

Article

Cognitive Adaptive Systems for Industrial Internet of Things Using Reinforcement Algorithm

Anand Singh Rajawat ¹, S. B. Goyal ^{2,*}, Chetan Chauhan ¹, Pradeep Bedi ³, Mukesh Prasad ^{4,*}
and Tony Jan ^{5,*}

¹ School of Computer Sciences & Engineering, Sandip University, Nashik 422213, India

² Faculty of Information Technology, City University, Petaling Jaya 46100, Malaysia

³ School of Computing Science and Engineering, Galgotias University, Greater Noida 203201, India

⁴ School of Computer Science, Faculty of Engineering, I.T, University of Technology Sydney, Sydney 2007, Australia

⁵ Centre for Artificial Intelligence Research and Optimization, Design and Creative Technology Vertical, Torrens University, Sydney 2007, Australia

* Correspondence: sb.goyal@city.edu.my (S.B.G.); mukesh.prasad@uts.edu.au (M.P.);

tony.jan@torrens.edu.au (T.J.)

Abstract: Agile product development cycles and re-configurable Industrial Internet of Things (IIoT) allow more flexible and resilient industrial production systems that can handle a broader range of challenges and improve their productivity. Reinforcement Learning (RL) was shown to be able to support industrial production systems to be flexible and resilient to respond to changes in real time. This study examines the use of RL in a wide range of adaptive cognitive systems with IIoT-edges in manufacturing processes. We propose a cognitive adaptive system using IIoT with RL (CAS-IIoT-RL) and our experimental analysis showed that the proposed model showed improvements with adaptive and dynamic decision controls in challenging industrial environments.

Keywords: 5G; Industrial Internet of Things; D2D; M2M; reinforcement learning algorithms



Citation: Rajawat, A.S.; Goyal, S.B.; Chauhan, C.; Bedi, P.; Prasad, M.; Jan, T. Cognitive Adaptive Systems for Industrial Internet of Things Using Reinforcement Algorithm. *Electronics* **2023**, *12*, 217. <https://doi.org/10.3390/electronics12010217>

Academic Editor: Djuradj Budimir

Received: 2 November 2022

Revised: 9 December 2022

Accepted: 23 December 2022

Published: 1 January 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In recent years, the Industrial Internet of Things (IIoTs) has been fuelling the fourth industrial revolution (IIoT). From early sensor networks to today's NB-IoT, LoRaWAN and LTE Cat M1 [1], IIoT has evolved significantly. Edge Computing [2], with its core components in networking, computers, storage and applications, can provide a platform that can extract critical information and reduces transmission stress. The smart IIoT is designed to encourage users to interact at the edges of the computing network. The IIoT should be able to sense, calculate, determine and communicate using edge intelligence. The range of possible IIoT edge intelligence applications is broad [3]. Through semantic representation, sensor correlation and network-wide AI modelling, IIoT-enabled cognitive technology can improve network awareness and semantic contextual comprehension. Cognitive technologies, however, require high-level situational awareness and still offer challenges to IIoT-enabled edge solutions.

The rapid growth in information science and computing intelligence offers some new solutions to smart edge IIoT applications [4]. In particular, smart edge IIoT has benefited from intelligent computing such as deep learning (DL) which learns intelligent behaviour from available machine data (made available) from the edge devices, such as computers or industrial controllers[5]. Perception, comprehension, learning, judgement, rationality, planning, design and resolution are all parts of DL. The DL with IIoT allows the network to represent, learn and argue. Humans can learn from new data analysis easily, but it is difficult for machines to quickly adjust their knowledge if the input information changes abruptly. Cognitive technology aims to automate and mimic human learning functions. DL used by

the edge gadgets, such as the Nvidia Jetson TX, attempts to simulate human brain operation. Cognitive technology can extract unstructured data from its underlying peripheries. A semantic database with concepts, entities and links was built to provide an actual dataset along with the industrial controller. The machine can abstractly observe the world thanks to cognitive technology. This is helpful for empirically testing a machine-learning model. Through the cognitive data sources from the IIoTs, DL enables network services to react adaptively to external events, just as an individual human would. Subsequently, intelligent applications that go beyond human-level engagements are also conceivable.

Edge Computing provides computer resources at the edge of the network for implementing machine learning algorithms [6,7]. Cognitive technologies enable context awareness at the intelligent IIoT edge. According to the current study, cognitive technology significantly affects user interaction with Edge Computing devices. Cognitive technology can only make decisions based on predetermined criteria and cannot extend to new knowledge easily. With cognitive technology as a starting point, we investigate how, why, where and when the cognitive capabilities of the intelligent IoT edge for the bi-physics system can become available.

In summary, the main contributions of this article are as follows:

- Available machine learning techniques are used to expand the cognitive abilities of the IIoT spectrum. At the edge of the network, we study the cognitive abilities of devices that can look at data from the IIoT and make smart decisions.
- The built-in processing power of Edge Computing is used to show how intelligent IIoTs perform in an environment with considerable network traffics. The core skeleton of Rim computing was used for this purpose. We examined how machine-learning-enabled edge technologies affect Intelligent IIoT edge networks.

In this article, deep reinforcement learning is used to demonstrate how IIoT can be intelligent at the edge. After careful observations of how a reconfigurable production line makes decisions, the machine operation schedule should be optimized based on how the production line orders behave. This case study provided innovative ideas for solving the industrial machinery scheduling problem and moving forward with real-world IIoT applications with edge intelligence.

This study demonstrates how deep learning can enable an intelligent edge-based IIoT. By combining cognitive techniques with ML algorithms, models, data and a coordination mechanism, we can improve the cognitive abilities of smart-edge IIoTs. Dynamic adaptive planning (DAP) for a production line was considered that can be updated using RL. Ideally, the production line can learn and reason within the IIoT environments. In the following sections, we examine related work followed by a discussion of the proposed model in experimental analysis.

2. Related Work

Siafara et al. [8] said that the manufacturing plants in Industry 4.0 are beginning to use a variety of distributed Cyber-Physical Systems (CPSs) both cognitively and practically. It is impossible to keep up with the increasing demands for adaptability, operating speed, efficiency and resilience with traditional all-in-one systems. It is possible to use SAMBA to automate the control and supervision of decentralized CPSs. The data collected on the factory floor were used in this framework, which was related to the Manufacturing Execution System (MES). As conditions and environments change, a system's capacity to respond quickly and intelligently can safeguard its performance.

Li et al. [9] showed that the linear quadratic regulator (LQR) can control systems with more than one rate with the help of matrix replacements. The authors used three methods to make better controllers in multi-rate systems: live policy iteration, off-policy approaches and reinforcement learning. The off-policy approach is changed to a model-free reinforcement learning process using least squares. The only pieces of information this method needed were the input and output.

You et al. [10] showed that once 6G is used, which has higher technical requirements than 5G, there will be no difference between natural and virtual worlds, creating an interesting paradigm for edge intelligence research.

Franco et al. [11] proposed that each manufacturer train the model on their end and then aggregate the models in a central cloud server, which is assembled to reach global optimization and reduce communication cycles. To model situations with more than one assignment, the dataset must be divided into subsets with the same number of devices as assignments. At each local iteration, each device chooses a subset that improves the model. Our early research showed that the method works best with a training dataset with a wide range of factories and devices.

