International Journal of Scientific Research in Computer Science, Engineering and Information Technology
ISSN : 2456-3307 (www.ijsrcseit.com)
doi : https://doi.org/10.32628/CSEIT2390147

A Review of Current Perspective and Propensity in Reinforcement Learning (RL) in an Orderly Manner

Shweta Pandey*[1], Rohit Agarwal[2], Sachin Bhardwaj[3], Sanjay Kumar Singh[4], Dr. Yusuf Perwej[5], Niraj Kumar Singh[6]

[1*]Scholar, B.Tech, Computer Science & Engineering, Ambalika Institute of Management & Technology, Lucknow, India

[2-6]Assistant Professor, Department of Computer Science & Engineering, Ambalika Institute of Management & Technology, Lucknow, Uttar Pradesh, India

ARTICLEINFO

Article History:

Accepted: 01 Feb 2023
Published: 25 Feb 2023

Publication Issue
Volume 10, Issue 1
January-February-2023

Page Number
206-227

ABSTRACT

Reinforcement learning is an area of Machine Learning. The three primary types of machine learning are supervised learning, unsupervised learning, and reinforcement learning (RL). Pre-training a model on a labeled dataset is known as supervised learning. The model is trained on unlabeled data in unsupervised learning, on the other hand. Instead of being driven by labels, RL is motivated by assessing feedback. By interacting with the environment and choosing the best course of action in each circumstance in order to maximize the reward, the agent learns the best way to solve sequential decision-making issues. The RL agent chooses how to carry out tasks on its own. Furthermore, since there are no training data, the agent learns by gaining experience. In order to make subsequent judgments, RL aids agents in efficiently interacting with their surroundings. In this essay, the state-of-the-art RL is thoroughly reviewed in the literature. Applications for reinforcement learning (RL) may be found in a wide range of industries, including smart grids, robots, computer vision, healthcare, gaming, transportation, finance, and engineering.

Keywords : Machine Learning, Reinforcement Learning, Fictitious Play, Multi-Agent SARSA Learning, Friend-or-Foe Q-Learning (FFQ), Nash-Q Learning.

I. INTRODUCTION

Since a few years ago, technology has played a significant role in our day-to-day lives, [1] and in some ways; we as a society are depending on it to our advantage and comfort. In current information technology era, also known as the era of smart innovations, every single person is somehow connected to the invention, whether consciously or unconsciously. One of the most significant innovations in recent years is artificial intelligence (AI) [2]. Artificial intelligence is becoming more and

more necessary. Three strands ran through the early history of reinforcement learning [3], the first dealt with learning by trial and error; the second, with the issue of optimum control; and the third, which emerged later and was built on concepts from the first two, with temporal-difference approaches [4]. The contemporary discipline of reinforcement learning was created in the late 1980s when all of these came together [5]. Today, together with supervised and unsupervised learning, reinforcement learning has established itself as one of the three primary machine learning paradigms [6]. The machine learning subfield of supervised learning has received the most attention and study. In supervised learning, a teacher or supervisor from outside the system assigns labels to a training set of data [7] and decides what actions the system should take in each case. The system must generalise its answers in order to respond appropriately in situations that are not represented in the training examples. By adding more training instances, the supervised learning system's performance improves [8]. The classification, object identification, picture captioning, regression, and labelling are a few examples of supervised learning issues.

Although this kind of education is crucial, it is insufficient for interactive settings since it is impossible to collect labelled data that are both accurate and representative. In interactive settings, learning will be more effective if the system is able to draw lessons from its own mistakes. Finding latent structure in a collection of unlabelled data is the goal of unsupervised learning [10]. Clustering, feature learning, dimensionality reduction, and density estimation are some instances of unsupervised learning. The fact that reinforcement learning does not use labelled data may make it appear to be unsupervised learning, but this is not the case because the goal of reinforcement learning is to maximise rewards rather than uncover hidden structure [11]. Researchers from a variety of disciplines are interested in reinforcement learning (RL), one of the most amazing subfields of machine learning [12]. Consecutive decisions may be made because RL aids agents in efficiently interacting with their surroundings. It encourages behavioral decision-making by utilizing interaction experience and evaluating feedback afterward [13]. The agent decides on a course of action depending on the current state and is rewarded or given feedback, along with a new state. Then, instead of learning from instructions, it makes an effort to discover an ideal course to take in order to maximize the reward it receives over time. A game-like scenario [14] is presented to an artificial intelligence during reinforcement learning. In order to solve the problem, the computer uses trial and error. The artificial intelligence is rewarded or punished for the steps it takes to make the machine accomplish what the programmer desires. To maximize the overall return is its aim.

