
Learning Rate	τ	0.0003/0.0004
Discount Factor	λ	0.998/0.989
Clip Range	δ	0.2/0.3
Epochs	E_{ephocs}	10 E_{ephocs}
No. of Sensors	N_{sn}	4–6, 10–16
Network	NT	IoT
Dimension	$\|S\|$	64–128
Data Generation Pattern	D_{PD}	Poisson distribution (λ = 2–5 packets/s)
Sensor Dynamics	–	static
Communication Topology	N_T	clustered
Transmission Range	R	100–250 m
Simulation Episodes	$E_{s/e}$	5000–10,000
Framework	FW	OpenAI Gym + PyTorch

6. Performance Evaluation of ASC-RL

Different methods are used to evaluate the performance of the RL-based scheme. Here, many parameters are used to verify performance evaluation, including joint Gaussian distributions and event correlation.

6.1. Joint Gaussian Distributions in ASC-RL

For measuring the actual energy flow in cognitive sensors, the joint Gaussian distributions $(JGDs)$ over energy, noise, and signal strengths were measured. These parameters are statistically correlated, and their behavior is visually mapped. JGD was calculated for real-time network scenarios, and here in ASC-RL, it is used for these three interdependent variables. In ASC-RL, the total cognitive sensors in these experiments consisted of four states: $S_{Act}, S_{Wt}, SRot$, and S_{Slp}. Different messages were sent, and the network was dynamically adjusted for optimal data flow with fully connected sensors. In Figure 4, the JDC for signal strengths and energy is visually mapped, with the x-axis showing the signal strength and the y-axis showing the energy with reward R_t. In this figure, the award (R_t) values are reflected by colored points at different locations. More color points at a higher order mean good/better R_t values, and that ASC-RL performs better. These colored values indicate strong signals with optimized or minimum energy consumption.

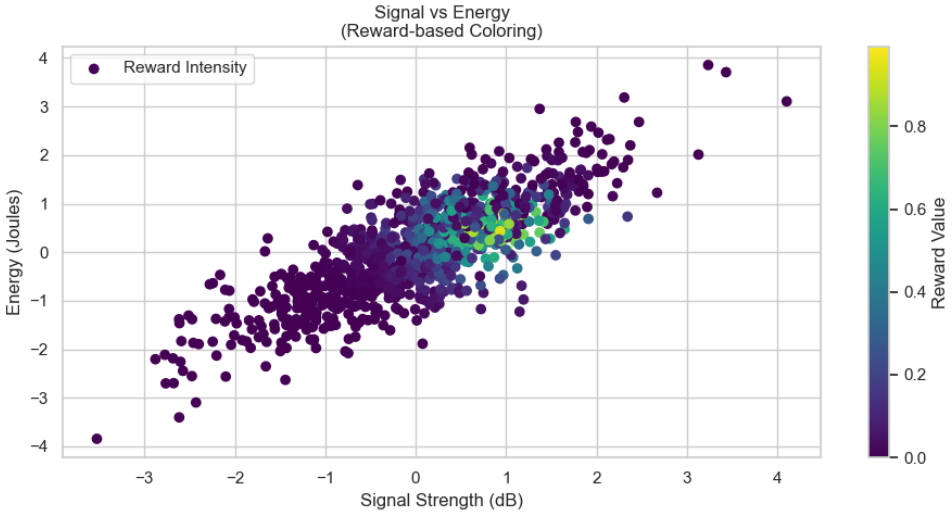

Figure 4. JDC for energy and signal strengths.

Based on the reward function, the JGD over-energy and noise are compared with the attributed values of the award R_t. In Figure 5, the JGD values are mapped at different points. The colored points at different positions depict the values of the reward function. This figure suggests lower noise values, moderate energy consumption, and a good R_t value. On the other hand, when the usage of noise and energy is considered, R_t decreases.

When comparing the values of the signal strength JGD with the noise and signal strength in Figure 6, the noise is inversely related to the signal strength, with low R_t. This suggests that low signals ultimately lead to poor communication with a low average R_t. In Table 4, a summary of the three graphs is shown with the appropriate attributes.

Table 4. ASC-RL using different parameters in JGD.

Figure	Parameters	Key Observations	Optimal Condition
Figure 4	Signal–Energy	Good signal strength and energy result in a good R_t, while lowering or raising energy consumption may result in a bad R_t	Better signal strengths, moderate energy

Table 4. *Cont.*

Figure	Parameters	Key Observations	Optimal Condition
Figure 5	Energy–Noise	Lesser noise with optimum energy improves R_t, higher noise results in bad R_t	Lower noise, moderate energy
Figure 6	Signal–Noise	Good signal and lower noise result in better R_t, a weaker signal or high noise reduces R_t	High signal, low noise

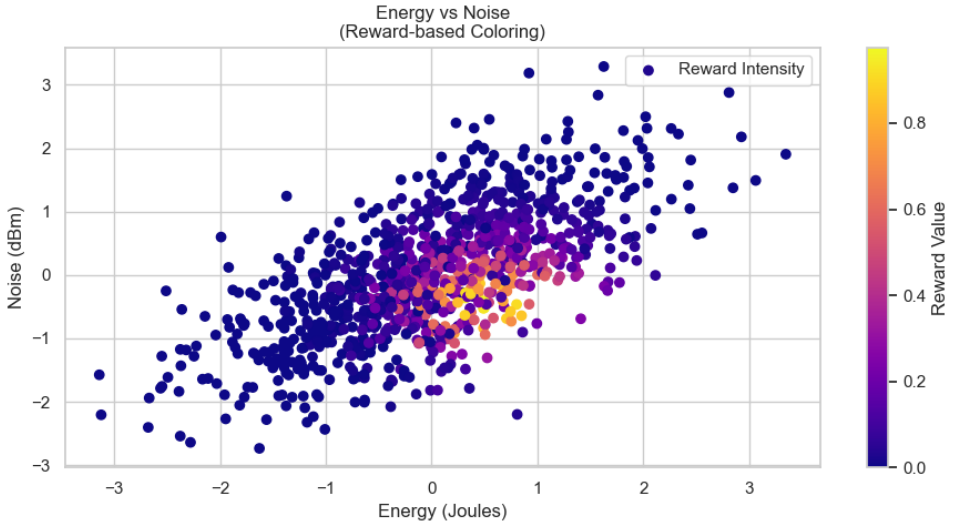

Figure 5. JDC for noise and energy.

