Artificial Intelligence for Web-based Educational Systems

Wang Dong
School of Computing, University of Washington, Seattle, WA.
wang89778@hotmail.com

ArticleInfo
R. Arulmurugan et al. (eds.), *First International Conference on Machines, Computing and Management Technologies*, Advances in Intelligent Systems and Technologies
Doi: https://doi.org/10.53759/aist/978-9914-9946-0-5_7

Abstract – Due to the global COVID-19 epidemic in the preceding two years, there has been a significant debate among different academics about how learners may be lectured through the web while maintaining a higher degree of cognitive efficiency. Students may have problems concentrating on their work because of the absence of teacher-student connection, but there are benefits to online learning that are not offered in conventional classrooms. The Adaptive and Intelligent Web-based Educational Systems (AIWES) is a platform that incorporates the design of students' online courses. RLATES is an AIWES that uses reinforcement learning to build instructional tactics. This research intends the aggregation and evaluation of the present research, model classification, and design techniques for integrated functional academic frameworks as a precondition to undertaking research in this subject, with the purpose of acting as an academic standard in the related fields to aid them obtain accessibility to fundamental materials conveniently and quickly.

Keywords – Adaptive and Intelligent Web-based Educational Systems (AIWES), Machine learning (ML), Reinforcement Learning.

I. INTRODUCTION

Machine Learning (ML) is the study of how computers learn, specifically how to use information gathered from past experiences to do better on future iterations of the same tasks. It is considered Artificial Intelligence (AI) because it mimics human thoughts. In order to make proper decisions and or predictions without being trained, ML algorithms structure a framework using a sample of data known as training data. ML algorithms are employed in different fields, including where it is impossible or difficult to design analogous algorithms to accomplish different required tasks, e.g., pattern recognition, agribusiness, voice recognition, email filtering, and medical analytics. However, not all ML is quantitative training, but a segment of it is connected to computer science, which is concerned with generating predictions using computers. Mathematical optimizations provide a platform for ML with novel tools, theoretical models, and prospective application areas. The exploratory data evaluation using unsupervised learning is the major emphasis and foundation of data mining, an associated domain of research. Some types of ML make use of neural networks and big data in a way that is reminiscent to how the brain of humans operates. ML is sometimes denoted as predictive analytics when used to commercial concerns.

Algorithms that learn from experience assume that successful prior approaches, algorithms, and conclusions will continue to provide positive results going forward. For example, "X% of families contain geographically different species with color variations, hence there is a Y% probability that unknown black swans exist." Machine learning software may carry out operations that were not included in their original code. Automated learning is the process of teaching a computer to do a job by seeing human examples of success. Programmable algorithms may instruct a computer on how to carry out a series of operations that will ultimately lead to a solution, eliminating the necessity for any form of training on the side of machines when dealing with straightforward problems. A human being may have trouble coming up with the required algorithms for rapidly increasing complex tasks. In addition, it may be more effective to aid computers in constructing its own algorithm compared to having human developer categorically define every step.

The objective of ML is to use a variety of techniques to train computers to complete tasks for which there is presently no adequate solutions [1]. Whenever there are increasingly multiple answers to select from, the best approach to follow is to designate the best one as "valid." Computers may use this data as training data to effectively define its approach of arriving at the best decisions. The MNISST database that integrates handwritten figures has been increasingly employed in training systems for different tasks for digital feature recognition. In the early 1960s, the Raytheon Company created Cybertron, an exploratory "learning machine" with perforated tape memory that could analyze sonar data, electrocardiogram (ECG), and voice modulation via the use of primitive reinforcement learning. It was fitted with a "goof" key to force it to reevaluate bad judgments and was "taught" by a human controller to spot patterns via repetition. Nilsson's Learning Machines, which focuses on using computers to classify data, is illustrative of the field of machine learning as it developed in the 1960s. In 1973, Zhao, Zhao, Xue, Yang, and Liao [2] highlighted how interest in pattern recognition had persisted into the next decade. It was reported in 1981 that neural networks might be trained to effectively identify 40 different symbols (4 specialized symbols, 10 numbers, and 26 letters) from a centralized computer system via the use of several training methodologies.

An even for formalized definition of the algorithm explored in ML was provided by Bulstra and Machine Learning Consortium [3], and is often cited: " When the program's performance on tasks in the class *T*, as assessed by the performance measure *P*, improves as a function of experience E, we say that the program has learned from experience E with respect to the class T and the performance measure *P*." As opposed to describing machine learning in cognitive terms, this characterization of different tasks with which it is concerned provides a comprehensibly operational definition. In Hossain and Miah [4]' work, the cognitive capability was discussed by asking: "Can machines think?" as "Can machines accomplish what humans (as thinking creatures) can achieve?" This discussion is in line with Crosby [5] recommendations about the level of cognition in machines. The first objective of modernized machine learning is to classify data based on pre-existing frameworks; second, is to forecast futuristic occurrences using these frameworks. Hypothetical data-categorization framework may be training to effectively differentiate between benign and malignant agents through computer vision of these agents and approaches to supervised learning.

Machine learning technology has numerous potential applications in many different areas, including business, biology, medicine and education. For many years now, big data has been utilized in the evaluation of data on different scholars at distinct stages in their learning process to develop learning policies, and machine learning technology has been extensively utilized in education, for example, to evaluate the accomplishments and performance of students, and to execute the appropriate steps to increase student engagement and assist them graduate. Researchers have begun to employ machine learning to create teaching techniques as the domains of ML technology continues to advance. This has led to the creation of a number of educational systems, including the ASES (Adaptive Smart Educational Systems [6]), AIES (Adaptive and Intelligent Educational Systems [7]), and Adaptive Learning System [8].

Adaptive and Intelligent Web-based Educational Systems (AIWES) frequently employ a reinforcement training technique called Reinforcement Learning (RL) in Adaptive and Intelligent Educational System (RLATES) to develop pedagogical strategies. A customized optimum learning approach may be developed via the interplay between the learner and the system, which is made possible by using reinforcement learning to the construction of teaching techniques. According to the available literature, the following issues often hinder RLATES' performance: (i) The conventional Q-learning method is used exclusively in the existing literature to train a network and plan suitable pedagogical approaches, however this algorithm has a flaw in that it exaggerates the value of an action in specific circumstances. For scenarios that have not been frequently taught, the model in an adaptive instructional system may not be enough. Because of the over-practice issue, however, more study time is not commensurate with increased practice. (iv) The next step is to provide further detail on these three restrictions.