II. Related Work

Reinforcement learning is the process of learning through interacting with the environment, trying new things, failing a lot, and succeeding a lot all while attempting to get the most out of the rewards you get. The contemporary discipline of reinforcement learning was created in the late 1980s when all of these came together [15]. One of the three primary machine learning paradigms at the moment is reinforcement learning, along with Learning that is supervised and unsupervised [16,17]. The performance of current RL algorithms has been greatly enhanced by recent deep learning integrations. Researchers coupled deep learning with RL to produce deep reinforcement learning, which is practically unbeatable in a number of video Atari games [18] [19], in order to develop a clever agent. Combining reinforcement learning and deep learning, Deepmind learnt to play 49 different games from self-play and the game's score using the same algorithm without adapting to a specific game, and it eventually

reached human level on Atari games. A Deep Q Network was used to map the raw screen pixels. Professional Atari player software was described in depth by DeepMind in 2013 [20]. The optimal machine learning paradigm is reinforcement learning (RL), whose algorithms quickly pick up information from their interactions with the environment. Deep reinforcement learning (DRL), whose algorithms are capable of learning more difficult tasks, was also created by combining deep neural networks (DNNs) with RL algorithms [22].

Google bought DeepMind later in the start of 2014 [23]. A backgammon-playing computer programme was created by IBM employee Gerald Tesauro utilising reinforcement learning. Human gamers could not compete with Gerald's software [24]. But until DeepMind taught its AlphaGo algorithm using reinforcement learning and beat Go's world champion, Lee Sedol, expanding this success to more challenging games was challenging. Go is a traditional Chinese game that was far more difficult for computers to learn to play [25]. AlphaGo, a software that defeated one of the greatest professional Go players of all time in 2016, was developed by Google's DeepMind subsidiary to continue the success of deep RL [26]. A deep quality-learning networks (DQNs) technique was also used by Carta et al. [27] to make accurate stock market forecasts [28][29]. One of the remarkable successes in the subject of reinforcement learning is AlphaZero, which in 24 hours of self-play reached a superhuman level in three different games, Go, Shogi (Japanese chess), and Chess. The innovation is that it does not require game-specific tweaking and reuses the same hyper parameters for all games [30]. The Markov Decision Process (MDP), a notion initially proposed by Bellman R. E. in 1957 [31], is used to represent this interaction. This concept reduces the interaction to three signals, state (the environment's current situation); action (the agent's operation or decision based on the state and its experience); and reward (the environment's numerical feedback letting the agent know whether their action was successful or unsuccessful) [32].

In the realm of control, Peter Abbeel and Andrew Ng achieved automated helicopter flight in 2006. To assist in developing a model of helicopter dynamics and a reward function, they had a pilot operate the chopper. In order to create an optimum [33] controller for the resultant model and reward function, they next employed reinforcement learning. Another research by Carta et al. [36] suggested an ensemble of deep Q-learning classifiers with various environmental experiences as a first step toward completing such a job. In order to create the next generation of intelligent self-driving cars, Google, Uber, and Tesla are accelerating their deep reinforcement learning research. The authors of research [37] also performed a survey on RL that was affected by natural language processing (NLP) via picking up information from textual domains, playing text games, and paying attention to directions. This study suggests that RL tasks can be accomplished using NLP strategies.

III. Reinforcement Learning (RL)

The advancement of machine learning technology in recent years has opened up new opportunities for the resolution of challenging issues. Modern machine learning techniques like convolutional neural networks (CNNs) [38] and recurrent neural networks (RNNs) [39] have proven their effectiveness in applications including natural language processing and image categorization. By speeding up the related computational processes, the development of hardware technologies such as GPU computing has also accelerated the evolution of machine learning technologies.