Fenza et al. [12] proposed a cognitive system that can use information learned from past interventions to give targeted suggestions about reducing the amount of time, resources and scope needed for routine maintenance. The system used formal conceptual models, incremental learning and ranking algorithms, among other things, to achieve this goal.

Kolchinsky et al. [13] offered a framework that can be applied to any physical system, whether it is biological or inorganic, because their framework was based on how a system's internal dynamics are related to its environment. In what follows, words like "information value", "semantic content" and "agency", which people have known intuitively to be related to semantic information, are given their formal definitions.

The other key literature studies are summarized in Table 1.

Table 1. Comparison of deep learning models used with IIoT.

Citation	Model/Algorithm	IoT Application	Advantage	Remark
Kegyes et al. [14]	Reinforcement learning (RL)	Development of Industry 4.0 Applications	Describe the Reinforcement learning model for industry 4.0	Theoretical approach
Osifekoet al. [15]	Convolutional Neural Networks (CNN), AI	Cognitive Sensing in Future IoT	Understanding of AI techniques deployed for cognitive sensing	Lightweight algorithms that work well on nodes with little resources.
Chen et al. [16]	Deep reinforcement learning (DRL) algorithms	IoT applications including smart grid, intelligent transportation systems	Industrial IoT applications, mobile crowdsensing and blockchain-empowered IoT.	Need to DRL in IoT application.
Khan et al. [17]	Hybrid Deep Learning Approach	Securing Industrial Internet of Things Against Botnet Attacks	Identifying accurately multi-variant sophisticated bot attacks	Sophisticated risks and cyber-attacks using computational IIoTs and DL-driven workflows.
Latif et al. [18]	Deep Feedforward Neural Networks Restricted Boltzmann Machines (RBM), Deep Belief Networks (DBN)	Deep Learning for the Industrial Internet of Things (IIoT)	IIoT applications	Lightweight Learning Frameworks.

Cyber Physical System (CPS)

CPS is a building block of digital manufacturing that makes use of computers and other related technologies, such as Industry 4.0 communication and intelligence, supporting production to be more efficient and flexible and better for the environment. Smart digital manufacturing can support companies to be more competitive in the long run. Intelligent control of CPS allows reconfigurable industrial applications—which can provide dynamic

and robust solutions—once supported and managed by a machine learning (or deep learning) decision model.

3. Proposed Methodology

RL Learning Paradigm

The RL learning paradigm assumes that only a certain number of possible states and actions exist. When a system is already in state x , action y moves it to the next state, which is another state of x . Real-number vector is often used to describe both x and y . The environment provides the agent with a reward, represented by $R(x, y)$. This reward tells the agent that when in state x , it should perform action y . In the Markov Decision Process, also called MDP, the agent's goal is to maximize the expected value of cumulative discounted rewards as they build up over time by taking the right actions. In this study, discounted payments are used to do two things: (1) get the agent to the goal state as quickly as possible and (2) ensure that the total amount of money paid out is finite. With the discount factor, the extent to which a discount will be applied to future prizes can be changed.

X is the set of all possible states and Y is the set of all possible results of using the suggested approach. $T_{xy}(x)$ represents the probability of moving from one state to another [19]. R represents the reward function and Q represents the set of all possible actions. Most of the time, it is up to the policy function to turn state action pairs into real numbers ($R : XYR$), whereas the reward function R is in charge of doing the exact opposite ($R : XYR$). In some programs, you can get a reward even if you did not do what you were supposed to ($R : XR$). The value function V is the expected sum of discounted rewards and is defined by a particular initial state and a policy that has already been decided. The equation for Y (refer to Equation (7)) shows the procedures that are carried out and they are carried out according to the rules that are in place when the system is in its current operating state (x, y) . The job of the value function is to give each state a value that is specific to that state. The concept of a value function is what the RL paradigm proposed. Determining the policy is sometimes more complex than calculating the value function. Therefore, the value function provided by RL is used to determine the policy function. In this section, we discuss one of these value functions.

$$V^\pi(x) = E\left(R(x_0) + \gamma R(x_1) + \gamma^2 R(x_2) + \dots | x_0 = x, \pi\right) \quad (1)$$

As shown in Equation (1), the discount factor $[0, 1]$ makes the present reward more important while making the future reward less critical. From the point of view of RL, the end goal is to come up with the best policy that will help earn the most discount rewards possible. When the best policies are implemented, the value functions that reach their full potential are called optimal value functions.

$$V^*(x) = \max_{\pi} V^\pi(x) \quad (2)$$

The Bellman equation for the optimal value function is given by Equation (2). According to this formula, state p 's immediate reward $R(x)$ and the discounted maximum predicted rewards from state p 's next state p 's cumulative discounted rewards are the sum of these two sums. This is an example of a stochastic process in which the next state can only be reached by chance. We can find the value of a state by using the Bellman Equation and moving forward in time one prediction step at a time. Using these numbers, we can find which situations are best. When we use a Markov Reward Process, on the other hand, the way we move from one state to the next is completely random. We will need to take steps and switch to a Markov Decision Process to reach this goal. Ultimately, decisions can only be made based on past decisions. Because of this, the player can choose both the next state to visit and the prize that comes with it. In a stochastic system, the expected result or outcome might not always happen. Now, both the reward function and the likelihood of

the next transition depend on the one that just happened with the policy determining the starting status. This means that the best decisions can be made in any future situation.

$$V^*(x) = R(x) + \gamma \max_{y \in Q} X_{xy}(x') V^*(x') \tag{3}$$

A non-deterministic system has no probabilities at all, except for the chance that it will change from one state to another (for which the probability is 1). The Bellman’s equation in Equation (1) describes an ideal policy in a deterministic setting. This type of policy is optimal because it maximizes the future discounted rewards. According to Equation (3), the best thing to do in state p is to give you the most expected discounted cumulative rewards in the next state p .

$$V^*(x) = R(x) + \gamma \max_{y \in Q} V^*(x') \tag{4}$$

In the abbreviated form of Bellman’s (1957) equations in Equation (3), the Value Iteration and Policy Iteration algorithms were employed. For these algorithms to operate, the state and action space in MDP must be finite. Value function states are set to 0 in the initial stage of the process for iterating over values. For each state, the value function is found by retracing the steps in Equation (2)’s Bellman’s Equation. A value function can be updated in two ways. A value function can be updated in various synchronous and asynchronous ways. Each state’s new value function is computed first in the synchronous technique of updating. There are no additional functions left to run. If value functions are updated asynchronously, new values are applied immediately after the old ones are updated. New value functions are employed in Equation (4) to determine the optimum course of action. The algorithms for the value iteration method are shown in Equations (5)–(7). First, a random collection of policies is generated in a policy-iteration process. After determining the policy’s value function, the policy is tested and evaluated.

$$\pi^*(x) = arg \max_{y \in Y} \sum_{x' \in X} X_{xy}(x') V^*(x') \tag{5}$$

$$\pi^*(x) = arg \max_{y \in Y} V^*(x') \tag{6}$$

This category is for RL algorithms that use value iteration to solve problems without a model. It is possible to update the Q -function online, by detecting the output of the following state change: As a result, an update to the Q -function can be performed as shown in the following example: $x_{k+1}, y_k = 1$

$$Y_{k+1}(x_k, y_k) = Y_k(x_k, y_k) + \alpha [r_{k+1} + \gamma \max_{y'} Y_k(x_{k+1}, y') - Y_k(x_k, y_k)] \tag{7}$$