First, a conventional Q-learning algorithm cannot escape the overestimation issue indicated, which is discussed by Qu, Yu, Houston, Conte, Nandi, and Bowman [9]. However, many research still favor using a conventional Q-learning algorithm whenever employing reinforcement learning to adaptive education system, and in various researches, authors in [10] fail to discuss the overestimation issue. More advanced than the predictive Q-learning approach, double DQN and the double Q-learning algorithms have been created for the area of reinforcement learning, and by using these algorithms to the adaptive learning framework, possible exaggeration concerns may be avoided to some degree. Second, Q-learning is utilized in an adaptive instructional program by Blomeyer Jr [11], however there is an implementation issue with the simulation model in case users are not trained effectively, i.e. if much iteration is not seen for the approach to train, then the systems might not work optimally. Their work must be considered despite the systemic flaws. According to their findings, the system for additional steps or states to be proactively incorporated instantaneously, which means that educators and learners alike may contribute material to systems, which they deem significant or needed, and the framework can be instantaneously upgraded to reflect these additions. This implies the system is very flexible and simple to use. In regards to the final point, there are writers that implement novel algorithms in courseware design, such as Partially Observable Markov Decision Process (POMDP) [12] and Proximal Policy Optimization (PPO) [13].

Over-practice was a major issue in Untila's research [14] since he found that having students complete more tasks did not necessarily lead to increased time spent on task. However, the neural networks employed in this article diminished the level of complexity of the state space and actions, which in turn required fewer samples for the method to converge. Although Zhang's study did not suffer from the over-practice issue, the paper's reinforcement learning algorithm's reward levels might be tweaked to improve its performance. The most important takeaway from this research is that the best answer may still be delivered by the POMDP (Partially Observable Markov Decision Process) even if the student only provides partial knowledge. In addition, Apriyanto et al.'s [15] article proposes a system that may collect the information locally while responding accurately to questions from student users.

The objective of this article is to help researchers in related disciplines improve their work by providing a high-level summary regarding how reinforcement learning might be employed in Adaptive and Intelligent Web-based Educational Systems (AIWES) and by contrasting and summarizing relevant efforts thus far. The rest of the article is organized as follows: Section II focuses on an overview of RL reviewing aspects of Markov Decision Process, Q-learning, Double Deep Q-Network and Deep Q-Network, and Comparisons with Bayesian Network. Section III focuses on RLATES with discussion of the current research, applied RL in RLATES and Model-free and Model-oriented RL. Section IV summarizes the paper drawing concluding remarks.

II. REINFORCEMENT LEARNING

According to Xu, Han, Jiao, and Gao [16], Reinforcement Learning (RL) focuses on how organisms might acquire the knowledge to understand the relationships between inputs, behaviors, and the occurrence of either rewards (positive outcomes) or punishments (negative outcomes). Both positive (rewards) and negative (punishments) are examples of reinforcers, and reinforcement refers to the process through which the reinforcer forms and strengthens these connections (negative reinforcers). The learner's behaviors are influenced by these connections in a number of ways, including the formation of automatic and vegetative reactions based on the expectation of rewards and punishments. Because of its obvious adaptive importance, RL has been found in diverse taxa as far removed from chordates as nematodes, arthropods, mollusks, and, of course, chordates. Contemporary neuro-computational theories of RL may be located at the crossroads where animal learning and AI research converged in the 20th century. Behavioral frameworks and psychological notions are part of the legacy of the first thread, whereas their formalization in mathematics is the legacy of the second.

Cardon, Ma, and Fleischmann [17] argue that both the algorithmic and the psychological perspectives hold that the learner (whether it be an animal or an artificial) is motivated by some kind of reward (goal-directness). Compared to other forms of learning, including procedural or observational learning, RL is distinguished by this aspect. Two characteristics become clear from this perspective: RL is selectional (the agent must attempt to pick among multiple other possibilities) and associative (Each option selected must correspond to a specific circumstance). Early studies of animal learning referred to RL as conditioning. Classical conditioning and instrumental conditioning are the two primary types of conditioning experimental paradigms.

Curiel and Poling [18] posit that associating a reinforcer with a stimulus or behavior is a key component of the minimum conditioning processes. When using classical conditioning, the reinforcer is given to the learner regardless of their behavior, and the observed response is modeled after automatic, pre-programmed reactions. When Pavlov's dog heard the bell that signaled the arrival of food, he instinctively started salivating. In instrumental conditioning, the recipient's behavior determines whether or not they get the reinforcer. De Aguilar-Nascimento's [19] early experimental studies of this process revealed this trait: a caged animal would learn to undertake certain behaviors (string pulling, lever pushing) in order to free itself from confinement or get food. Several requirements have been demonstrated to be essential when examining the causative factors of conditioning, including temporal contiguity (an action or a stimulus must be temporally near to the result for an association to be made), contingency (the likelihood of an outcome should be greater after the action or the stimulus, i.e. the action or the stimulation ought to be forecasters of the occurrence), and prediction error (In cases when the learner was unable to foresee a certain consequence from a given action or stimulus, an association is made between the two).

This latter concept was initially presented by Young [20]. They were particularly curious in a phenomenon in conditioning known as the "blocking effect." An animal is initially presented with a first conditioned stimulus (here, a bell ring) that anticipates the delivery of a reinforcer (food pellets) (i.e., food). Once the animal has learned to associate the bell with the food, they will be exposed to a different stimulus (in this case, a light) alongside the meal. Therefore, the bell and the light are both indicators of forthcoming nourishment. As though "blocked" by the initial connection, the animal does not learn to associate the light with the food when tested. It was hypothesized in the Rescorla-Wagner theory by Kimmel and Lachnit [21] that conditioning takes place not only when two events happen at the same time, but even when the coincidence between them is not obvious from prior experience. Because the bell perfectly predicts the arrival of food in the previous case, no fresh link is formed between the light and food. Specifically, they use a "prediction error," which they describe as the discrepancy between the expected and actual reinforcer, as their primary learning signal. The accuracy of a forecast of a reinforcer (reward or punishment) is measured by its error, and the Rescorla-Wagner theory is an error reduction strategy.

The case of reinforcement theory provides a useful psychological paradigm for understanding RL from an AI perspective, which is a branch of machine learning seeking computational alternatives to a range of problems. The agent is imagined to move from one state of the environment to another, making decisions about what to do and what not to do in order to maximize a quantitative reward1. Taking this into account, an RL agent needs to be able to do two primary things: (i) predict what reward will be received from a given state (reward prediction); and (ii) choose the best possible action to take in order to maximize that reward (choice). Temporal difference (TD) learning is a key component of many prominent current RL models. Learning in this model is based on a reward forecast error term, similar to the learning rule employed by Rescorla-Wagner theory. The TD learning algorithm uses this information to construct correct reward forecasts from delayed rewards. Q-learning is an extended version of TD learning that focuses on learning the expected reward for each action individually. In this case, the best option is the one that has the highest expected reward. Q-learning relies on a TD error as well.