As a result, solutions for traffic engineering and routing with the use of machine learning have been suggested. In a nutshell, figure 1 illustrates the three

primary categories of machine learning technologies, supervised learning, unsupervised learning, and reinforcement learning. Due of these factors, a distinct strategy known as reinforcement learning has lately attracted a lot of attention (RL).

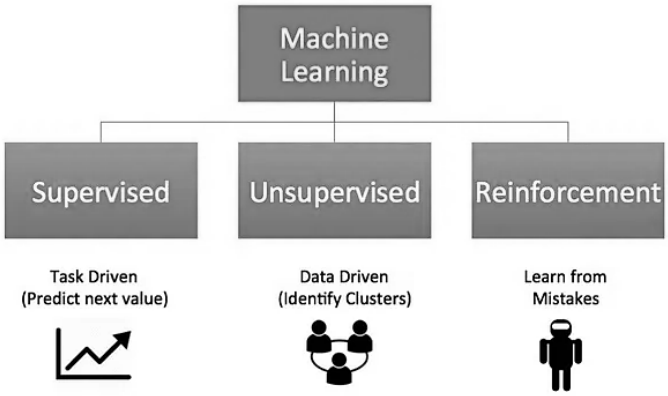


Figure 1. The Types of Machine Learning

Both a neural network type and a neural network substitute, reinforcement learning is neither. Instead, it uses an orthogonal strategy to tackle a separate, trickier issue. Strong machine learning systems are produced by the combination of supervised learning with dynamic programming, or reinforcement learning. Due of its generality, reinforcement learning is appealing to many academics. The computer is simply given a task to do in real life. Following interactions with its surroundings that include trial and error, the computer learns how to accomplish that aim. As a result, many academics are working on this type of artificial intelligence and are thrilled about the prospect of tackling issues that have hitherto been intractable. By executing actions and seeing the outcomes of those actions, an agent learns how to behave in a given environment via reinforcement learning, a feedback-based machine learning approach. The agent receives positive feedback for each positive activity, and negative feedback or a penalty for each poor action.

- **Positive Reinforcement:** Positive reinforcement is when behaviour is reinforced by an event that occurs as a result of that behaviour, increasing its strength and frequency. In other words, it influences behaviour in a favourable way. The following benefits of reinforcement learning.

  - ✓ Increases Performance
  - ✓ Maintain Change for a Long Time
  - ✓ Too Much Reinforcement Can Diminish Results by Creating An Overload Of States

- **Negative Reinforcement** – Negative **reinforcement** is the strengthening of a behaviour as a result of stopping or avoiding a negative state. Reinforcement learning has several benefits [40].

  - ✓ Improves behaviour
  - ✓ Show disobedience to a certain level of performance
  - ✓ It only offers enough to satisfy the bare minimum of behaviour.

In contrast to supervised learning, reinforcement learning uses feedback to autonomously train the agent without the need of labelled data. The agent can only learn from its experience because there is no labelled data. Reinforcement learning has distinct objectives than unsupervised learning.

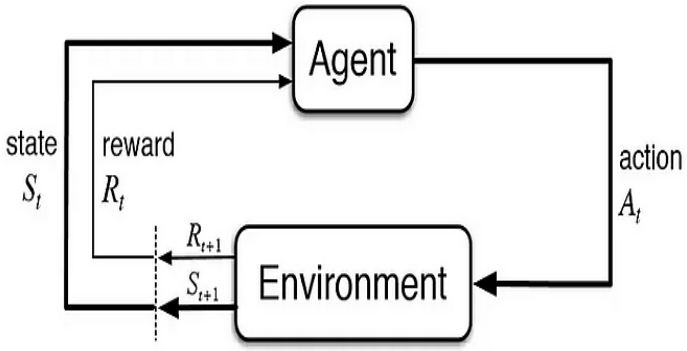


Figure 2. The Reinforcement Learning

Although finding similarities and differences between data points is the aim of unsupervised learning, the aim of [41] reinforcement learning is to identify an appropriate action model that would maximise the overall cumulative reward of the agent. The action-reward feedback loop of a general RL model is shown in figure 2.