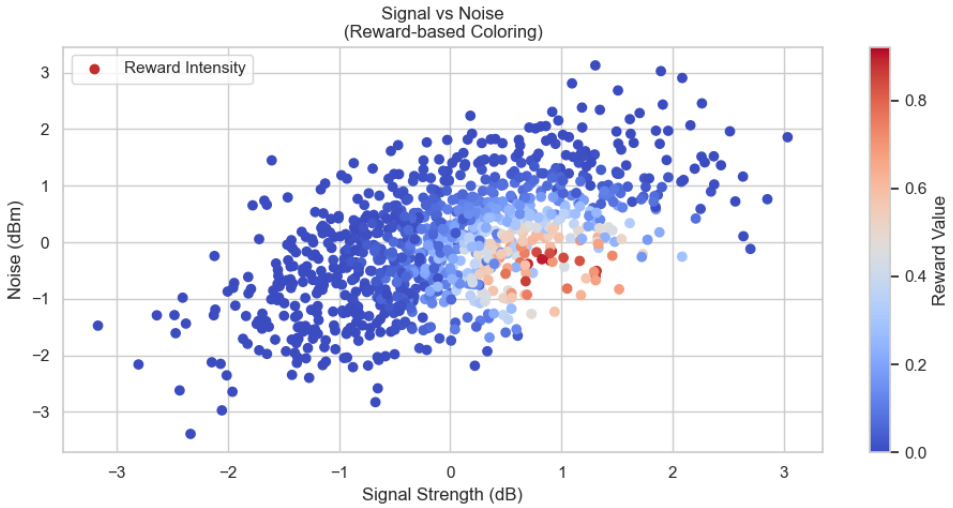

Figure 6. JDC for noise and signal strength.

6.2. Event Correlations Inside ASC-RL

This parameter is used to predict the next state and depends on various factors. Probabilistic temporal logic is applied to check the contents of bit values in different sensing cycles, and event correlation was derived from cognitive sensors. As shown in Figure 7, correlations between different cognitive sensors were derived and mapped with high and low correlations. The range of values is from "−1" to "+1", in which "+1" is a complete positive correlation with increasing and decreasing correlating values. A 0 correlation means that these sensor values have no dependency, and they have no bits that correlate with each other. A "−1" correlation is an inverse correlation; as one increases, the other decreases. If there is a strong correlation, like 0.85, it means that these sensors sense similar nature data with the same bit repetitions. In weak correlations, such as 0.05, these sensors

sense rare types of data, or they are deployed very far apart from each other. In a negative correlation, such as -0.05, it works inversely; when one increases, the other decreases. In ASC-RL, this correlation is helpful in state change based on the correlation.

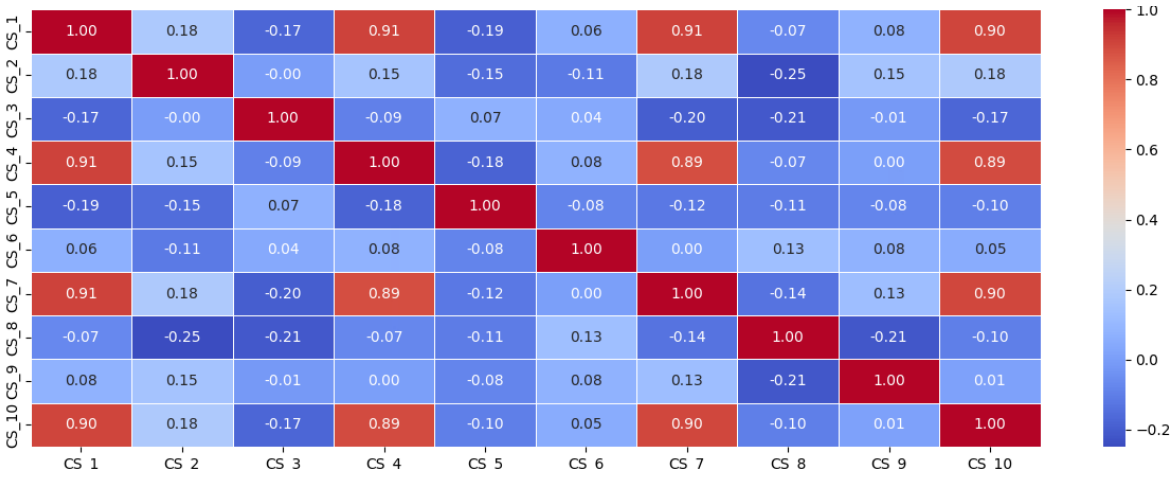

Figure 7. ASC-RL event correlation.

In correlating the values in Figure 7 with the values in Figure 1, four states for any sensor are shown with a structured Markov transition matrix. The probability of a sensor remaining in its current state is defined by the diagonal element μ, but the off-diagonal elements are defined as $\frac{1-\mu}{3}$. This equally divides the remaining probability among transitions to the other three states. Higher values of μ suggest stronger persistence in the current state. This means that each node has a constant and balanced probability of changing states. On the other hand, Figure 7, illustrates the relationships between event patterns in ten sensors. Some sensors show stronger correlations, such as CS_1, CS_4, CS_7, and CS_10, as they share a base signal. It can be correlated with Figure 1, as this phenomenon is consistent, which states that sensors with high μ and similar functions are more likely to exhibit state changes over time. The heatmap in Figure 7 illustrates that organized transitions in a Markov model can result in visible correlations between sensors.