In this way, RL algorithms allow the experimenter to extrapolate important computational parameters of these simulations and generate quantitative predictions on the expected evolution of neurological and behavioral data given the model's assumptions. These cognitive constructs are called "hidden variables" to distinguish them from the experimental observables (decisions, response times) from which they are produced. We'll examine the monkey brain's mapping of these computationally hidden variables in the following part, with a special emphasis on prediction mistakes. Agent, surroundings, action, incentive, and state are the five main components of reinforcement learning. There are two key components in a reinforcement learning algorithm: the agent and the surroundings. The agent can observe the state of the

surroundings and respond accordingly depending on the reward it has got from the environment after interacting with it to create an action. A clearer depiction of this procedure is shown in **Fig 1**.

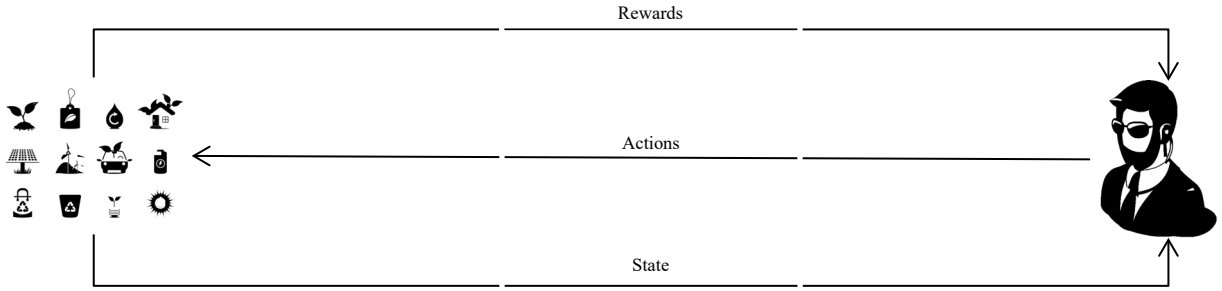


Fig 1. A representation framework of RL case

Model-Free and Model-oriented Reinforcement Learning

There are two major forms of Reinforcemnt Learning (RL) algorithms: Model-free and Model-oriented. Model-oriented RL algorithms works where the agent must interact with the virtual world, learn from it, and then use that model to inform its future actions. In model-free RL algorithms, agents do not construct a model of its setting but instead formulates and learns different actions directly by interacting with the environment. The statistics in **Fig. 2** were collected by scouring publications available in many popular databases published in the previous five years and indicate the model-oriented RL algorithms are not as popular in the area of education as model-free reinforcement learning algorithms (2018–2022). Some examples of modelbased reinforcement learning algorithms include the Imagination-Augmented Agents (I2As), the World Models and the Model-oriented Value Expansion (MBEV). Model-free RL algorithms, such as Soft Actor-Critic (SAC) [22], Proximal Policy Optimization (PPO) [23], Categorical Distributional RL (CDRL) [24], Deep Deterministic Policy Gradient (DDPG) [25], Deep Q-Network (DQN) [26], and Q-learning, are used in a wider variety of applications than model-oriented RL methods.

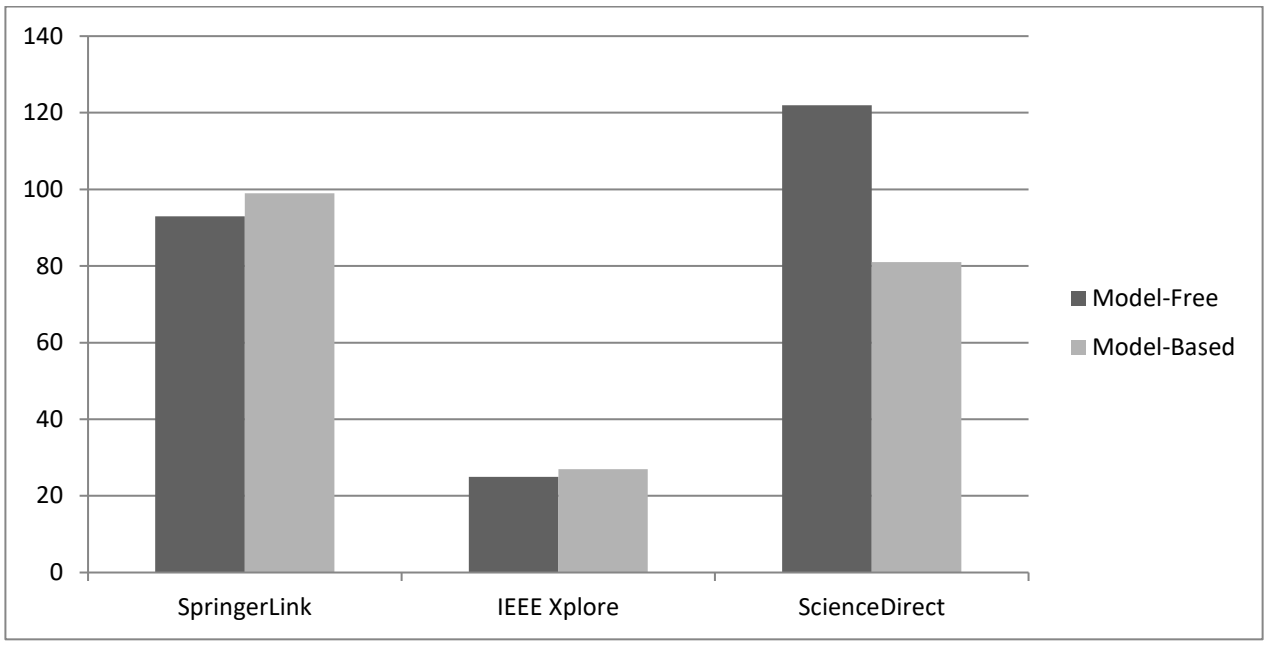


Fig 2. Publications comparing model-oriented and model-free RL algorithms

Markov Decision Process

A Markov Decision Process (MDP) [27] is the foundation of reinforcement learning; each MDP tuple includes a limited number of actions and variable transfer possibilities. Below are examples of the reward and transitional functions that make up the MDP paradigm.

$$S \times A \times T: S \to (0, \ldots 1) \cdots S: R \times S \times A \to R \tag{1}$$

Over time, several variations of the Markov decision process have emerged; three of these variants are summarized in **Table 1**.