3.1 Reinforcement Learning Terminology

- **Agent ():** An entity that can perceive/explore the environment and act upon it.
- **Environment ():** In RL, we take the assumption that the environment is stochastic, or essentially random.
- **Action ():** An agent's actions are its movements inside its surroundings.
- **State ():** Returns the state of the environment following each action the agent has done.
- **Reward ():** A response from the environment that the agent can use to gauge its performance [41].
- **Policy ():** Policy is a technique used by the agent to determine what to do next depending on the situation as it is right now.
- **Value ():** It is supposed to be opposite of the short-term reward and to be long-term calibrated with the discount factor.
- **Q-value ():** It is mostly similar to the value, but it takes one additional parameter as a current action (a).
- **Agent ():** A thing that can see and investigate its surroundings and take appropriate action.

3.2 Methods for Using Reinforcement Learning

The two subcategories of reinforcement strategies that are included in [42]'s taxonomy are model-based and model-free approaches.

3.2.1 Model-based Reinforcement Learning

Using the transition function and reward function, the agent calculates the best course of action in this learning process. Future states are attempted to be anticipated using this process. Model-based techniques rely heavily on planning as their main element. In order to forecast the subsequent state using the policy network, starting states must be used. The Deep Q Network is the most well-known illustration of this learning [43]. Currently, the main recommendation algorithm is a DQN. A generative adversarial network can be used to mimic user-agent interaction for offline policy learning using model-based reinforcement learning techniques.

3.2.2 Model-free Reinforcement Learning

This learning technique allows for the construction of optimum policies without estimating interactions between states and reward functions. Future conditions or rewards cannot be predicted. Methods without models rely significantly on learning. This eliminates the need to forecast the following state using the initial states. Reinforcement learning is exemplified by the actor-critic strategy. Model-free RL might result from a discrete mathematical approach to a particular problem [43].

- **Value-based:** The optimal value function, which is the maximum value at a state under any policy, is what the value-based method is going to discover.

- **Policy-based:** By employing a policy-based approach rather of a value function, the best strategy for maximising future rewards may be found. In this method, the agent seeks to implement a policy in a way that each action serves to maximise the reward in the future. The two primary categories of policies in the policy-based approach are:

  - ✓ **Deterministic:** Any state's policy (π) results in the same action.
  - ✓ **Stochastic:** In this strategy, probability governs the action that is generated.

3.3 Elements of Reinforcement Learning

Following are the four key components of reinforcement learning:

1. Policy
2. Reward Signal
3. Value Function
4. Model of the environment

1) Policy: A policy is the way an agent acts at a specific moment in time. It connects the perceived environmental conditions to the responses to those conditions. The fundamental component of the RL is a policy since only a policy may specify how an agent will behave. It could be a straightforward function or lookup table in certain situations, but comprehensive computing like a search procedure might be necessary in others.

2) Reward Signal: The reward signal establishes the purpose of reinforcement learning [44]. The environment immediately transmits a signal known as a reward signal to the learning agent at each state. These incentives are offered in accordance with the agent's successful and unsuccessful acts. The agent's principal goal is to increase the overall quantity of incentives for doing the right thing. The policy can be altered by the reward signal. For instance, if an action chosen by the agent yields a poor reward, the policy may be altered to choose different behaviors in the future.

3) Value Function: The value function informs an agent about the merits of a given scenario and course of action, as well as the potential rewards. A value function defines the excellent condition and action for the future, but a reward indicates the immediate signal for each good and poor activity. The reward is a necessary component of the value function since value cannot exist without it. To reap additional advantages, one uses value estimation.

4) Model: The model, which imitates the behavior of the environment, is the last component of reinforcement learning. One can draw conclusions about the behavior of the environment using the model. A model, for instance, can forecast the subsequent state and reward if a state and an action are provided.

The model is used for planning, which means it offers a mechanism to choose a course of action by taking into account all potential outcomes before those outcomes actually occur. The term "model-based approach" refers to methods for tackling RL issues with the use of models. In contrast, a model-free strategy is one that doesn't employ a model.