6.3. Prediction Accuracy and Energy Efficiency with Combined Reward Score

ASC-RL has been tested with using three performance metrics: prediction accuracy (PA), energy efficiency (EE), and a combined reward score (CR). The values start with a range of 0.0 to 1.0 with increments of 0.01, as shown in Figure 8. This means that 0 values indicate that the sensors are inactive in saving energy. On the other hand, the value of 1 confirms that sensors are in any of the three states, with a maximum of S_{Act} state sensors. With the increasing S_{Act} states of multiple sensors, the PA increases (blue color) because more S_{Act} sensors collect more data and information, resulting in more accurate results. But the results do not follow linear growth, with little fluctuation. This deviation in obtained values may be due to some environmental factors like noise and interference. The weighted accuracy and efficiency (red color) is the reward score over $E_P = 0.3$ and $A_P = 0.7$, and indicates that the expected PA is approximately 70%. However, at this stage, EE is around 30%, which reveals a balance between resource conservation and predictive performance. With increasing activation of many S_{Slp} sensors, it increases the PA, but a broad activation of all sensors may affect the EE due to continuous sending. First, the reward curve increases up to a peak level and then starts declining in the region of 0.6 and 0.8 activation levels inside the network.

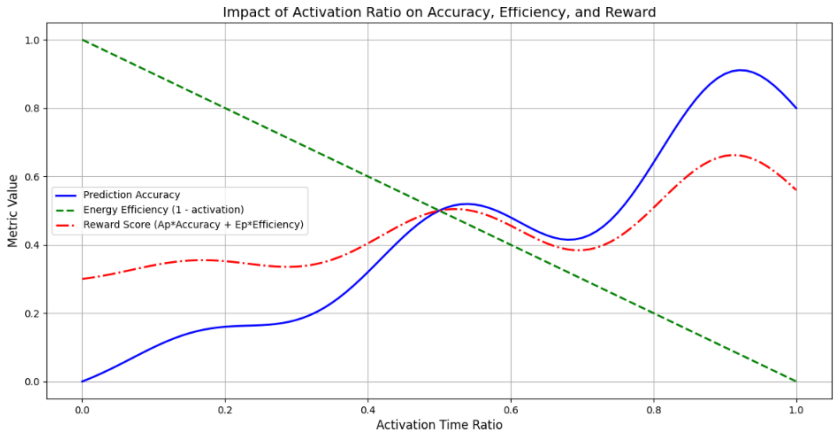

Figure 8. Prediction accuracy and energy efficiency with combined reward score.

7. Comparative Analysis

ASC-RL was experimented with and compared with other schemes to ensure its authenticity and efficiency. ASC-RL was compared to CAS-IIoT-RL [22], LSTM-RL [27], and AEM-RL [28] for various metrics. These experiments are described in more detail in the following sections.

7.1. Detection and Transition Probabilities

In the first experiment, ASC-RL was tested with other schemes in detection and transition probabilities. At the start of the experiment, at low transition probabilities of 0.0, all schemes work in the same way with an average detection probability of 0.55, as shown in Figure 9. With the increasing trend of transition probability around 0.4, ASC-RL shows better results than other schemes. This increase in detection probability suggests that ASC-RL can detect better, and it dynamically changes the states where other schemes (CAS-IIoT-RL, LSTM-RL and AEM-RL) have no such four-state policy, and their detection probability is lower. In analyzing the average percentage increase in the three schemes with the proposed scheme, as depicted in Table 5, increasing transition probabilities is shown to enhance overall performance by 3.5%. These experiments cover the transition probabilities from 0.0 to 1.0, but do not include the confidence intervals. These experiments are deterministic without repeated sampling. In the future, we want to improve the system's robustness and reproducibility by testing it with repeated trials.

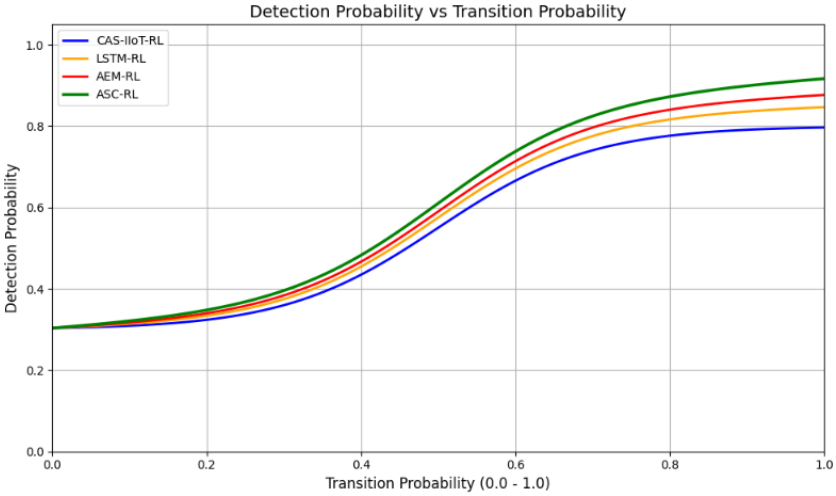

Figure 9. Detection probability and transition probability comparison of ASC-RL with CAS-IIoT-RL, LSTM-RL, and AEM-RL.

Table 5. Detection probability comparison of ASC-RL with CAS-IIoT-RL, LSTM-RL, and AEM-RL.

Pr_{St}	CAS-IIoT-RL	LSTM-RL	AEM-RL	ASC-RL	% Increase
0.0	0.55	0.55	0.55	0.55	0.00%
0.2	0.56	0.57	0.58	0.59	5.41%
0.4	0.61	0.63	0.64	0.66	6.84%
0.6	0.73	0.75	0.77	0.80	6.25%
0.8	0.86	0.89	0.92	0.95	6.93%
1.0	0.92	0.96	0.98	1.00	5.43%

7.2. False Alarm Probabilities

In comparing the values of the false alarm rate of ASC-RL with those of other schemes, initially, they exhibit the same values but increase with more episodes, as shown in Figure 10. The ASC-RL performs differently, with values decreasing due to frequent state changes based on the reward function and the training of A_g. Toward episode 100, ASC-RL achieves a 0.296 false alarm rate with an average of 0.335 for the other three schemes, representing a 10.10% increase. Similarly, upon reaching 500 episodes, the proposed scheme reduces the false alarm rate to 0.112, with an average of 0.1506 for the other schemes. Compared with the average of the three schemes, the proposed scheme accounts for around 25.7%, which shows its ability to learn the ratio of the reward function to decrease false alarm rates. These values are displayed in Table 6.