Equation (7) states that r_{k+1} is the reward or return obtained by the intelligent reinforcement learning agent. This reward or return has a discount factor and learning rate. The dynamic range of the learning agent was between 0 and 1. When both the state space and action space are broken up into a finite number of steps, the Q function converges to its best value, Q , after many iterations. Equation (8) is used to determine the policy function based on the Q -function.

$$\pi^*(x) = arg \max_{y \in Y} V^*(x') \tag{8}$$

In this type of RL, agents take action by either using their action space or looking into it. The word “exploitation” refers to using the knowledge and skills you have gained from your past experiences. When we talk about “exploration”, on the other hand, we mean the process of doing something that has never been done before. Exploration and extraction will always cost money in one way or another. The “exploration vs exploitation conundrum” is a well known idea in the field of learning and development that has to do

with the real world. It was called the “soft-max action selection approach” and it worked well to solve the “exploration-exploitation conundrum”.

4. ML-Enabled Framework of Edge Intelligent IIoT

Utilizing cognitive IIoT networking technologies such as perceptual control, network communications and information technology, it is feasible to establish information links between physical and virtual worlds.

Edge intelligent IIoT and DL techniques use state-of-the-art production, network cooperation, customization and extension in service offerings. The perception, transmission and application layers of the DL-enabled IIoT framework are illustrated in Figure 1.

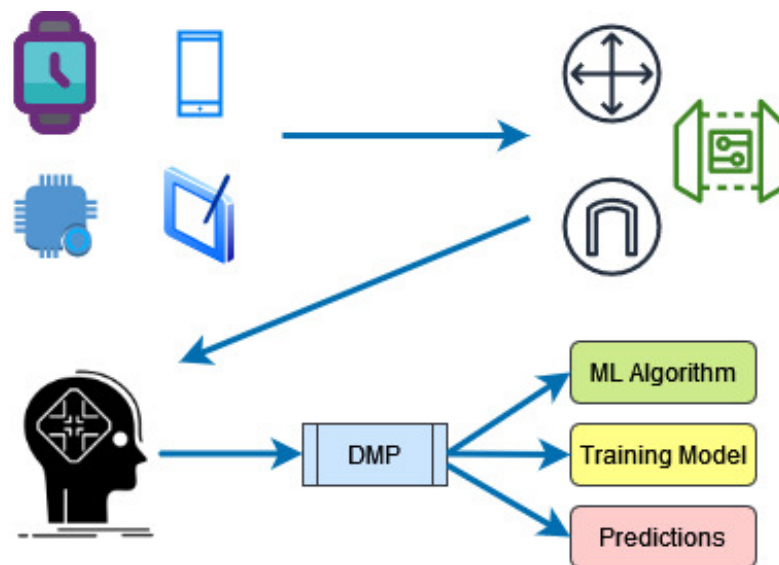


Figure 1. The ML-enabled framework of cognitive IIoT.

4.1. Perceptual Layer

During the life cycle of an industrial process, raw data can be collected through sensors or radio frequency identification. The perceptive layer serves as the data source and foundation for an intelligent plant to achieve its maximum potential use. In order to collect data in the perceptual layer, we use a Programmable Logic Controller (PLC) as well as a monitoring system and a distributed monitoring system. Therefore, it is possible to link the physical form of the field equipment with its quality (e.g., an industrial robot). Edge Computing is utilized to facilitate the transition from a centralized control model to one that is dispersed to solve this issue. RESTful web nodes can provide application programming interfaces (APIs) for field devices if they use RESTful web services (Representative State Transfer). Utilizing movable sensor nodes is one method that can be utilized to improve the mobility of the perceptual stratum [10].

4.2. Transmission Layer

The transmission level incorporates industrial Ethernet, a technology for wireless communications that is used in the short term and a vast network that uses very little power. NB-IoT, LoRaWAN, LTE CAT M1 and cellular 5G are technologies currently used as the standard transmission layer technologies. Regarding applications in the industrial sector, the Internet of Things (IoT) has always met stringent criteria, such as high dependability with low power consumption and high levels of security. On the other hand, sample transfer can be required if a large amount of the raw data is lost. Because of these discoveries, an intelligent IIoT aircraft must include a cognitive science-based and knowledge-based aircraft. Installing Raspberry Pi-based Raspberry edge nodes on a solid foundation is made possible by intelligent IoT-enhanced edge services and applications. Cognitive models are used to describe data related to the IIoT in terms of its cognitive standards and needs. This

new paradigm can potentially eliminate data ambiguity caused by numerous protocols, allowing for semantic data collection.

4.3. Perceptual Layer

The IIoT data were analyzed, computed and gathered in the application layer. Optimization: The development of intelligent apps relies heavily on DL techniques in the IIoT. A.I. models may be better built and validated using the DL theory, which relies on simple network resources. The need for a precise representation of a single cognitive technology has also been addressed. DL approaches are well supported in intelligent IoT applications (e.g., pattern identification, precise modeling, information processing, thinking and decision-making). A realistic resource planning approach for IIoT services can be achieved through intelligent applications. In connection with Pattern Identification (PI) and Precise Modelling (PM) information Processing (IP), we arrived at the Thinking and Decision-Making Process (DMP) that required the following attributes:

The interactive functions of influence ($F(k_i)$) have described elements of personalization and D.M.P. has been studied based on these hypotheses and the decision-maker model shown in Figure 2.

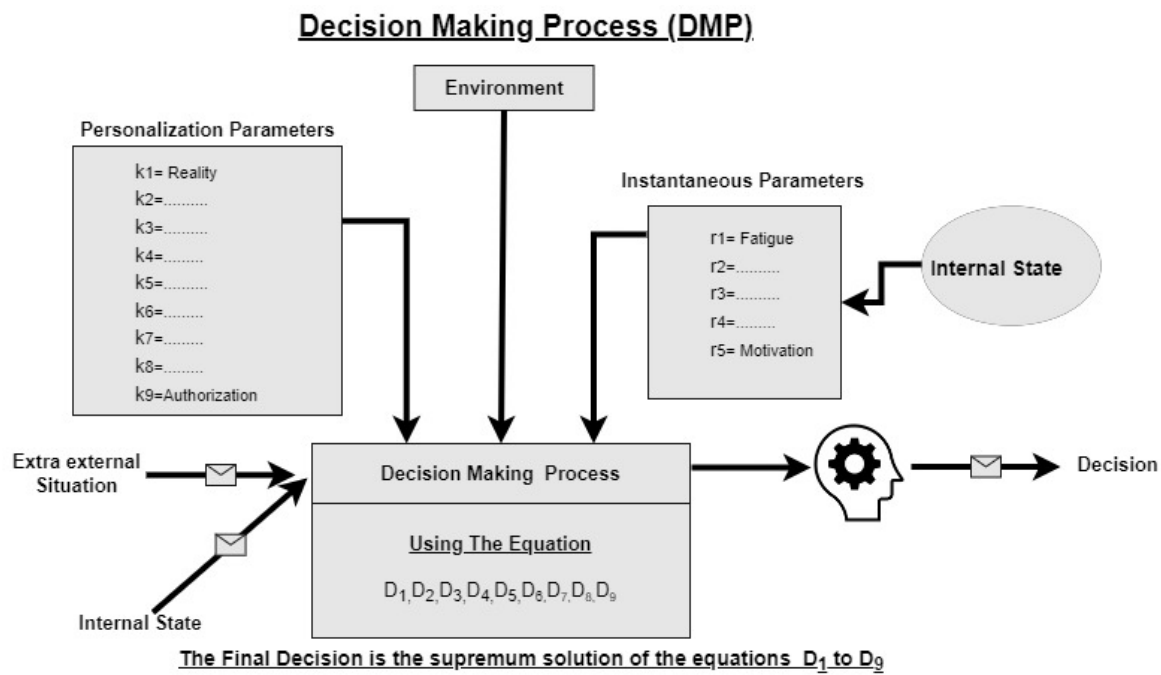


Figure 2. Decision-making process.