3.4 Markov Decision Process (MDP)

The typical example of reinforcement learning is a Markov Decision Process (MDP). This is the standard scientific paradigm for single agent reinforcement learning. The challenges in this paradigm involve making decisions repeatedly, where each state requires choosing an activity by going to the relevant framework [45]. The decision-making process in discrete, stochastic, sequential contexts is modeled by Markov decision processes (mdps). The core of the paradigm is that a decision maker, or agent, resides in a setting that changes state arbitrarily in response to the decision-choice maker's of actions. The environment's state influences both the agent's immediate reward and the likelihood of subsequent state shifts. The agent's goal is to make decisions that will maximize a long-term indicator of total reward. Large planning issues from artificial intelligence, operations research, economics, robotics, and the behavioral sciences may be described and addressed using effective algorithms for mdps based on dynamic programming, linear programming, and more recently, compact representations [46].

These problems are prevalent throughout stochastic control hypothesis, and their roots are understandable. Reinforcement Learning problems may be represented by a quadruple (S, A, P, R) Markov Decision Process (MDP), where S is the

collection of states and A(s) is the set of activities available in states., A(s) is the set of activities accessible in state S, P : S × A × S → [0, 1] is a progress dispersion that establishes the likelihood of entering the particular state following the execution of a given move in a given state. R : S × A → r is the action that denotes the immediate reward during the execution of a given move in a given state. Finding out how to choose activities that increase the cumulative amount of compensations over time, r0 + γr1 + γ2r2 + …. where γ ∈ (0, 1) is the markdown factor, which determines how strongly the rewards are weighted in comparison to future compensation. If the immediate reward is restricted, a markdown factor of γ < 1 ensures that the future limited return Rt is dependably a limited number if the prompt reward is limited. If γ < 1 then the reward is not exponentially exactly same as the reward received at the initial stage then MDP is known as the limited reward MDP and when γ = 1 then the value of MDP is known as undiscounted. Robot control, manufacturing, traffic signal [47] control, and other disciplines requiring sequential decision making and typified by unpredictable state transitions can all benefit from MDPs.

IV. Reinforcement Learning Algorithms

There are many reinforcement learning algorithms. We'll focus on those that are crucial to understand right now. These algorithms fall under certain categories. These algorithms are model-based and model-free. We are further dividing them into on-policy and off-policy categories. We should thus have a model in model-based algorithms that would learn from current activities and from state changes. It would become impractical after a while since it would need to keep all the status and action data in the memory. With contrast, there is no need to worry about a model taking up a lot of space in model-free algorithms. We don't need to save the states and actions because this algorithm operates on a trial-and-error basis.

4.1 Nash-Q Learning Algorithm

The Q-learning method we suggest shares many characteristics with conventional single-agent Q-learning, but it varies in one key area: how to use the Q-values of the subsequent state to update those of the present state. Our multi-agent Q-learning algorithm updates with future Nash equilibrium payoffs, while single-agent Q-learning updates are focused on the agent's own greatest payoff; our multi-agent Q-learning method updates with future Nash equilibrium payoffs. The agent must monitor not just its own reward but that of others in order to understand these Nash equilibrium payoffs. The outcomes will rely on how closely the proxy is matched to genuine rewards in contexts where this is not possible. It is necessary to identify some observable proxy for other-agent rewards. Our learning agent, indexed by I forms an arbitrary estimate at time 0 and learns about its Q-values from there [48]. One straightforward assumption would be to set Qi 0(s, a^1, . . . , a^n) = 0 for all s ∈ S, $a^1 \in A^1$, . . . , $a^n \in A^n$. At Every time t, agent I evaluates the situation and acts accordingly. After that, it monitors its own reward, all other agents' activities, other agents' rewards, and the newly created state s'. It then calculates a Nash equilibrium $\pi^1(s') \cdots \pi^n(s')$ for the stage game ($Q^1_t(s')$, . . . , $Q^n_t(s')$), and updates its Q-values according to

$$Q^i_{t+1}(s,a^1,\ldots,a^n) = (1-\alpha_t)Q^i_t(s,a^1,\ldots,a^n) + \alpha_t\left[r^i_t + \beta Nash Q^i_t(s')\right],$$

Where

$$Nash Q^i_t(s') = \pi^1(s')\cdots\pi^n(s')\cdot Q^i_t(s'),$$

In general, different strategies for choosing from among many Nash equilibria will produce different results. NashQ I t (s') is agent i's payoff in state's' for the selected equilibrium. Note that $\pi^1(s') \cdots \pi^n(s')\cdot Q^i_t(s')$ is a scalar. The summary of this learning method is shown in table 1.