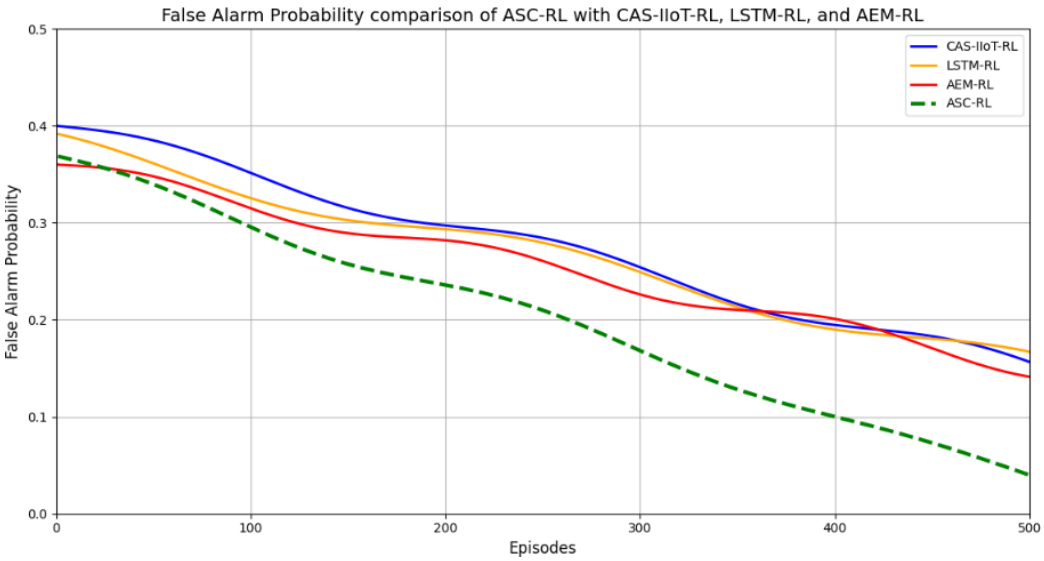

Figure 10. False alarm probability comparison of ASC-RL with CAS-IIoT-RL, LSTM-RL, and AEM-RL.

Table 6. False alarm probability of ASC-RL with CAS-IIoT-RL, LSTM-RL, and AEM-RL.

Episode	CAS-IIoT-RL	LSTM-RL	AEM-RL	ASC-RL	Improvement (%)
100	0.355	0.336	0.315	0.296	10.10%
200	0.305	0.292	0.275	0.252	13.54%
300	0.255	0.248	0.235	0.208	16.96%
400	0.205	0.200	0.195	0.160	19.61%
500	0.155	0.152	0.145	0.112	24.06%

7.3. Transmission Success Rate

For the transmission success rate, ASC-RL was implemented in different settings with the other three schemes, and it was found that the proposed scheme obtained a better transmission success rate with maximum latency thresholds. It performed better than the

other schemes, achieved about 5.73% on average, and finally reached 6.25%, as shown in Figure 11. This steady improvement suggests that ASC-RL is more suited for high-latency dynamic settings where performance is greatly affected by clever adaptive techniques, as shown in Table 7.

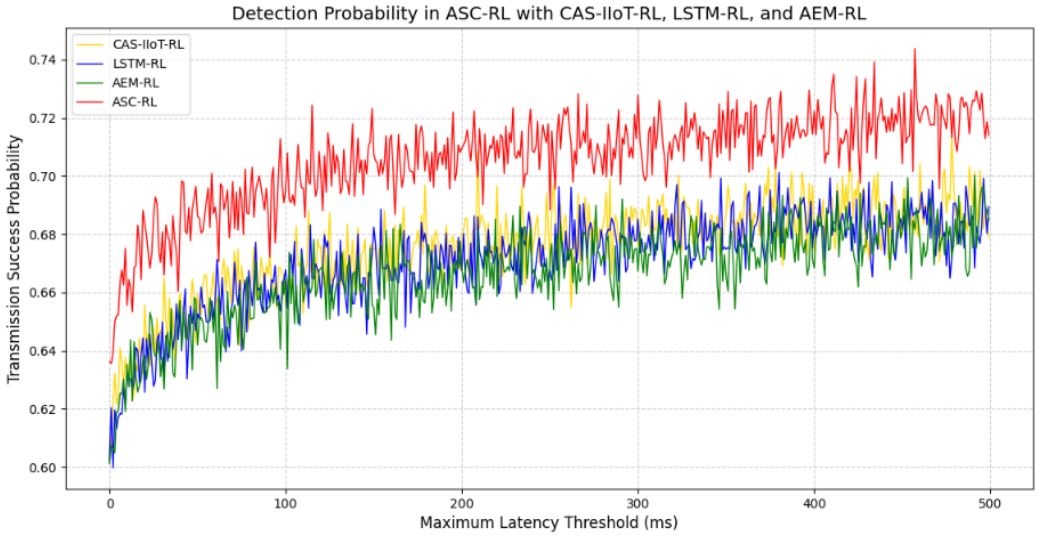

Figure 11. False alarm probability comparison.

Table 7. Transmission success probability with increasing latency threshold.

Latency Th (ms)	CAS-IIoT-RL	LSTM-RL	AEM-RL	ASC-RL	% Increase
100	0.65	0.64	0.63	**0.69**	**6.25%**
200	0.68	0.67	0.66	**0.72**	**6.06%**
300	0.71	0.70	0.69	**0.75**	**5.71%**
400	0.73	0.72	0.71	**0.77**	**5.48%**
500	0.75	0.74	0.73	**0.79**	**5.17%**

7.4. Energy Efficiency and Reliability Threshold

ASC-RL was tested with another energy parameter related to the reliability threshold. It outperforms all three schemes because LSTM-RL is based on temporal learning through recurrent networks; AEM-RL works on adaptive energy distribution, and CAS-IIoT-RL is based on context-aware sensing, which delimits the balanced energy consumption and reliability, as shown in Figure 12. ASC-RL uses a multi-objective reward function that maximizes reliability and energy efficiency. It improves the energy usage, with the reliability factor of the other three schemes being 35% and marked at each point in Table 8.