Concerning the decision-making process, a close examination of the correlations between the functions of degradation ($I(R.I.)$) and the rapid state was also completed.

In the next section, we attempt to formulate human behavior attributes into mathematical formulas utilizing artificial states: Two functions are used to formulate the concepts; function F is a polynomial function of one or several variables. The functions I are also polynomial. We utilized a two-dimensional polynomial with varying degrees of complexity and variable seeding. There are several ways to represent the whole of the processes. The polynomial function, denoted by the letter F , is one that is dependent on one or more variables. The polynomial function is also used to represent I . Assuming that each variable in personalization and instantaneous state is independent, the functional relationships of $F(k_n)$ and $I(r_m)$ are expressed by the following attributes.

$$k_1 = \text{Reality}$$

$k_2 = \text{knowledge}$

$k_3 = \text{Relation to time}$

$k_4 = \text{Expensiveness}$

$k_5 = \text{Ego}$

$k_6 = \text{Creativity}$

$k_7 = \text{Risk}$

$k_8 = \text{Anxiety Level}$

$k_9 = \text{Authorization}$

Rapid state attributes are as follows:

$r_1 = \text{Fatigue}$

$r_2 = \text{Integration}$

$r_3 = \text{Sleep deprivation}$

$r_4 = \text{Tiredness}$

$r_5 = \text{Morality}$

$r_6 = \text{Motivation}$

Using the above attributes, the equation can be formulated for each state as follows;
Irrational Behavior State:

$$D_1 = F(k_1, k_2, k_5, k_8) + I(r_1, r_5) \quad (9)$$

Observation of what we want to state:

$$D_2 = F(k_1, k_2, k_3, k_4, k_7, k_9) + I(r_1, r_2, r_5) \quad (10)$$

Possessions State:

$$D_3 = F(k_1, k_2, k_3, k_4, k_7, k_8, k_9) + I(r_2, r_3, r_4, r_5) \quad (11)$$

Losing sight of our Main Goal state:

$$D_4 = F(k_1, k_2, k_3, k_5, k_7, k_8) + I(r_2, r_4, r_5) \quad (12)$$

Expectations from our mindfulness state:

$$D_5 = F(k_3, k_4, k_5, k_7, k_9) + I(r_2, r_4, r_5) \quad (13)$$

Performance state:

$$D_6 = F(k_3, k_4, k_7, k_8) + I(r_1, r_4, r_5) \quad (14)$$

Untrustworthy state:

$$D_7 = F(k_1, k_2, k_3, k_8) + I(r_2, r_4) \quad (15)$$

Natural state:

$$D_8 = F(k_3, k_4, k_7, k_8) + I(r_4, r_5) \quad (16)$$

No cost state:

$$D_9 = F(k_5, k_7, k_8) + I(r_1, r_3) \quad (17)$$

5. ML Methods for Improving the Cognitive Ability of Edge Intelligent IIoT

There is a terminal for every step of industrial development in the IIoT array of terminals. Mobile communication terminals, all-in-one computer terminals and environmentally sensitive terminals are a few options available today.

The Intelligent IIoT connects low-cost sensors and smart distributed terminals to modern computers by making these devices more popular. Edge Smart IIoT can transport cloud services with low latency, high bandwidth and low jitter, as illustrated in Figure 3. We may be able to detect changes in industrial circumstances if we ensure that the data we collect makes sense. The intelligent IIoT border's intelligent Deep Learning (ML) approaches make the entire IIoT system integrated, easy to comprehend and complete.

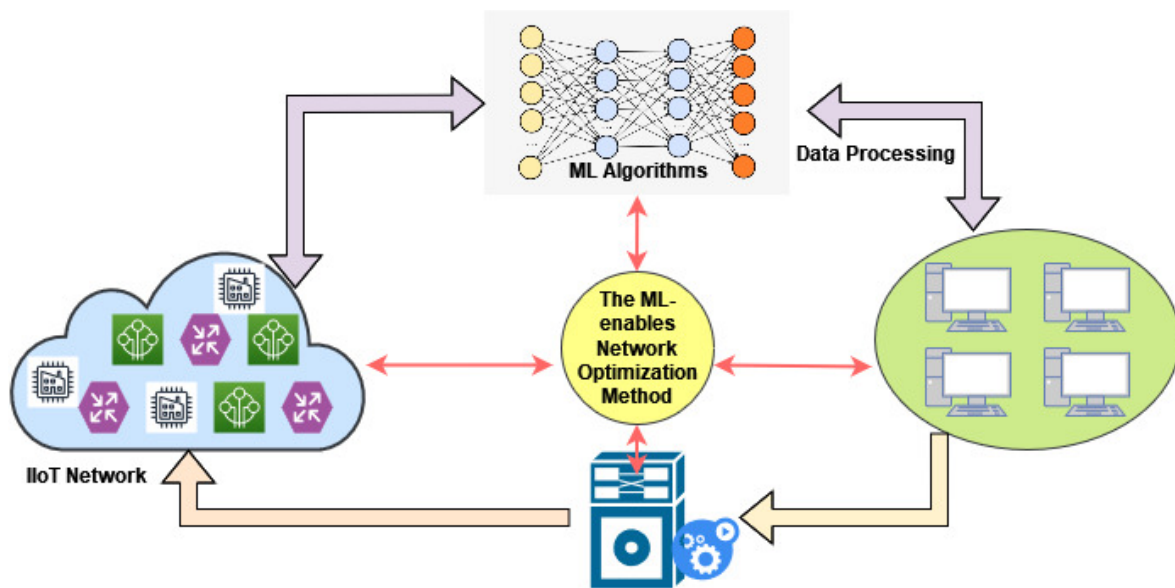


Figure 3. ML-enabled network optimization method.

Machine learning algorithms are essential tools that may improve decision-making precision in real-time industrial data processing and offline training. Intelligent IIoT applications are the basis upon which a robust machine-learning model can be constructed. It is essential to gain a deep understanding of the algorithms behind machine learning. Deep learning (DL) is one of the most common methods, along with improved study and profound enhancements.