Table 1: The Nash Q-learning algorithm

Initialize:
Let $t = 0$, get the initial state s_0.
Let the learning agent be indexed by i.
For all $s \in S$ and $a^j \in A^j$, $j = 1,\ldots,n$, let $Q_t^j(s,a^1,\ldots,a^n) = 0$.
Loop
Choose action a_t^i.
Observe $r_t^1,\ldots,r_t^n; a_t^1,\ldots,a_t^n$, and $s_{t+1} = s'$
Update Q_t^j for $j = 1,\ldots,n$
$Q_{t+1}^j(s,a^1,\ldots,a^n) = (1-\alpha_t)Q_t^j(s,a^1,\ldots,a^n) + \alpha_t[r_t^j + \beta NashQ_t^j(s')]$
where $\alpha_t \in (0,1)$ is the learning rate, and $NashQ_t^k(s')$ is defined in
Let $t := t+1$.

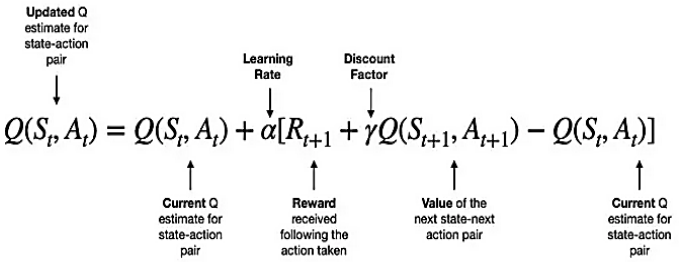
In order to calculate the Nash equilibrium ($\pi^1(s')$, . . . , $\pi^n(s')$), agent i would need to know $Q^1{}_t$ (s'), . . . , $Q^n{}_t$ (s'). Agent I must also learn about the Q-values of other agents because such information is not provided [48]. At the start of the game, Agent I makes hypotheses regarding certain Q-functions, such as, $Q^j{}_0(s, a^1, \ldots, a^n) = 0$ for all j and all s, $a^1, \ldots, a^n$. Agent i watches as the game progresses. That information can then be used to up- date agent i's conjectures on other agents' Q-functions. Agent i updates its beliefs about agent j's Q-function, according to the same updating rule, it applies to its own,

$$Q_{t+1}^j(s,a^1,\ldots,a^n) = (1-\alpha_t)Q_t^j(s,a^1,\ldots,a^n) + \alpha_t\left[r_t^j + \beta NashQ_t^j(s')\right]$$

Note that $\alpha t = 0$ for (s, a^1, . . . , a^n) 6 = (st , $a^1{}_t$, . . . , $a^n{}_t$). Because of this, above does not update every entry in the Q-functions. Only the entry that reflects the current state and the agents' chosen actions is updated. Asynchronous updating refers to such updating. Hu and Wellman [48] went on to emphasize that under some circumstances, this learning algorithm integrates with Nash equilibrium techniques in multi-player situations and adds extra expectations to the payout structures.

4.2 Multi-Agent SARSA Learning Algorithm

Because Nash-Q and Minimax-Q learning algorithms modify the max operator of a particular Q-learning algorithm by their dominant reaction, which is known as the Nash equilibrium policy, these algorithms are really referred to as off-strategy Reinforcement Learning algorithms. No matter what strategy is to be used, an off-approach learning algorithm used in reinforcement learning continually tries to combine the best Q values of optimum strategies. The 1998 SARSA algorithm, according to Sutton, is one of the on-policy Reinforcement Learning algorithms that make an effort to converge to ideal Q values of the strategy being used at the time [49]. One aspect that sets SARSA apart from Q-Learning is that it uses an on-policy algorithm (off-policy algorithm). On-policy refers to the usage of the same acting policy for the agent throughout training as well as updating the value function (updating policy). In the meanwhile, we apply several rules for acting and updating when using the off-policy technique.