Table 8. Energy efficiency comparison of schemes with ASC-RL improvement.

Reliability Threshold	CAS-IIoT-RL (bits/Hz)	LSTM-RL (bits/Hz)	AEM-RL (bits/Hz)	ASC-RL (bits/Hz)	% Increase Over Avg
0.600	0.7550	0.7217	0.7176	0.8833	20.77%
0.644	0.6983	0.6844	0.6692	0.8328	21.76%
0.688	0.6849	0.6714	0.6502	0.8310	24.24%
0.732	0.6669	0.6661	0.6364	0.8313	26.63%
0.777	0.6461	0.6392	0.6311	0.8214	28.58%
0.822	0.6544	0.6381	0.6173	0.7991	25.53%
0.866	0.6261	0.6100	0.6103	0.8082	31.31%
0.910	0.6306	0.6161	0.5934	0.8146	32.81%
0.955	0.6111	0.6065	0.5937	0.8177	35.43%
1.000	0.6238	0.6087	0.5841	0.8040	32.77%

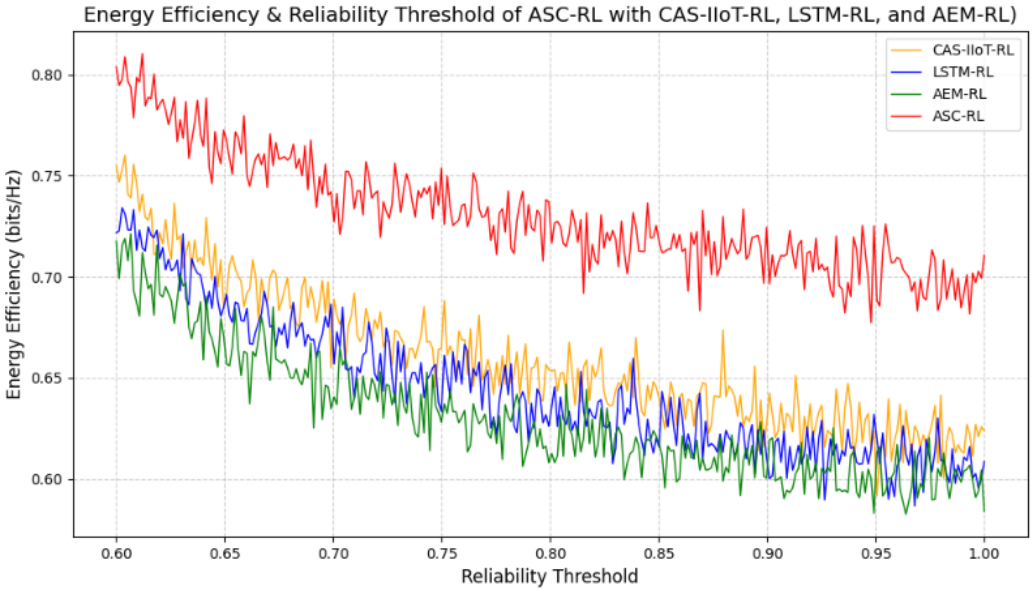

Figure 12. Energy efficiency and reliability threshold of ASC-RL with CAS-IIoT-RL, LSTM-RL, and AEM-RL.

7.5. Training Performance with Comparative Evaluation

In these experiments, the ASC-RL was compared with other baseline schemes (CAS-IIoT-RL, LSTM-RL, and AEM-RL) in an equal number of episodes with a median success rate. In fixing some hyperparameters like the discount rate and exploration strategy, the proposed scheme achieves high success rates, while the other schemes, with moderate adaptation, exhibit low success rates. In Figure 13, the behavior of these schemes is shown with 90% up to 50 epochs and less noise for ASC-RL. CAS-IIoT-RL with a delay component of up to five epochs exhibits slightly slower convergence. The other two, LSTM-RL and AEM-RL, have more noise and delayed convergence. The success rate is 85–90% and 80–88%, respectively. The statistical values of this experiment are shown in Table 9, in which ASC-RL training performance is shown with the other three baseline schemes. A total of 200 epochs, with different success rates, were used for the main metric. In this table, it is shown that ASC-RL has the highest success rate of 0.997, while the other schemes' highest success rate is 0.862, with the lowest standard deviation of 0.205. In this experiment, it is observed that a success rate greater than 0.8 is observed in just 36 epochs, demonstrating faster convergence. As far as hardware details are concerned, we intend to implement it in hardware in the future. Our published work implemented the off-the-shelf "CC2420" module with some APIs to simulate IoT-based communication.

Table 9. Training performance of ASC-RL in 200 epochs with success rate with CAS-IIoT-RL, LSTM-RL, and AEM-RL.

Method	Success Rate	Mean Success Rate	Standard Dev	Epoch@0.8+ SR
ASC-RL	0.997	0.865	0.207	36.21
CAS-IIoT-RL	0.966	0.823	0.227	43.56
LSTM-RL	0.945	0.782	0.241	49.12
AEM-RL	0.936	0.778	0.256	48.34

To verify the overall efficiency and optimization in IoT networks, ASC-RL was thoroughly compared with all known parameters with other baseline schemes, as shown in Table 10. Each scheme has its implementation, creating specific results based on the environ-

ment. Most of the parameters in this table are types of algorithms, action spaces, state spaces, confidence intervals, discount factors, the number of episodes, and implementation tools.