It is possible to teach a machine how to represent data through a process called machine learning. The development of a neural network in DL allows the study and comprehension of the activity that occurs in the human brain [20]. When it comes to determining functions and fitting models, DL offers many benefits that are difficult to overcome. Mathematical models can easily handle large datasets owing to their inherent flexibilities. The value of a precise Smart IIoT model cannot be overstated. RL is an interactive decision-making learning system. RL is a more alluring model for learning because it is more in line with human behavior and psychology than other learning models. It can map outputs that are dependent on inputs that can be either a single input or multiple inputs simultaneously (e.g., the Markov Decision Process). RL was a revolutionary method to achieve remarkably accurate results in allocating IIoT content-centered services. It is possible to use RL to alter the status of the environment concerning the activities that can be carried out in the condition that is currently present. Learning occurs through observing and comparing different pairs of operational states in their respective environments (or creating a state-operation pairs table). It is utilized to calculate monetary incentives for state-operating partnership programs. In order to accomplish the desired result for the environment, the most well known activity is selected.

As a result of the presence of DL, the player is no longer confined to a small region, which results in increased levels of comfort and mobility. As a result, deep reinforced learning (DRL) is most likely to be utilized in an IIoT scenario that involves multiple dimensions and numerous components. Mao et al. [21] established DRL to address the challenges caused by global cluster planning. During the experimental phase of the project, dynamic adaptive routes for reconfigurable DRL lines were put through their paces.

5.1. Data-Driven Learning and Reasoning

As the amount of IIoT data increases, so does the need for more advanced data mining techniques [22]. DL is able to better utilize large volumes of data than the previous methods. The typical IoT technologies utilized in intelligent IIoT applications can only partially represent industrial datasets. Instead of relying on specialized knowledge, DL uses data that accurately portrays the underlying problem. Consequently, the mathematical model is more detailed than expert systems. Most IIoT ML systems operate in two steps: learning and reasoning. The proposed model weights and partial bias were both built using data from the training sets. The proposed model was evaluated using a suitable method. In order to generate an evidence-based forecast, the reasoning is the process of gathering information and spotting occurrences. As a general rule, models are judged on the basis of the accuracy of their reasoning. The DL model design approach is schematically illustrated in Figure 4. Through extensive data training on the verification set, a candidate model with an excellent match was selected and avoided in the cross-validation. As a result, the model's ability to generalize was enhanced. In the actual world, too, prior knowledge is necessary.

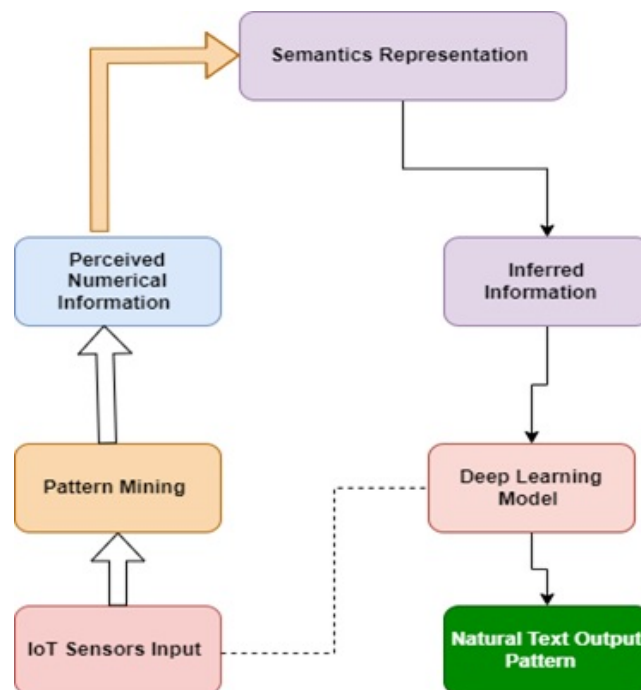


Figure 4. Flow chart of the proposed approach.

Data sets generated by Cognitive IIoT can be utilized to make intelligent inferences, predictions and decisions when external variable change. Tagging, semantic tagging and abstract functionality are some of the cognitive IIoT network transfer functions that can be used. Typical IIoT applications can benefit from online and offline training (e.g., active operation and maintenance). Data-driven reasoning and prediction create intelligent decisions based on learning principles. Predictive testing and enhanced data modeling can now be achieved by moving ordinary Industrial Internet of Things (IIoT) sensors closer

to the edge. In order to reduce environmental turbulence, application models may gather data at the network edge and then refine their models.

5.2. Coordination with Cognitive Methods

The development of intelligent capabilities for the Internet of Things (IoT) can be facilitated through machine learning and cognitive processes. Utilizing semantic perception technology allows for automatic search, localization and access to sensitive equipment (e.g., ontology). Semantic technology makes it feasible to recreate real-world items and correctly recognize electronic devices. It is possible to engage with intelligent edge IIoT sources or cooperate with them by using cognitive techniques based on semantics. Intelligent devices can be modeled more efficiently using logical, semantic and computational modelling methodologies. The production equipment used by the IoT has been simplified because of its own decisions to experience the world. However, it is very challenging for them to think outside of the parameters they have been given. Within the cognitive sphere, the author of the codes holds the final say. It is necessary to have a deep understanding of the surrounding environment and a significant amount of free time to successfully perform the resource management task [23]. An insightful study of data from the IIoT indicates that machine learning is unsuccessful. Deep Learning techniques, when combined with cognitive approaches, have the potential to build accurate semantic analysis models. This model can carry out autonomous ideas and behaviours that affect their surrounding environment. It is much simpler for Deep Learning to comprehend the perceptual environment when employing cognitive semanticization tactics. The gathering and modification of sensory inputs, idea identification and problem finding are the three main components of cognitive techniques known as information processing. Cognitive coping skills can be highly beneficial when confronted [24] with information sources that are inconsistent, erroneous or otherwise difficult to understand. In machine learning, abstraction is an essential concept. In the end, it boiled down to the following:

1. Gaining a new perspective on cognition
2. Abstraction of the underlying information
3. Dynamic human programming

Because the vast majority of IIoT data are stored in a time series, more advanced machine learning algorithms may be able to generate predictions by drawing on previously collected information (e.g., predictive maintenance of equipment). CAS-IIoT-RL can be utilized in dynamically adaptive scheduling, adjusting system parameters and optimizing measures, such as removing the network status from the IIoT. The CAS-IIoT-RL technique offers large-scale IIoT monitoring because it does not rely on a sample node or specialized field knowledge. An accurate machine learning model is essential to construct an intelligent service for edge IIoT. It is challenging to obtain realistic test sets closer to the virtual environment of an IIoT network. It may be simpler to get around the obstacles an unauthorized machine learning model presents. Learning that is embedded in machines employs models that are more specific in order to interact with domain experts. The next subsection examines the planning that is Dynamic and Adaptive for a reconfigurable production line.

Because the market demands customization of several products, intelligent organizations frequently need to respond quickly to changes in the orders they receive. In order to produce diverse products using a reconfigurable production line, a DAP is required for the processing process along that line. Currently, four grasping robots, one packing robot and one stock robot are operating on our prototype line. A method for describing anything is to use a test scenario. Orders for candies can be placed on the Internet. Depending on the tastes of individual customers, confections can be purchased in various forms and dimensions. Because each robot receives a unique candy flavor, the production line needs to adjust the path that the candy takes to be packaged.