Q is the value function, and the term on the left Q(St,At) is the new value for the specific state-action pair. Note, S refers to State, and A refers to Action. On the right-hand side of the equation, we find the same term Q(St,At), which, in this case, is the current value for that same state-action pair. To update the current value, we take the reward (Rt+1) following the action taken by the agent, add the value for the next state-next action pair (St+1,+1) discounted by gamma, and subtract the current value Q(St,At).

So, the terms in the square brackets produce a positive, zero or negative value, which leads to either increase, no change or a decrease in the new value of Q(St,At). Note that we also apply a learning rate (alpha) to control the "size" of each update. Due to SARSA's usage of the Temporal Difference (TD) technique, the algorithm will continue to update the

Q-table following each step until the maximum number of iterations are reached or the solution converges to an ideal one [49].

4.3 Policy Hill Climbing (PHC) Algorithm

A reinforcement learning technique called Policy Hill Climbing (PHC) extends Q-learning to teach probabilistic rules for multi-agent games. Similar to the fake play method, this algorithm updates the Q values, but it also maintains the mixed policy, also known as the stochastic policy, by performing hill-climbing in the space of these policies. By accepting the concept of Win or Learn Fast and exploiting the variable learning rate, Bowling and Velso, 2002 suggested a PHC method called WoLF (Win or Learn Fast). If the agent is not acting carefully and effectively while utilizing this technique, it results quickly in the agent learning. Since the separate agents' evolving tactics won't be too adapted by this shift in learning rates, the convergence will benefit from it. By utilizing a variable learning rate and the concept of Win or Learn Fast (WoLF), Bowling and Veloso [50] suggested a WoLF-PHC algorithm. The agent may learn rapidly when doing poorly and cautiously when performing well as a result of the WoLF principle. By preventing the learning rates from over fitting to the other agents' shifting policies, the change in the learning rates will aid in convergence. The WoLF-PHC algorithm is appealing at this time. Although [51] gave several instances ranging from MGs to zero-sum and general-sum SGs, a comprehensive demonstration for the convergence features has not yet been offered. Since the learning elements of other agents in the non-Markovian environment are completely ignored, the WoLF-PHC algorithm is still not a multi-agent version of the PHC algorithm. As a result, only stationary strategies are being used by the other agents for it to be ratio- nal and fair. When the WoLF approach is used, the convergence may also become extremely sluggish [49].

4.4 Friend-or-Foe Q-Learning (FFQ) Algorithm

Each and every agent in the framework is classified as either a "friend" or a "foe" in the FFQ algorithm. For RL in general-sum SGs, Littman [51] created the Friend- or-Foe Q-learning (FFQ) method. The key concept is that every agent in the system is classified as either a "friend" or a "foe." As a result, the equilibrium can be categorized as either adversarial or coordinated. The FFQ-learning can offer a greater convergence guarantee than Nash-Q learning. The following findings were provided by Littman [52] to demonstrate the convergence of the FFQ-learning algorithm. Theorem 3 If the game has an adversarial equilibrium, Foe-Q learns values for a Nash equilibrium policy, and Friend-Q learns values for a Nash equilibrium policy. Regardless of opposition behaviour, this is true. Although the FFQ-learning algorithm's convergence feature has been enhanced compared to the Nash-Q learning algorithm, Friend-or-Foe ideas have not yet been fully addressed [52]. The FFQ-learning method does not necessitate learning estimates to the opponents' Q-functions, in contrast to the Nash-Q learning process. The agent must be aware of how much equilibrium there are in the game and that each equilibrium is known to be either coordinating or adversarial in advance in order for the FFQ-learning to be applied. Finding a Nash equilibrium or determining if a Nash equilibrium is adversarial or cooperative cannot be done with FFQ-learning alone. Similar to Nash-Q learning, FFQ learning is inapplicable to systems with no coordination or adversarial equilibrium.