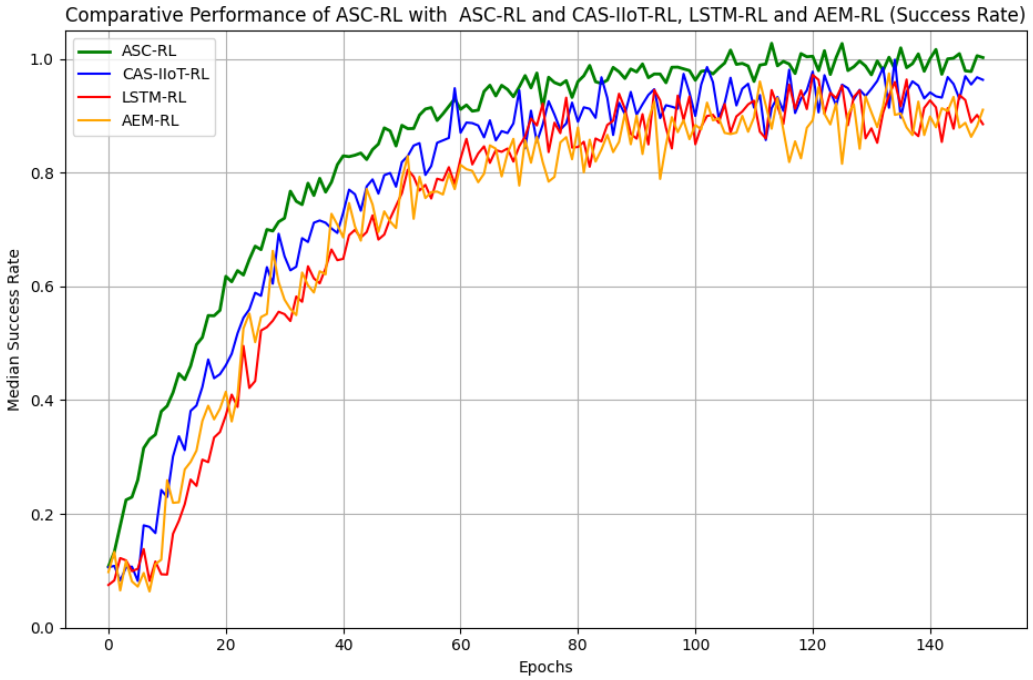

Figure 13. Compartive Performance of ASC-RL with CAS-IIoT-RL, LSTM-RL, and AEM-RL (Success Rate).

Table 10. Comparison of ASC-RL and CAS-IIoT-RL, LSTM-RL, and AEM-RL.

Parameter	CAS-IIoT-RL	LSTM-RL	AEM-RL	ASC-RL
Algorithm	DQN	LSTM	A2C	PPO
Space	Feature V_c	Time-Series	QoS Metrics	S_{sn}, μ
A_{cn}	Discrete	Discrete	Continuous	Discrete
τ	0.001	0.0005	0.0003	0.0003
λ	0.9	0.95	0.98	0.99
Archeteture	2-layer NN	LSTM + Dense	3-layer NN	3-layer NN + ReLU
Episodes	5000	7000	8000	10,000
Reward R_t	Delay, Energy	Latency, PDR	Latency, Energy	Energy, Delay
Environment	Static IIoT	Edge IIoT	Edge+ Fog	Dynamic IoT
Implementation	Python-2	TensorFlow 1.x	Python + Keras	PyTorch + Gym

8. Conclusions

Cognitive sensors have limited resources, and if no appropriate approach is used during deployment, the sensors are disconnected due to redundant sensing of the same data. The collection of redundant information in IoT networks has an impact on system performance, including energy efficiency, reconfiguration time, latency, and packet loss. To address these issues, we suggest a new reinforcement learning-based technique, "Adaptive scheduling in cognitive IoT sensors to optimize network performance using reinforcement learning (ASC-RL)". It constructs a function for a state-changing policy from three kinds of parameters: internal parameters (states), environmental parameters (sensing values), and historical parameters (energy levels, roles, and number of switching states). These states eliminate extensive sensing, reduce processing costs, and minimize communication over the radio link. The suggested system minimizes network traffic while improving network performance in terms of energy. The primary parameters assessed are joint Gaussian distributions and event correlations, with the resulting signal intensities, noise, prediction

accuracy, and energy efficiency with a combined reward score. In a comparative analysis, ASC-RL improves the overall system performance by 3.5% in detection and transition probability. The probability of false alarms is reduced by 25.7%, the transmission success rate increased by 6.25%, and the energy efficiency and reliability thresholds are increased by 35%.

In future work, we intend to create a Q-learning approach combined with reinforcement learning and apply the transfer learning procedures that allow rewards and agents to apply generic knowledge in the same tasks, reducing time and functional cost. To obtain high learning performance, we intend to incorporate prioritized sampling, experience replay, and adaptive exploration.

Author Contributions: M.N.K.: conceptualization, methodology, software, validation, and writing—review and editing. M.S.: supervision and writing—review and editing. S.L.: supervision and reviewing. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Nuclear Safety Research Program through the Korea Foundation of Nuclear Safety (koFONS) using the financial resource granted by the Nuclear Safety and Security Commission (NSSC) of the Republic of Korea (RS-2021-KN051410, Development of cyber incident response assessment technology for critical digital assets of nuclear power plant (NPP))

Conflicts of Interest: The authors declare that none of the work described in this publication may have been influenced by any known conflicting financial interests or personal ties.