We know that a candy packaging line features typical multi-packing characteristics. These characteristics can be observed for a variety of sizes. It is not easy to schedule a candy packaging line because there are frequent shifts in the order of importance of different

tasks. This is because the production process for many goods can be customized. Essential supplies are typically in short supply, but many categories of material stock are currently abundant. CAS-IIoT-RL is a clever method that combines the sensitive nature of DL with the decisiveness of RL in order to get optimal results. Depending on how the conditions of the packaging line are perceived, a CAS-IIoT-RL model achieves the aim with the maximum amount of incentives possible. This learning aimed to develop a more effective strategy for context-based ML planning.

5.3. CAS-IIoT-RL Model for Dynamic Adaptive Planning

Figure 4 shows the RL model that takes into account the environment. The order data, material inventories and workload of one engine were considered, taking the model's current state into account. The inputs showed different changes that occurred in the environment. The output of dynamic planning for CAS-IIoT-RL is an adaptable production line that is used as an input for dynamic adaptive planning. When determining the value of our solution, we examined several factors, such as shorter total completion times, less energy use and better use of equipment. We used utility value analysis to find a strategy that would allow us to accurately and precisely change the CAS-IIoT-RL model. When a decision was made regarding where the road would go, the intelligent agent changed how the candy packing line worked. Even though the status of the packing line and the intelligent agent's utility value remained the same, the RL structure for the packing line maintained order quantity decisions and planned decisions separately. CAS-IIoT-RL was used to estimate constant spaces and these qualities were considered. A condensed version of high monitoring data, a condensed version of high-dimensional monitoring data, was made so the packaging process could be recreated. Because CAS-IIoT-RL model training is difficult, a CAS-IIoT-RL context-sensitive model was developed for Raspberry Pi. Order information, current stock level and amount of work done by the unit are three standard inputs for edge intelligent IIoT. Rewards were provided based on how long it took to finish, how much energy it used and how well it was used. These factors were used by the cognitive agents [25–28] in our previous work to determine the location of the product line. The packing line with sensors always acts as the “intelligent actor”. With the help of Edge Intelligent IIoT, we were able to find data that would make it possible to focus on the workplace. Massive data sets from production clouds were used to train the CAS-IIoT-RL model offline. The model was improved by the new data we obtained and the things we learned from research on the Internet.

6. Experimental Results

We tested both ML and Cognitive Adaptive Systems for IIoTs using the Reinforcement algorithm (CAS-IIoT-RL) adaptive planning on the platform for the prototype. We acted like consumers by simulating their behavior by placing orders on the customer side of the production chain. We found that the increase in the order amount caused CAS-IIoT-RL planning to have a larger effect on the production line. Figure 3 shows that we took separate measurements to determine how long it took to finish an order, how much energy the production line used and how often the equipment was used. The CAS-IIoT-RL is a the Static Algorithm [29]. When the order size was small, it took much work to tell the difference between the three stages. Figures 5 and 6 show that when the number of orders was increased to 1900, the static system and CAS-IIoT-RL mechanism worked better than the centralized scheduling system. This was true in terms of both the amount of time and power used. The CAS-IIoT-RL mechanism improved as the number of orders increased. Figure 7 shows how the production line's equipment [29] uses ratio changes over time.

In IIoT applications, many events require real-time decision processing. These events are; however, often highly complex and sparse, lacking information for decisions on many occasions. DL/ML can support such decision processing. In this experimental analysis, each event occurrence was added to one of the data sets. Among the collected data, 70 percent of event-occurrence data were used for training the DL/ML models, 15 percent

for verification and 15 percent for testing. Once the ML/DL model error output was acceptably small, each ML/DL model was compared for its overall performance as part of the IIoT automation applications. Figures 5–7 show:

- The amount of time it takes to complete the task.
- The amount of energy that is used.
- The amount of variation in the amount of equipment that is used along the manufacturing line.

Figures 5–7 indicate that as the number of tasks increased, the proposed CAS-IIoT-RL showed less time, energy and utility variations compared to the other comparable models.

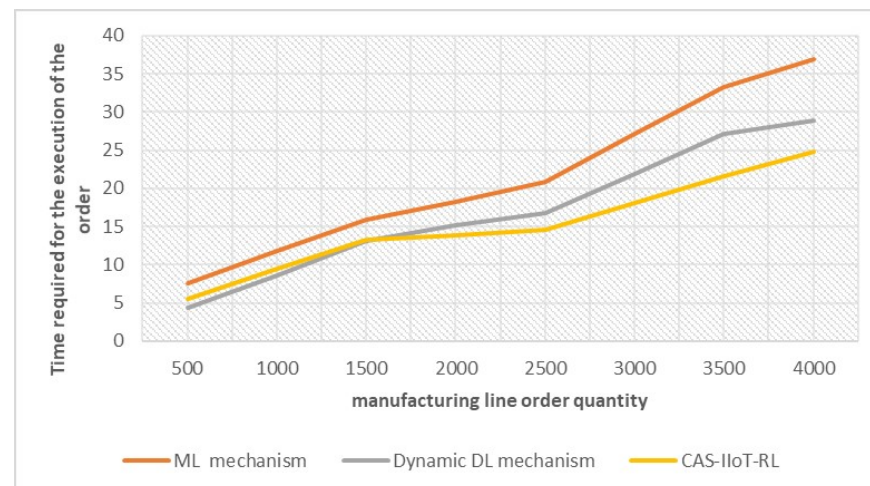


Figure 5. Evaluation of time (seconds) required to complete the task.

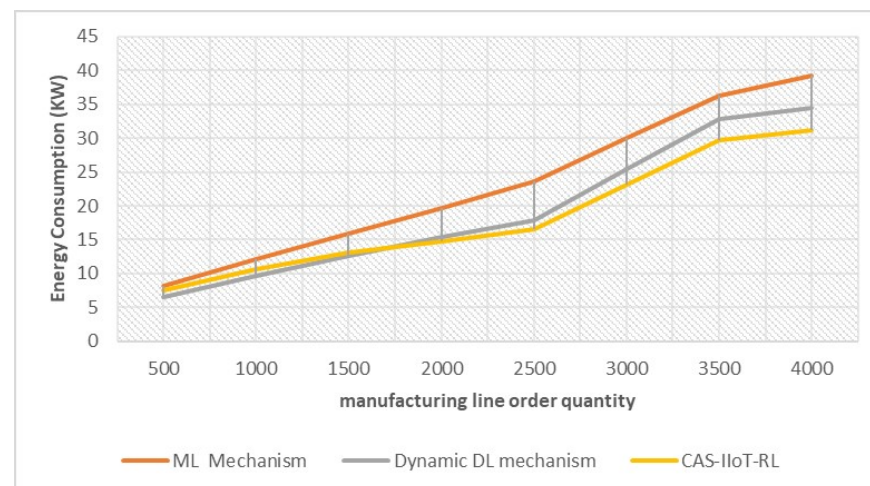


Figure 6. Evaluation of energy required to complete the task.