Table 1. Distinct MDP versions

MDP versions	Type of system	Features
Fully Observabe MDPs (FMDPs)	Discrete	Fully observable
Semi MDPs (SMDPs)	Continuos	Generally observable
Partially Observable MDPs (POMDPs)	Discrete	Incompletely observable

Q-Learning

As a model-free grounds approach that has been extensively used in various reinforcement learning studies, Q-learning serves as the gold standard for Optimization algorithms [28]. The expression below is the principle of Q-learning, which is generated from the Bellman framework.

$$\sum_a \pi\,(a,s) \sum_s\, p\,(s,a) \neq\, s\,[\gamma\, V_\pi\,(s') + W_{s\to s' \neq a}] = V_\pi \tag{2}$$

$$\sum_{s'}\, p\,(a,\ s \neq s') \left[\gamma\, \sum_{s'}\,(a',s')\, Q_{\pi_{(s',a')}} + \&\, W_{s\to s' \neq a}\right] = Q_\pi\,(s,\ a) \tag{3}$$

V_π alludes to the SVF (State Value Fuction) while Q (s, a) alludes to the AVF (Action Value Function) within the Bellman equation. Since going from state s to state $s\ +\ 1$ is unpredictable, we need to add the expectation E to the equation above, where r stands for the reward. Using a Q-table organized as s, a, whereby s signifies different sates, whereas a signifies different actions/activities, Q-learning algorithm chooses the appropriate policy. The preceding phase may be calculated from the existing state of affairs using the provided Q-table. After making a decision on what to do next, the agent carries out that decision and, upon successfully completing the action, agents obtain their rewards from the environment. Q-table in its setting is typically upgraded after each action, and it is modified using the given equations.

$$Q\,(a_t, s_t) \leftarrow Q\,(\,a_t, s_t) + \alpha\, \left[\gamma \max_{at} \ldots . Q\,(a_t, s_{t+1}) + r - Q\,(\,a_t, s_t)\right] \tag{4}$$

The variables in this equation are the reward (r), the action (a), the state (s), the rate of learning (α), and the discounting factor (α and γ) both and operate on the same 0–1 scale. In 1993, however, Bradley, Pooley, and Kockelmann [29] proposed that, due to the possibility of random errors occurring uniformly within the action/activity values, bias could be evident in the pursuit for ideals amounting to sub-optimum solutions instead of the optimum one. In a research published in 2014, Tiwari, Kumaraswamidhas, and Garg [30] demonstrated that ambient noise might contribute to overestimation issues in traditional Q-learning algorithms, a topic that hadn't been addressed until then. It was claimed in Li, Meng, Gao, Zhang, and Chen's [31] research that Double Q-learning algorithm could be employed in addressing over-estimation issue; a similar technique, dubbed Double DQN, was subsequently published by Lee, Jung, and Chung [32]; the latter's approach is described in more depth below.

Deep Q-Network and Double Deep Q-Network

Since Deep Q-Networks (DQN) employ "Deep" "Convolutional Neural Networks" (CNN), the "Deep" in "Deep Q Networks" (DQNs) alludes to the employment of such networks. CNNs represent to a form of deep learning that takes cues from the human visual cortex in its attempt to interpret the visual data being collected by external sensors (the eyes). A human-defined algorithm is taught separately to comprehend the items within an image, the specific cases, and locations of every case, and this simplified dataset is therefore fed into agents as inputs to structure streamlined states for agents to effectively operate on. In the latter case, we also addressed the manner in which we provide RL agents the capability to autonomously normalize the states of raw image pixel so that is could derive insights from them. There, we also had a short discussion of CNN's (Convolutional Neural Networks) significance.

CNNs are multi-layered networks of Convolutional Neurons, with each layer's Convolutional Neurons using a unique kernel (function) to incrementally cover the image. The convolutional layer could produce different convolution lesser maps as opposed to input pixel sizes of about N x N for every channel, but all of the resultant maps utilize the same weighting for the kernels if the input images incorporate three distinct channels of colors, each of which represent the N x N pixels. Convolutional Neural Network (CNN) is higher compared to Deep Neural Network (DNN) that is based on multi-layer perception for handling images since weights for kernels in the layers remain the same, requiring optimization of just a single vector to bring out the salient characteristics in an image. However, a CNN's output is a multi-dimensional tensor that is useless when used as input for a regression or classification (value estimation) system. As a result, a CNN's final convolutional layer is often linked to one or more flat layers (that are analogous to encrypted elements with DNN) before being input in a Soft-Max" activation map for categorization as Linear" activation system for regression. Class-probabilities for all classes requiring classification are generated by the 'SoftMax' activation layer, and the optimal course of action is calculated by selecting the output class with a high category-probability (arg-max).

In case you missed it, the CNN networks are really included inside the DQN one. In the previous section, we focused on a specific DQN that was able to perform well across 49 Atari games simultaneously by employing an architectural feature that started with a CNN with multiple convolutional layers, continued with two whole layers, and concluded with an 18-class 'SoftMax' classification scheme. This list of 18 groups represents the many ways in which the game may react to player actions (One 8-way gamepad and single buttons for various games house Atari Controllers). Do-nothing, eight-category signifying eight joystick directions (i.e. Move Diagonal Left Up, Move Straight Left, Move Diagonal Right Down, Move Straight Right, Move Diagonal Right Up, and Move Straight Up), Press-Button (without moving), and eight actions that correspond to pressing buttons create the 18 categories utilized with DeepMin. At each juncture when the agent must take some kind of action (which, as we will see, need not necessarily correspond to each and every one of the steps), the agent makes a selection among the available actions (it should be noted that one of these actions is Do-Nothing).

Varghese and Mahmoud [33] purposed to provide readers the tools they needed to build their own practical RL agent. We may need to tailor the CNN ecosystem and the design of the output units for a specific sector and use case, such as the ones we may implement for Atari, to keep the 60 FPS visual output rate stable. This implies that in (i) a single second, (ii) the game will produce and (iii) deliver 60 images. This signal might be used to inform the condition of our agent. One potential problem of training a Q-Learning-Network with raw image pixels and directly handling subsequent frameworks at a prompt frame rate is that training of Q-Learning-Network might potentially diverge or become caught in a hunting loop instead of converging. The DQN needed these three improvements to achieve downward convergence and practical application while dealing with high frame rate, correlated images, and high dimension data.

Traditional Q-learning techniques have trouble handling increasingly complicated tasks because it is difficult for computer systems to maintain all the data needed to do the calculation, and because an excessively big Q-table renders Q-values recovery substantially passive. By using states and actions as inputs to effectively construct essential Q-values, convolutional neural networks reduce the need for both storage space and processing time for the Q-table. In order to project value functions, DQN uses CNN. In DQN, the -greedy approach is utilized to decide what to do next. Each time a choice has to be made, a greedy approach like -greedy the one that would bring in the most money. The DQN's methodology is then outlined. To begin, the agent receives a state value from the surrounding environment and uses this value to determine all the states from which it may choose and the actions it can do. The environment delivers the selected action's reinforcement and the updated state value after a -greedy policy has been applied to choose the next action. DQN develops larger loops of these to efficiently attain and design an optimum policy, building on the selected loop that has been used so far.