References

1. Li, C.; Wang, J.; Wang, S.; Zhang, Y. A review of IoT applications in healthcare. *Neurocomputing* **2024**, *565*, 127017. [CrossRef]
2. Gkagkas, G.; Vergados, D.J.; Michalas, A.; Dossis, M. The Advantage of the 5G Network for Enhancing the Internet of Things and the Evolution of the 6G Network. *Sensors* **2024**, *24*, 2455. [CrossRef] [PubMed]
3. Khan, M.N.; Rahman, H.U.; Khan, M.Z. An energy efficient adaptive scheduling scheme (EASS) for mesh grid wireless sensor networks. *J. Parallel Distrib. Comput.* **2020**, *146*, 139–157. [CrossRef]
4. Ullah, I.; Adhikari, D.; Su, X.; Palmieri, F.; Wu, C.; Choi, C. Integration of data science with the intelligent IoT (IIoT): Current challenges and future perspectives. *Digit. Commun. Netw.* **2024**, *11*, 280–298. [CrossRef]
5. Casillo, M.; Cecere, L.; Colace, F.; Lorusso, A.; Santaniello, D. Integrating the internet of things (IoT) in SPA medicine: Innovations and challenges in digital wellness. *Computers* **2024**, *13*, 67. [CrossRef]
6. Rajkumar, Y.; Santhosh Kumar, S. A comprehensive survey on communication techniques for the realization of intelligent transportation systems in IoT based smart cities. *Peer-to-Peer Netw. Appl.* **2024**, *17*, 1263–1308. [CrossRef]
7. Pulimamidi, R. To enhance customer (or patient) experience based on IoT analytical study through technology (IT) transformation for E-healthcare. *Meas. Sens.* **2024**, *33*, 101087. [CrossRef]
8. Shahab, H.; Iqbal, M.; Sohaib, A.; Khan, F.U.; Waqas, M. IoT-based agriculture management techniques for sustainable farming: A comprehensive review. *Comput. Electron. Agric.* **2024**, *220*, 108851. [CrossRef]
9. Duguma, A.; Bai, X. Contribution of Internet of Things (IoT) in improving agricultural systems. *Int. J. Environ. Sci. Technol.* **2024**, *21*, 2195–2208. [CrossRef]
10. Magara, T.; Zhou, Y. Internet of things (IoT) of smart homes: Privacy and security. *J. Electr. Comput. Eng.* **2024**, *2024*, 7716956. [CrossRef]
11. Nassereddine, M.; Khang, A. Applications of Internet of Things (IoT) in smart cities. In *Advanced IoT Technologies and Applications in the Industry 4.0 Digital Economy*; CRC Press: Boca Raton, FL, USA, 2024; pp. 109–136.
12. Khan, M.N.; Rahman, H.U.; Khan, M.Z.; Mehmood, G.; Sulaiman, A.; Shaikh, A.; Alqhatani, A. Energy-efficient dynamic and adaptive state-based scheduling (EDASS) scheme for wireless sensor networks. *IEEE Sens. J.* **2022**, *22*, 12386–12403. [CrossRef]
13. Nilima, S.I.; Bhuyan, M.K.; Kamruzzaman, M.; Akter, J.; Hasan, R.; Johora, F.T. Optimizing Resource Management for IoT Devices in Constrained Environments. *J. Comput. Commun.* **2024**, *12*, 81–98. [CrossRef]
14. Poyyamozhi, M.; Murugesan, B.; Rajamanickam, N.; Shorfuzzaman, M.; Aboelmagd, Y. IoT—A Promising Solution to Energy Management in Smart Buildings: A Systematic Review, Applications, Barriers, and Future Scope. *Buildings* **2024**, *14*, 3446. [CrossRef]
15. Pandey, S.; Bhushan, B. Recent Lightweight cryptography (LWC) based security advances for resource-constrained IoT networks. *Wirel. Netw.* **2024**, *30*, 2987–3026. [CrossRef]