When utilizing the ML approach, carrying out the order before anything else was common practice. The DL mechanism analyzed how long it would take to complete the task and how effectively various production lines and pieces of machinery collaborated. CAS-IIoT-RL has the cognitive capacity to consider additional factors, such as order behavior, stock and strain placed on a single machine. Because of this, it is at the forefront of intelligent IIoT. It was crucial to look at the machinery's condition, how it was utilized and how much of it was in stock when determining whether it was functioning. In the end, CAS-IIoT-RL gained new knowledge regarding the operation of the production chain. The CAS-IIoT-RL mechanism, which is CAS-IIoT-RL, has made significant progress due to the creation of the Deep Learning model.

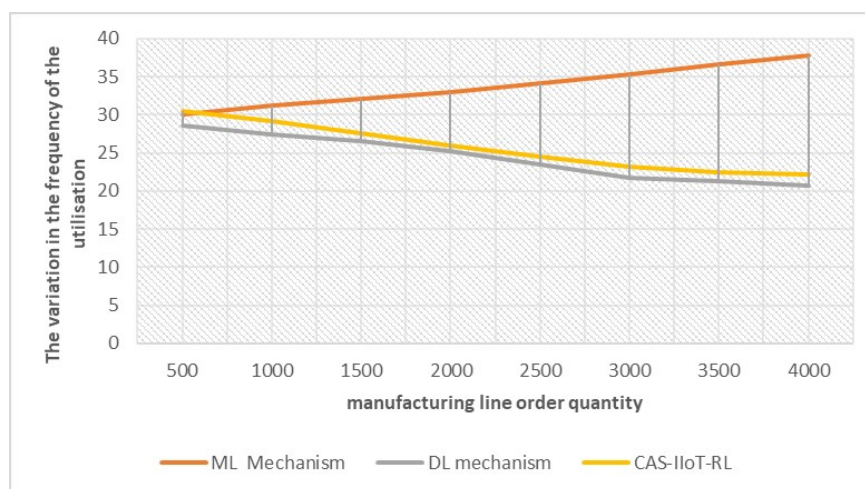


Figure 7. Evaluation of equipment variation used along the manufacturing line.

7. Conclusions

This paper investigated the application of computer-assisted edge cloud strategies to optimization planning. Combining historical scheduling information with cloud server information allowed the optimization of the future performance of the intelligent edge IIoT. The cognitive knowledge provided by the edge IIoT enabled the observation of diverse industrial environments. We provided DL-based optimization methods that assist in strengthening the cognitive capability of the IIoT intelligent edge as the network services became sensitive to data. In order to conduct a context-aware and exploratory test of the multi-product customization production line, DAP based on DRL was employed. The complex problem of juggling several duties was addressed. The experimental outcomes showed that the proposed CAS IIoT-RL model required less time, energy and utility variations compared to the other comparable models (refer to Figures 5–8).

Cognitive IoT was able to make use of the one-of-a-kind characteristics of social networks to extract the most value possible from a network and ensure that it functions to its full potential. In this paper, we demonstrated a socially aware, enhanced D2D communication network model for the cognitive Internet of Things (IoT). This model made use of information pertaining to social orientation. This model considered the fact that various Internet of Things devices had varying requirements for the quality-of-service (QoS) that they required, ranging from ultra-reliable and low-latency communications to a minimum data rate. In order to accomplish this, we presented the optimization problem as a multi-agent reinforcement learning formulation and offered a new coordinated multi-agent deep reinforcement learning-based resource management approach to optimize the combined radio block assignment and transmission power control strategy. This is done by describing the optimization problem as a multi-agent reinforcement learning formulation.

Transferrable DL models are yet to reach a level of maturity that allows them to satisfy the requirements of demanding industrial applications. More data are essential for developing intelligent edge applications for IIoTs. This paper investigated the possibilities presented by integrated intelligent IIoT cognitive technologies as its primary objective. The planning strategy for Edge Computing and cloud computing can be improved. In order to better the process of future development, data from the prior timing are integrated with data from the cloud server.

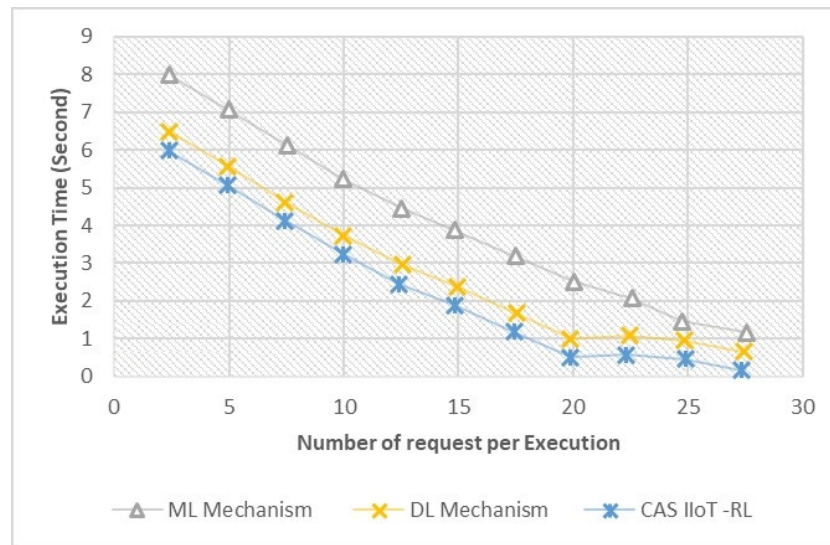


Figure 8. Comparison of ML, DL and CAS IIoT-RL.

Cognitive Adaptive Systems for the IIoTs using reinforcement algorithms should address IoT data, security and system vulnerabilities appropriately. Even small security improvements would greatly encourage service adoption of IIoT-driven edge intelligence. We can achieve pervasive connectivity by extending IIoT combined with smart cloud service and security models.