Consequently, DQN shares Q-inability learning's to avoid the overestimation issue; nevertheless, a novel method was presented in a 2016 publication that offers an early solution to the exaggeration problem. The fundamental cause of the overestimation issue is addressed in this article by the definition of a function originally named Double DQN, which prevents the overestimation from continually spreading from one condition to the next. After the maximal Q-value has been chosen, the same value is chosen in a second communication topology on the corresponding action values, and the chance of overstatement is much reduced when the Q-value is the largest value including both subsystems, but it could still appear on occasions. When compared to traditional Q-learning and traditional DQN, Double DQN significantly improves upon the overestimation issue.

Using Double Q-learning, a Double DQN breaks out the maximum function in the target into adaptive control and action assessment to reduce overestimation. Although the greedy strategy is assessed in terms of the online network, its value is calculated in terms of the target network. The upgrade is similar for DQN, but it replaces the target $Y_t^{double\ DQN}$ with:

$$Y_t^{double\ DQN} = R_{t+1} + \gamma Q\left[S_{t+1}, \underset{a}{\operatorname{argmax}}\, Q(s_{t+1}, a; \theta_t); \theta -_t\right] \tag{5}$$

In Double DQN, the weights of the network communication t are used instead of the values of the second network θt− to assess how well the current greedy strategy is working. This is an improvement over the initial conception of Double Q-Learning.

Comparisons with Bayesian Network

To express a collection of variables and their conditional dependencies, a Bayesian network (Bayes network, belief network, Bayes net, or decision network) uses a Directed Acyclic Graph (DAG) [34]. When there are several plausible explanations for a given occurrence, Bayesian networks may accurately forecast the chance that any given explanation was responsible. The probabilistic associations between illnesses and symptoms, for instance, may be represented by a Bayesian network. The network may be used to estimate the likelihood of different illnesses given a list of symptoms. In Bayesian networks, inference and learning may be performed via efficient algorithms. Dynamic Bayesian networks allude to the Bayesian network, which model variable patterns (such as protein patterns or voice signals). Influence images are Bayesian network generalizations, which could address and model decision challenges involving uncertainty.

Research has also looked at using other AI algorithms, such as the Bayesian Network, in intelligent educational systems, so it's not only reinforcement learning that's being used. Bayesian Network is a technique for finding the best answers, much as reinforcement learning is. This system design, if employed to smart training systems, is congruent with the ideology of learning from learners' student and knowledge features and, on the basis of this, suggesting training approaches for the essential groups. However, Bayesian networks are an iterative procedure and stateless. Despite being iterative in nature, the results of one function call have no bearing on those of subsequent calls. In contrast, the optimization process for reinforcement learning is a stateful overall process, and every transition between states has an impact on the next one. This implies that the decision made in the previous phase has an impact on the transition towards the next state. In contrast to Bayesian Network, reinforcement learning requires a cumulative total of all rewards in order to find the optimal solution.

III. ADAPTABILITY AND INTELLIGENCE OF EDUCATIONAL SYSTEMS

Reinforcement Learning (RL) is studied by many different disciplines because to its adaptability. Some of these disciplines include: statistics, multi-agent networks, simulation-based algorithms, pattern recognition, systems engineering, operations research, and cognitive science. RL alludes to neuro-dynamic programming or approximation dynamic programming in the domain of operational research and operational control. Although the theory of optimal control has investigated many of the same issues that interest reinforcement learning, its focus has been on the existence and characterisation of optimum solutions and methods for their accurate calculation rather than on approximation or learning, in particular in the aspect of mathematical representation of the surroundings. Reinforcement learning could be used to provide light on the emergence of equilibrium in the presence of restricted rationality in economics and game theory.

An MDP, or Markov decision process, is used to represent elementary RL: (i) a collection of states (S) describing the status of the environment and the agent; (ii) an inventory (A) of the agent's activities; The objective of RL is to teach the agent to adopt a strategy that maximizes the "reward function" or some other reinforcement signal supplied by the user, which is accumulated from the immediate rewards. This seems to be a process that also occurs in the mind of animals. To provide just one example, genetic brains are designed to perceive signals such as hunger or pain as negative reinforcements, and signals such as food intake and pleasure as positive enforcements. Animals tend to learn how to maximize their chances of receiving these rewards under certain conditions. It is possible that animals may learn with the help of reinforcement after all. Problems are said to be fully observable if and therefore only if they can be expressed as a Markov Decision Process (MDP) on the premise that the agent has perfect knowledge of the current configuration of the surroundings. An agent is said to have partial observability if it can only see a subset of different states, or in case the visualized states are full of noise; in this case, the issues have to be defined formally as POMDP. Constraining the options of agents is considered an option in both cases. For instance, the condition of account balances could be limited to be positive; in case the present value of the states is 3 and the state transitions try to decrease the overall value by 4, the transitions will not be allowed.

The ideology of "regret" arises whenever the performance of an agent is compared to that of optimally-acting agents. The agents should be consider the long-term impacts of actions (i.e. maximizing future revenues) even if doing so will result in a negative immediate reward in order to operate near optimally. In this way, challenges involving a short-term vs. long-term reward trade-offs are especially well-suited to reinforcement learning. In addition to its use in robot control, elevator scheduling, telephony, checkers, backgammon, and Go, it has been used effectively to a wide variety of issues (AlphaGo). The application of sample to boost performance and the application of function estimation to deal with massive environments are two major elements, which render RL effective. Because of these two factors, reinforcement learning may be used in the following settings with significant resources: When (i) an environmental model is available but no analytical solution exists, or (ii) a simulation model of the ecosystem is given (the aspect of simulation-oriented optimization), simulation-based optimization is the method of choice. (iii) Interacting with the world around you is the only way to learn more about it. We may classify the first two as planning issues (given the existence of a model) and the third as a true learning challenge. Reinforcement learning, however, transforms both planning issues into machine learning challenges.

Recent years have seen a rise in interest in distance education. Distance learning is crucial when both students and instructors are unable to be physically present in the same classroom. Basic needs for distant learning may be addressed by learning using web-based materials (text, video, photos, audio, etc.) or online lessons supplied by instructors. One-to-many teacher-student interactions eventually fail to fulfill one role, adaptive instructions, since students cannot correctly discover the solutions to issues they confront via online resources. Due to high expenses on one-on-one educational programs, it is not effective to employ this approach to all students' groups despite the fact that these classes are more productive and more fulfilling than small group courses. In light of this reality, the Adaptive Intelligence Educational System was created to provide each student access to their virtual instructor and enable them to benefit from one-on-one training methodology at a minimum cost, provided they have access to a computer.