16. Sun, Y.; Jung, H. Machine Learning (ML) Modeling, IoT, and Optimizing Organizational Operations through Integrated Strategies: The Role of Technology and Human Resource Management. *Sustainability* **2024**, *16*, 6751. [CrossRef]
17. Arshi, O.; Rai, A.; Gupta, G.; Pandey, J.K.; Mondal, S. IoT in energy: A comprehensive review of technologies, applications, and future directions. *Peer-to-Peer Netw. Appl.* **2024**, *17*, 2830–2869. [CrossRef]
18. Khan, M.N.; Rahman, H.U.; Hussain, T.; Yang, B.; Qaisar, S.M. Enabling Trust in Automotive IoT: Lightweight Mutual Authentication Scheme for Electronic Connected Devices in Internet of Things. *IEEE Trans. Consum. Electron.* **2024**, *70*, 5065–5078. [CrossRef]
19. Khan, M.N.; Khalil, I.; Ullah, I.; Singh, S.K.; Dhahbi, S.; Khan, H.; Alwabli, A.; Al-Khasawneh, M.A. Self-adaptive and content-based scheduling for reducing idle listening and overhearing in securing quantum IoT sensors. *Internet Things* **2024**, *27*, 101312. [CrossRef]
20. Mumuni, A.; Mumuni, F. Automated data processing and feature engineering for deep learning and big data applications: A survey. *J. Inf. Intell.* **2024**, *3*, 113–153. [CrossRef]
21. Hu, B. Deep learning image feature recognition algorithm for judgment on the rationality of landscape planning and design. *Complexity* **2021**, *2021*, 9921095. [CrossRef]
22. Rajawat, A.S.; Goyal, S.; Chauhan, C.; Bedi, P.; Prasad, M.; Jan, T. Cognitive adaptive systems for industrial internet of things using reinforcement algorithm. *Electronics* **2023**, *12*, 217. [CrossRef]
23. Rubio-Martín, S.; García-Ordás, M.T.; Bayón-Gutiérrez, M.; Prieto-Fernández, N.; Benítez-Andrades, J.A. Enhancing ASD detection accuracy: A combined approach of machine learning and deep learning models with natural language processing. *Health Inf. Sci. Syst.* **2024**, *12*, 20. [CrossRef] [PubMed]
24. Muzaffar, M.U.; Sharqi, R. A review of spectrum sensing in modern cognitive radio networks. *Telecommun. Syst.* **2024**, *85*, 347–363. [CrossRef]
25. Ge, J.; Liang, Y.C.; Wang, S.; Sun, C. RIS-assisted cooperative spectrum sensing for cognitive radio networks. *IEEE Trans. Wirel. Commun.* **2024**, *23*, 12547–12562. [CrossRef]
26. Wang, J.; Wang, Z.; Zhang, L. A simultaneous wireless information and power transfer-based multi-hop uneven clustering routing protocol for EH-cognitive radio sensor networks. *Big Data Cogn. Comput.* **2024**, *8*, 15. [CrossRef]
27. Laidi, R.; Djenouri, D.; Balasingham, I. On predicting sensor readings with sequence modeling and reinforcement learning for energy-efficient IoT applications. *IEEE Trans. Syst. Man, Cybern. Syst.* **2021**, *52*, 5140–5151. [CrossRef]
28. Gao, A.; Wang, Q.; Wang, Y.; Du, C.; Hu, Y.; Liang, W.; Ng, S.X. Attention enhanced multi-agent reinforcement learning for cooperative spectrum sensing in cognitive radio networks. *IEEE Trans. Veh. Technol.* **2024**, *73*, 10464–10477. [CrossRef]
29. Malik, T.S.; Malik, K.R.; Afzal, A.; Ibrar, M.; Wang, L.; Song, H.; Shah, N. RL-IoT: Reinforcement learning-based routing approach for cognitive radio-enabled IoT communications. *IEEE Internet Things J.* **2022**, *10*, 1836–1847. [CrossRef]
30. Ghamry, W.K.; Shukry, S. Spectrum access in cognitive IoT using reinforcement learning. *Clust. Comput.* **2021**, *24*, 2909–2925. [CrossRef]
31. Liu, X.; Sun, C.; Yu, W.; Zhou, M. Reinforcement-learning-based dynamic spectrum access for software-defined cognitive industrial internet of things. *IEEE Trans. Ind. Inform.* **2021**, *18*, 4244–4253. [CrossRef]
32. Tan, X.; Zhou, L.; Wang, H.; Sun, Y.; Zhao, H.; Seet, B.C.; Wei, J.; Leung, V.C. Cooperative multi-agent reinforcement-learning-based distributed dynamic spectrum access in cognitive radio networks. *IEEE Internet Things J.* **2022**, *9*, 19477–19488. [CrossRef]
33. Hemelatha, S.; Kumar, A.; Manchanda, M.; Manashree, K.G.; Kulkarni, O.S. Cognitive Radio-Enabled Internet of Things Communications: A Reinforcement Learning-Based Routing Method. In Proceedings of the 2024 Global Conference on Communications and Information Technologies (GCCIT), Bangalore, India, 25–26 October 2024; IEEE: Piscataway, NJ, USA, 2024; pp. 1–7.
34. Pham, T.T.H.; Noh, W.; Cho, S. Multi-agent reinforcement learning based optimal energy sensing threshold control in distributed cognitive radio networks with directional antenna. *ICT Express* **2024**, *10*, 472–478. [CrossRef]
35. Ghamry, W.K.; Shukry, S. Multi-objective intelligent clustering routing schema for internet of things enabled wireless sensor networks using deep reinforcement learning. *Clust. Comput.* **2024**, *27*, 4941–4961. [CrossRef]
36. Mukherjee, A.; Divya, A.; Sivvani, M.; Pal, S.K. Cognitive intelligence in industrial robots and manufacturing. *Comput. Ind. Eng.* **2024**, *191*, 110106. [CrossRef]
37. Dvir, E.; Shifrin, M.; Gurewitz, O. Cooperative Multi-Agent Reinforcement Learning for Data Gathering in Energy-Harvesting Wireless Sensor Networks. *Mathematics* **2024**, *12*, 2102. [CrossRef]
38. Matei, A.; Cocoșatu, M. Artificial Internet of Things, sensor-based digital twin urban computing vision algorithms, and blockchain cloud networks in sustainable smart city administration. *Sustainability* **2024**, *16*, 6749. [CrossRef]
39. Al-Quayed, F.; Humayun, M.; Alnusairi, T.S.; Ullah, I.; Bashir, A.K.; Hussain, T. Context-Aware Prediction with Secure and Lightweight Cognitive Decision Model in Smart Cities. *Cogn. Comput.* **2025**, *17*, 44. [CrossRef]
40. Sultan, S.M.; Waleed, M.; Pyun, J.Y.; Um, T.W. Energy conservation for internet of things tracking applications using deep reinforcement learning. *Sensors* **2021**, *21*, 3261. [CrossRef]

41. Bai, W.; Zheng, G.; Xia, W.; Mu, Y.; Xue, Y. Multi-User Opportunistic Spectrum Access for Cognitive Radio Networks Based on Multi-Head Self-Attention and Multi-Agent Deep Reinforcement Learning. *Sensors* **2025**, *25*, 2025. [CrossRef]
42. Tripathy, J.; Balasubramani, M.; Rajan, V.A.; Aeron, A.; Arora, M. Reinforcement learning for optimizing real-time interventions and personalized feedback using wearable sensors. *Meas. Sens.* **2024**, *33*, 101151. [CrossRef]
43. Suresh, S.S.; Prabhu, V.; Parthasarathy, V.; Senthilkumar, G.; Gundu, V. Intelligent data routing strategy based on federated deep reinforcement learning for IOT-enabled wireless sensor networks. *Meas. Sens.* **2024**, *31*, 101012. [CrossRef]
44. Chen, J.; Zhang, Z.; Fan, D.; Hou, C.; Zhang, Y.; Hou, T.; Zou, X.; Zhao, J. Distributed Decision Making for Electromagnetic Radiation Source Localization Using Multi-Agent Deep Reinforcement Learning. *Drones* **2025**, *9*, 216. [CrossRef]
45. Flandermeyer, S.A.; Mattingly, R.G.; Metcalf, J.G. Deep reinforcement learning for cognitive radar spectrum sharing: A continuous control approach. *IEEE Trans. Radar Syst.* **2024**, *2*, 125–137. [CrossRef]
46. Mei, R.; Wang, Z. Multi-Agent Deep Reinforcement Learning-Based Resource Allocation for Cognitive Radio Networks. *IEEE Trans. Veh. Technol.* **2024**. [CrossRef]
47. Canese, L.; Cardarilli, G.C.; Dehghan Pir, M.M.; Di Nunzio, L.; Spanò, S. Design and Development of Multi-Agent Reinforcement Learning Intelligence on the Robotarium Platform for Embedded System Applications. *Electronics* **2024**, *13*, 1819. [CrossRef]
48. Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; Klimov, O. Proximal policy optimization algorithms. *arXiv* **2017**, arXiv:1707.06347.