Author Contributions: Conceptualization, Writing—original draft A.S.R. and S.B.G.; Supervision, S.B.G.; Validation, C.C. and P.B.; propose the new method or methodology, A.S.R. and S.B.G.; Formal Analysis, Investigation, P.B.; Resources, T.J. and M.P.; Software, C.C. and P.B.; Writing—review & editing, S.B.G., T.J. and M.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Zhandos, K.; Ilya, J. Adaptive Supply Chain: Demand-Supply Synchronization Using Deep Reinforcement Learning. *Algorithms* **2021**, *14*, 240. [\[CrossRef\]](#)
- Marx, R.; Chitra, A. Extractive Document Summarization Using an Adaptive, Knowledge Based Cognitive Model. *Cogn. Syst. Res.* **2019**, *56*, 56–71. [\[CrossRef\]](#)
- Coralie, M. Adaptive Early Classification of Temporal Sequences Using Deep Reinforcement Learning. *Knowl.-Based Syst.* **2019**, *190*, 105290. [\[CrossRef\]](#)
- Alhasnawi, B.N.; Jasim, B.H. Internet of Things (IoT) for smart grids: A comprehensive review. *J. Xi'an Univ. Archit* **2020**, *63*, 1006–7930.
- Chen, B.; Wan, J.; Lan, Y.; Imran, M.; Li, D.; Guizani, N. Improving cognitive ability of edge intelligent IIoT through machine learning. *IEEE Netw.* **2019**, *33*, 61–67. [\[CrossRef\]](#)
- Udayakumar, K.; Ramamoorthy, S. Intelligent Resource Allocation in Industrial IoT using Reinforcement Learning with Hybrid Meta-Heuristic Algorithm. *Cybern. Syst.* **2022**. [\[CrossRef\]](#)
- Sulimani, H.; Sajjad, A.M.; Alghamdi, W.Y.; Kaiwartya, O.; Jan, T.; Simoff, S.; Prasad, M. Reinforcement optimization for decentralized service placement policy in IoT-centric fog environment. *Trans. Emerg. Telecommun. Technol.* **2022**, e4650. [\[CrossRef\]](#)
- Siafara, L.C.; Kholerdi, H.; Bratukhin, A.; Taherinejad, N.; Jantsch, A. SAMBA -an architecture for adaptive cognitive control of distributed Cyber-Physical Production Systems based on its self-awareness. *Elektrotech. Inftech* **2018**, *135*, 270–277. [\[CrossRef\]](#)
- Li, Z.; Xue, S.R.; Yu, X.H.; Gao, H.J. Controller Optimization for Multirate Systems Based on Reinforcement Learning. *Int. J. Autom. Comput* **2020**, *17*, 417–427. [\[CrossRef\]](#)
- You, X.; Wang, C.X.; Huang, J.; Gao, X.; Zhang, Z.; Wang, M.; Huang, Y.; Zhang, C.; Jiang, Y.; Wang, J.; et al. Towards 6G wireless communication networks: Vision, enabling technologies and new paradigm shifts. *Sci. China Inf. Sci* **2021**, *64*, 110301. [\[CrossRef\]](#)
- Franco, N.; Van, H.M.; Dreiser, M.; Weiss, G. Towards a Self-Adaptive Architecture for Federated Learning of Industrial Automation Systems. In Proceedings of the 2021 International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS), Madrid, Spain, 18–24 May 2021; pp. 210–216. [\[CrossRef\]](#)

12. Fenza, G.; Gallo, M.; Loia, V.; Marino, D.; Orciuoli, F. A Cognitive Approach based on the Actionable Knowledge Graph for supporting Maintenance Operations. In Proceedings of the 2020 IEEE Conference on Evolving and Adaptive Intelligent Systems (EAIS), Bari, Italy, 27–29 May 2020; pp. 1–7. [\[CrossRef\]](#)
13. Kolchinsky, A.; Wolpert, D.H. Semantic information, autonomous agency and non-equilibrium statistical physics. *Interface Focus* **2018**, *8*, 20180041. [\[CrossRef\]](#) [\[PubMed\]](#)
14. Keyges, T.; Süle, Z.; Abonyi, J. The Applicability of Reinforcement Learning Methods in the Development of Industry 4.0 Applications. *Complexity* **2021**, *2021*, 7179374. [\[CrossRef\]](#)
15. Osifeko, M.O.; Hancke, G.P.; Abu-Mahfouz, A.M. Artificial intelligence techniques for cognitive sensing in future IoT: State-of-the-art, potentials and challenges. *J. Sens. Actuator Netw.* **2020**, *9*, 21. [\[CrossRef\]](#)
16. Chen, W.; Qiu, X.; Cai, T.; Dai, H.N.; Zheng, Z.; Zhang, Y. Deep reinforcement learning for Internet of Things: A comprehensive survey. *IEEE Commun. Surv. Tutor.* **2021**, *23*, 1659–1692. [\[CrossRef\]](#)
17. Hasan, T.; Malik, J.; Bibi, I.; Khan, W.U.; Al-Wesabi, F.N.; Dev, K.; Huang, G. Securing Industrial Internet of Things against botnet attacks using hybrid deep learning approach. *IEEE Trans. Netw. Sci. Eng.* **2022**. [\[CrossRef\]](#)
18. Latif, S.; Driss, M.; Boulila, W.; Jamal, S.S.; Idrees, Z.; Ahmad, J. Deep Learning for the Industrial Internet of Things (IIoT): A Comprehensive Survey of Techniques, Implementation Frameworks, Potential Applications and Future Directions. *Sensors* **2021**, *21*, 7518. [\[CrossRef\]](#) [\[PubMed\]](#)
19. Buchholz, V.; Kopp, S. Towards an Adaptive Assistance System for Monitoring Tasks: Assessing Mental Workload using Eye-Tracking and Performance Measures. In Proceedings of the 2020 IEEE International Conference on Human-Machine Systems (ICHMS), Rome, Italy, 7–9 September 2020; pp. 1–6. [\[CrossRef\]](#)
20. Buchholz, V.; Kopp, S. Towards Adaptive Worker Assistance in Monitoring Tasks. In Proceedings of the 2021 IEEE 2nd International Conference on Human-Machine Systems (ICHMS), Magdeburg, Germany, 8–10 September 2021; pp. 1–4. [\[CrossRef\]](#)
21. Mao, H.; Alizadeh, M.; Menache, I.; Kandula, S. Resource management with deep reinforcement learning. In Proceedings of the 15th ACM Workshop on Hot Topics in Networks, Atlanta, GA, USA, 9–10 November 2016; pp. 50–56.
22. Siafara, L.C.; Kholerdi, H.A.; Bratukhin, A.; TaheriNejad, N.; Wendt, A.; Jantsch, A.; Treytl, A.; Sauter, T. SAMBA: A self-aware health monitoring architecture for distributed industrial systems. In Proceedings of the IECON 2017-43rd Annual Conference of the IEEE Industrial Electronics Society, Beijing, China, 29 October–1 November 2017; pp. 3512–3517. [\[CrossRef\]](#)
23. Petrenko, S. *Developing a Cybersecurity Immune System for Industry 4.0*; CRC Press: Boca Raton, FL, USA, 2022.
24. Petrenko, S. *3 Trends and Prospects of the Development of Immune Protection of Industry 4.0*; River Publishers: Gistrup, Denmark, 2020.
25. Rajawat, A.S.; Bedi, P.; Goyal, S.B.; Alharbi, A.R.; Aljaedi, A.; Jamal, S.S.; Shukla, P.K. Fog Big Data Analysis for IoT Sensor Application Using Fusion Deep Learning. *Math. Probl. Eng.* **2021**, *2021*, 6876688. [\[CrossRef\]](#)
26. Rajawat, A.S.; Barhanpurkar, K.; Goyal, S.B.; Bedi, P.; Shaw, R.N.; Ghosh, A. Efficient Deep Learning for Reforming Authentic Content Searching on Big Data. In *Advanced Computing and Intelligent Technologies*; Springer: Singapore, 2022; Volume 218. [\[CrossRef\]](#)
27. Goyal, S.B.; Bedi, P.; Kumar, J.; Varadarajan, V. Deep learning application for sensing available spectrum for cognitive radio: An ECRNN approach. *Peer-to-Peer Netw. Appl.* **2021**, *14*, 3235–3249. [\[CrossRef\]](#)
28. Shilpa, B.; Budati, A.K.; Rao, L.K.; Goyal, S.B. Deep learning based optimised data transmission over 5G networks with Lagrangian encoder. *Comput. Electr. Eng.* **2022**, *102*, 108164. [\[CrossRef\]](#)
29. Petrenko, S. *4 From the Detection of Cyber-Attacks to Self-Healing Industry 4.0*; River Publishers: Gistrup, Denmark, 2020.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.