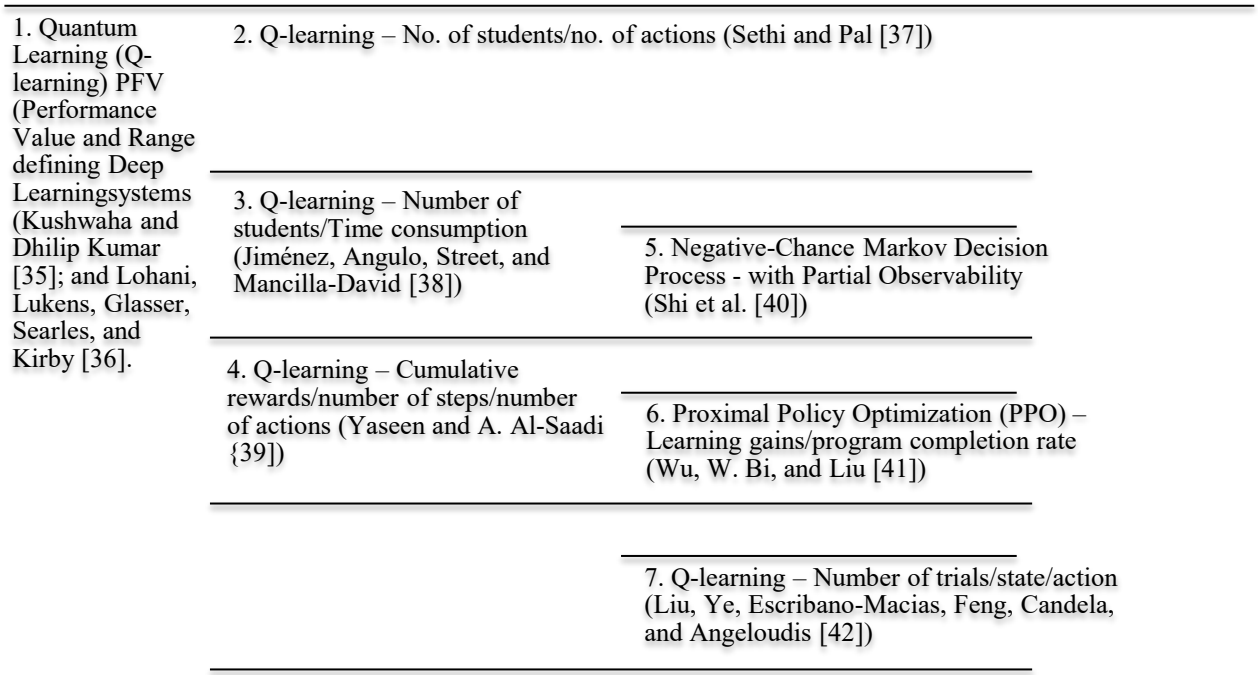
AIWESs use a number of machine learning algorithms to understand student characteristics in order to re-sequence all course content modules depending on those individuals' unique profiles. Incorporating learning algorithm, which allows for more natural student engagement and a more robust learning paradigm, is an approach advocated by AIWESs to enhance an overall system performance. Reinforcement Learning Automated Tutoring System (RLATES) is a system that

incorporates RL into AIWESs. The knowledge framework and the educational approach framework are also parts of RLATES. The knowledge model is where choices like which chapters of the textbook to cover and how those lessons will be presented (through video, audio, text, or images) are made. In the pedagogical strategy framework, the approach for delivering the content is designed. However, RLATES cannot be taught in its entirety from the ground up at this time. In the early stages, training data is used to educate the model how to best approach each student's unique traits. When constructing the system, it is fundamental to subdivide the whole experimental procedure into two phases: training and teaching. Real-world pedagogical use is contingent upon the model's having been trained effectively.

Current Research

An overview of where things are in terms of study into intelligent pedagogical systems is provided here. The retrieval also indicates that only a small percentage of the research focused AIWESs really employed reinforcement learning methods. See **Fig. 3** for specifics.

Fig 3. The current publications for AIWES



It is seen from **Fig. 3** that the classic Q-learning method is still frequently employed in the area of AIWES. Given that the Q-learning technique is a model- and policy-free RL technique, it makes perfect sense to apply it to these kinds of setups. However, Q-learning's flaws slow down processing and lengthen the time it takes for the system to respond when the Q-table is too big. On the other hand, many authors have settled on the Q-learning technique because it belongs to the classic RL algorithms and is easier to implement in practice than model-free alternatives. All five of the aforementioned articles use Q-learning methods, but each has its own set of criteria for success.

Most research utilizes the amount of time spent, the number of scholars, and the number of steps to assess the effectiveness of their models. One of the most interesting pieces is Publication 1, in which the writers create their own assessment measure they call PFM. If the PFM is less than 60, the article's writers consider the model's performance to be excellent, and if it is more than 60, the performance is considered to be bad. Additionally, the results of a PFM evaluation might provide some insight into the challenge level of the course material: low scores suggest the material is more difficult than expected. Even though it precludes a side-by-side comparison of the model's efficiency and productivity across different publications, this assessment measure is used by the authors to evaluate the three tactics inside the article, and it makes the assessment findings more intuitive and clear to read and comprehend.

The process of learning educational policies according to the requirements of learners in an AIWES is the best match for RL. Employing RL in AIWES from the start is impractical since previous efforts have shown that a considerable deal of experience is required for the mode to learn to train appropriately. Theoretical researches have shown that the amount of experience needed to acquire an appropriate educational strategy may be reduced by seeding AIWES with an earlier value function learnt with simulated learners. We show empirically that the AIWES can get extremely accurate starting educational policy from a value function trained with simulated learners. More than seventy first-year computer science students participated in the assessment, proving that they were able to acquire a helpful and efficient overview of the course material.

There has been a recent uptick in investment towards the study of distance learning. Programs in pedagogical models have often consisted of unmodifiable, static websites. But beginning in the 1990s, scientists have been introducing

flexibility into their designs. Utilizing educational knowledge representation oriented to RL allows a pedagogical system to effectively customize training for each individual learner. The system is designed to sequence its material optimally since no fixed and preset instructional rules need to be created for individual students. However, a lot of training data is required for RL algorithms to converge on a solid strategy for action. Furthermore, RL systems respond nearly arbitrarily according to a value function started randomly in the first trials of the learning if the system has not been preinitialized to a pedagogical method. Since disinterested or bored students may be detrimental to an educational system, it is crucial that lessons be presented in a rational manner at all times.

Initiating the value function with one that was trained for performing a comparable problem using a similar model has been shown to speed up learning in a research. Learning the action policy may be sped up by using previously recorded experience tuples to initialize the value function. By instantiating the system with an educational approach, even if it does not perfectly correspond with the present students' demands, we have shown experimentally with simulated students in prior work that the complexity of the learning stage may be decreased.

Applied RL in RLATES

It is clear from Section II's overview that reinforcement learning consists of five distinct parts. To successfully implement RL in RLATES, it is crucial to verify that the system's parts match up with the five main parts of RL algorithms. In this article, we will discuss how reinforcement learning may be used with RLATES to improve the quality of the results. RLATES' components are first described in terms of their counterparts in the algorithm for reinforcement learning in **Table 2** below:

Table 2. Overview of RLATES Components

RLATES' components	
Agent	It is the learner, or "agent," who acts as the focus of RLATES. This, the learner is analogous to an agent in the RL algorithm since the training model is utilize by the learners through interactions with the model for following operations.
Environment	The environment, which may be thought of as the sum total of the system's accumulated knowledge, is responsible for assessing students' mastery of course material and gathering demographic data.
Action	Each knowledge module in RLATES corresponds to an action, since actions are the decisions an agent must make at each stage.
State	The state in reinforcement learning algorithms is the condition that the surroundings returns to after an action is considered by the agent. As a result, in RLATES, the state represents the student's level of mastery of the material. In this case, a vector is utilized to keep track of information, and each state's value falls between zero and one. If the pupil has achieved mastery, the state value is 1. The state value is 0 in case the student has not understood the materials.
Reward	Each choice in a reinforcement learning algorithm results in a distinct reward, and in RLATES, every knowledge module has its own reward that varies with its importance. And in RLATES, the goal is to maximize this payoff over time.

System 1 then details how the RLATES dataset was fed into the RL algorithm. The procedure below results from combining RLATES's parts with those of a reinforcement learning algorithm:

Algorithm 1 Apply a recurrent neural network for RLATES
Set $Q(s,a)$ for $a \in A \; and \; s \in S$
Inquire into the state of the students' understanding
Repeat after each episode,
Select a learning module a and provide it to the learner via $\varepsilon -$greedy policy
Collect the incentive r; if the RLATES objective is met, r would be non-negative, and otherwise, r will be zero.
Inquire into the state of the students' understanding
Update $Q\ (s,a)$:

$$Q\ (\ a_t, s_t) \leftarrow Q\ (\ a_t, s_t) + \alpha \left[\gamma \max_{a\ t} Q(a_t + s_{t+1}) + \ r - Q\ (a_t + s_t)\right]$$

Until s exceeds the set condition

IV. CONCLUSIONS

This article provides a short overview of the ideas and algorithms behind intelligent pedagogical systems, as well as a literature review on the creation of such systems that emphasizes adaptability. This article provides a concise overview of current studies that may be used as a resource for scholars in related fields. The findings of this literature review are as follows. (i) Due to the qualities of adaptive education system, Reinforcement Learning (RL) is suited for application within the development of systems, and could be significant in providing enough training techniques for individuals with the same qualities. The more advanced reinforcement learning algorithms have seldom been applied to the subject of smart

educational systems (ii) despite the fact that many researchers have pondered how to implement RL into Adaptive and Intelligent Web-based Educational Systems (AIWES). Most studies use the same set of assessment measures for gauging the effectiveness of the system under study, allowing researchers to easily draw parallels across the various analyses.

There are limits to being able to compare experimental outcomes between researches; however some studies have established their own assessment criteria to better analyze the experimental data for future improvement. Although online education is becoming more and more of a need for today's students, little study has been done on the topic of applying RL to AIWES. As with most trends in history, online learning would not ever be able to totally replace traditional classroom instruction. However, advances in both computing power and educational theory mean that the latter will become more obsolete as the former becomes the norm. This paper's limitations include the fact that the given and analyzed reinforcement learning and AIWES are based only on literature elements and have not been confirmed and assessed in real tests. Future study will include integrating the combining a Bayesian Network with RL techniques for improved system's operational effectiveness and the algorithm's computational complexity.

References

[1]. M. T. Barros, H. Siljak, P. Mullen, C. Papadias, J. Hyttinen, and N. Marchetti, “Objective supervised machine learning-based classification and inference of biological neuronal networks,” Molecules, vol. 27, no. 19, 2022.

[2]. P. Zhao, S. Zhao, J.-H. Xue, W. Yang, and Q. Liao, “The neglected background cues can facilitate finger vein recognition,” Pattern Recognit., vol. 136, no. 109199, p. 109199, 2023.

[3]. A. E. J. Bulstra and Machine Learning Consortium, “A machine learning algorithm to estimate the probability of a true scaphoid fracture after wrist trauma,” J. Hand Surg. Am., vol. 47, no. 8, pp. 709–718, 2022.

[4]. M. S. Hossain and M. S. Miah, “Machine learning-based malicious user detection for reliable cooperative radio spectrum sensing in Cognitive Radio-Internet of Things,” Machine Learning with Applications, vol. 5, no. 100052, p. 100052, 2021.

[5]. M. Crosby, “Building thinking machines by solving animal cognition tasks,” Minds Mach. (Dordr.), vol. 30, no. 4, pp. 589–615, 2020.

[6]. S. M. AlAli and J. M. Al Smady, “Validity and reliability of a Jordanian version of the Adaptive Behavior Assessment System (ABAS-II) in identifying adaptive behavior deficits among disabled individuals in Jordan,” J. Educ. Psychol. Stud. [JEPS], vol. 9, no. 2, pp. 248–261, 2015.

[7]. F. A. Dorça, L. V. Lima, M. A. Fernandes, and C. R. Lopes, “Comparing strategies for modeling students learning styles through reinforcement learning in adaptive and intelligent educational systems: An experimental analysis,” Expert Syst. Appl., vol. 40, no. 6, pp. 2092–2101, 2013.

[8]. S. Prasomphan, “Toward fine-grained image retrieval with adaptive deep learning for cultural heritage image,” Comput. Syst. Sci. Eng., vol. 44, no. 2, pp. 1295–1307, 2023.

[9]. C. Qu, Q. Yu, P. Houston, R. Conte, A. Nandi, and J. Bowman, “Many-body Δ-Machine Learning brings the accuracy of conventional force field to coupled cluster: application to the TTM2.1 water force field,” Research Square, 2022.

[10]. Y. Cho and KDI국제정책대학원, “Effects of AI-based personalized adaptive learning system in higher education,” J. Korean Assoc. Inf. Educ., vol. 26, no. 4, pp. 249–263, 2022.

[11]. R. L. Blomeyer Jr, “Instructional policy and the development of instructional computing: Maintaining adaptive educational programs,” Educ. consid., vol. 13, no. 3, 1986.

[12]. X. Xiang and S. Foo, “Recent advances in Deep Reinforcement Learning applications for solving partially observable Markov Decision Processes (POMDP) problems: Part 1—fundamentals and applications in games, robotics and natural language processing,” Mach. Learn. Knowl. Extr., vol. 3, no. 3, pp. 554–581, 2021.

[13]. A. Nahhas, A. Kharitonov, and K. Turowski, “Deep reinforcement learning techniques for solving hybrid flow shop scheduling problems: Proximal policy optimization (PPO) and asynchronous advantage actor-critic (A3C),” in Proceedings of the Annual Hawaii International Conference on System Sciences, 2022.

[14]. A. A. Untila, ITMO University, N. N. Gorlushkina, and ITMO University, “Conceptual models of computer games in the tasks of managing the involvement of students in the learning process,” Economics. Law. Innovaion, pp. 48–55, 2022.

[15]. H. Apriyanto et al., “The development of real-time monitoring and managing information system for digitalization of plant collection data in Indonesian Botanical Garden,” aisthebest, vol. 7, no. 1, pp. 16–30, 2022.

[16]. L. Xu, X. Han, K. Jiao, and T. Gao, “Research on the integration and optimization of MOOC teaching resources based on deep reinforcement learning,” Int. J. Contin. Eng. Educ. Life Long Learn., vol. 1, no. 1, p. 1, 2023.

[17]. P. W. Cardon, H. Ma, and C. Fleischmann, “Recorded business meetings and AI algorithmic tools: Negotiating privacy concerns, psychological safety, and control,” Int. J. Bus. Commun., p. 232948842110370, 2021.

[18]. H. Curiel and A. Poling, “Web-based stimulus preference assessment and reinforcer assessment for videos: Web-based preference and reinforcer assessment,” J. Appl. Behav. Anal., vol. 52, no. 3, pp. 796–803, 2019.

[19]. J. E. de Aguilar-Nascimento, “Fundamental steps in experimental design for animal studies,” Acta Cir. Bras., vol. 20, no. 1, pp. 2–8, 2005.

[20]. R. Young, “Discriminative stimulus effects of an imidazolidine-derived appetite suppressant,” Med. Chem. Res., 2022.

[21]. H. D. Kimmel and H. Lachnit, “The Rescorla-Wagner theory does not predict contextual control of phasic responses in transswitching,” Biol. Psychol., vol. 27, no. 2, pp. 95–112, 1988.

[22]. A. Sharma, S. Tokekar, and S. Varma, “Actor-critic architecture based probabilistic meta-reinforcement learning for load balancing of controllers in software defined networks,” Autom. Softw. Eng., vol. 29, no. 2, 2022.

[23]. I. N. Yazid and E. Rachmawati, “Autonomous driving system using proximal policy optimization in deep reinforcement learning,” IAES Int. J. Artif. Intell. (IJ-AI), vol. 12, no. 1, p. 422, 2023.

[24]. M. Böck and C. Heitzinger, “Speedy categorical distributional reinforcement learning and complexity analysis,” SIAM Journal on Mathematics of Data Science, vol. 4, no. 2, pp. 675–693, 2022.

[25]. S. Tufenkci, B. Baykant Alagoz, G. Kavuran, C. Yeroglu, N. Herencsar, and S. Mahata, “A theoretical demonstration for reinforcement learning of PI control dynamics for optimal speed control of DC motors by using Twin Delay Deep Deterministic Policy Gradient Algorithm,” Expert Syst. Appl., vol. 213, no. 119192, p. 119192, 2023.

[26]. Y. T. Kim and S. Y. Han, “Cooling channel designs of a prismatic battery pack for electric vehicle using the deep Q-network algorithm,” Appl. Therm. Eng., vol. 219, no. 119610, p. 119610, 2023.

[27]. C. Wernz, “Multi-time-scale Markov decision processes for organizational decision-making,” EURO j. decis. process., vol. 1, no. 3–4, pp. 299–324, 2013.

[28]. C. A. Duncan, M. T. Goodrich, and E. A. Ramos, “Efficient approximation and optimization algorithms for computational metrology,” Comput. Stand. Interfaces, vol. 21, no. 2, pp. 189–190, 1999.

[29]. J. Bradley, D. E. Pooley, and W. Kockelmann, “Artifacts and quantitative biases in neutron tomography introduced by systematic and random errors,” J. Instrum., vol. 16, no. 01, pp. P01023–P01023, 2021.
[30]. S. K. Tiwari, L. A. Kumaraswamidhas, and N. Garg, “Time-series prediction and forecasting of ambient noise levels using deep learning and machine learning techniques,” Noise Control Eng. J., vol. 70, no. 5, pp. 456–471, 2022.
[31]. Q. Li, X. Meng, F. Gao, G. Zhang, and W. Chen, “Approximate cost-optimal energy management of hydrogen electric multiple unit trains using double Q-learning algorithm,” IEEE Trans. Ind. Electron., vol. 69, no. 9, pp. 9099–9110, 2022.
[32]. C. Lee, J. Jung, and J.-M. Chung, “Intelligent dual active protocol stack handover based on double DQN deep reinforcement learning for 5G mmWave networks,” IEEE Trans. Veh. Technol., vol. 71, no. 7, pp. 7572–7584, 2022.
[33]. N. V. Varghese and Q. H. Mahmoud, “A hybrid multi-task learning approach for optimizing deep reinforcement learning agents,” IEEE Access, vol. 9, pp. 44681–44703, 2021.
[34]. S. Antunović and D. Vukičević, “Evaluating topological ordering in directed acyclic graphs,” Electron. J. Graph Theory Appl., vol. 9, no. 2, p. 567, 2021.
[35]. A. Kushwaha and T. J. Dhilip Kumar, “Benchmarking PES-Learn’s machine learning models predicting accurate potential energy surface for quantum scattering,” Int. J. Quantum Chem., vol. 123, no. 1, 2023.
[36]. S. Lohani, J. Lukens, R. T. Glasser, T. A. Searles, and B. Kirby, “Data-Centric Machine Learning in Quantum Information Science,” Mach. Learn.: Sci. Technol., 2022.
[37]. V. Sethi and S. Pal, “FedDOVe: A Federated Deep Q-learning-based Offloading for Vehicular fog computing,” Future Gener. Comput. Syst., vol. 141, pp. 96–105, 2023.
[38]. D. Jiménez, A. Angulo, A. Street, and F. Mancilla-David, “A closed-loop data-driven optimization framework for the unit commitment problem: A Q-learning approach under real-time operation,” Appl. Energy, vol. 330, no. 120348, p. 120348, 2023.
[39]. H. S. Yaseen and A. Al-Saadi, “Q-learning based distributed denial of service detection,” Int. J. Electr. Comput. Eng. (IJECE), vol. 13, no. 1, p. 972, 2023.
[40]. G. Shi et al., “Risk-aware UAV-UGV rendezvous with Chance-Constrained Markov Decision Process,” arXiv [cs.RO], 2022.
[41]. C. Wu, W. Bi, and H. Liu, “Proximal policy optimization algorithm for dynamic pricing with online reviews,” Expert Syst. Appl., vol. 213, no. 119191, p. 119191, 2023.
[42]. Y. Liu, Q. Ye, J. Escribano-Macias, Y. Feng, E. Candela, and P. Angeloudis, “Routing planning for last-mile deliveries using mobile parcel lockers: A Hybrid Q-Learning Network approach,” arXiv [cs.AI], 2022.