4.5 Minimax-Q Learning Algorithm

For zero-sum game scenarios, the learning player grows its payoffs under any conditions using the Minimax-Q learning method. For zero-sum games, Littman [53] introduced a Minimax-Q learning method where the learning player maximizes its payoffs in the worst case. The game's players have

opposing goals in mind. The Minimax-Q learning algorithm is essentially a value-function reinforcement learning technique. In the Minimax-Q learning, the player always tries to maximize its expected value when faced with the opponent's worst-case scenario for action. As a result, after learning, the player would become more cautious. Littman [53] employed straightforward linear programming to calculate the probability distribution or the player's best course of action. The Minimax-Q learning technique was initially presented in [53], which only provided empirical data on a straightforward zero-sum soccer SG game. Later publications [54] provide a thorough convergence proof that is included in the following theorem. An agent using the Minimax-Q learning algorithm will with probability one converge to the ideal Q-function in a two-player zero-sum multiagent SG environment. Additionally, if the limit equilibrium is unique, an agent following a GLIE (Greedy in the Limit with Infinite Exploration) strategy will converge in behavior with probability 1. As it may be used independently of the presence of an opponent, the Minimax-Q learning algorithm may offer a safe policy [55]. In the absence of knowing the opponent's policy, the Minimax-Q learning algorithm's policy can ensure that it obtains the maximum value feasible. Although the Minimax-Q learning approach exhibits several benefits in the two-player zero-sum multi-agent SG scenario, one specific disadvantage of this algorithm is that it learns extremely slowly since linear programming is required for each episode and state. Before the system converges, the use of linear programming considerably raises the cost of computing.

4.6 rQ-Learning Algorithm

The rQ -learning algorithm was created to handle issues with a vast search space. R-state and an r-action set must always be clearly stated at the beginning of this method. A so-called rQ-learning method was created by Morales [56] to handle situations involving a wide search area. An r-state and an r-action set must be established for this algorithm. A collection of first-order relationships, such as a goal in front, a team robot to the left, an opponent robot carrying the ball, etc., constitute an r-state. A collection of preconditions, a generalized action, and conceivably a set of post conditions describe an r-action. The following need must be met in order for an r-action to be specified correctly: if an r-action applies to one specific instance of an r-state, then it should apply to all instances of that r-state. Although the rQ-learning approach appears to be beneficial for dealing with problems involving huge search spaces, it may be quite challenging to identify an appropriate r-state and r-action set, particularly when one has little understanding about the concerned MRS. Additionally, there is no assurance that the defined r-actions in the r-state space are sufficient to establish an optimal sequence of primitive actions, and sub-optimal policies may be created [56].

4.7 Fictitious Play Algorithm

Fictitious play was first described by George W. Brown [57], who hypothesised that it would converge to the value of a zero-sum game. Julia Robinson [58] (yes, the Julia Robinson from Hilbert's tenth problem [59]) then established the convergence qualities of fictitious play. Games that are played often and where the play becomes better with time are said to have Fictitious play as its algorithm for learning in today's society. This interpretation states that hypothetical play continues as follows: in the ith round of play, take into account the opponent's historical distribution of play (i.e., the percentage of times each action was played in the first I 1 rounds), and play a best response to this distribution. The historical play distributions of the players are guaranteed to converge to a Nash equilibrium under specific circumstances, such as when the game is zero-sum [60], when the payoffs are generic and 2 2, when it

can be solved by iterated strict dominance [61], or when it is a weighted potential game [62]. (However, there are other games where the distributions under fake play do not converge [63]). This technique may also be used to create an algorithm for playing a game only once; all that is needed is to mimic what would occur in the game's repeated form. If both players used hypothetical play up to a predefined round r, the simulation would output the historical distribution of play as the strategy. In reality, when fake play was first presented, this interpretation was given to it, thus the term fictitious play; therefore this interpretation of fictitious play is not at all fresh. The algorithm known as fictional play is basic and ancient. Since then, several other, more complex algorithms have been developed, both for learning how to play games and for solving (or roughly solving) games (such as the approximation algorithms outlined above) (e.g.,[64],[65]).I think it is, at least for the factors listed below. Playing fictitiously is quite easy. This makes it an extremely adaptable algorithm that is often used. In comparison to the more complex algorithms, it also represents human behavior more convincingly. Finally, because of its simplicity, it is simple to examine (which may be connected to the fact that other algorithms do not provide results that are comparable). When the adversary is not (or may not be) adjusting and instead employing a set strategy, fictional gaming is a natural way to learn. This is particularly helpful in situations where the agent is unsure if she is genuinely up against an opponent or merely random nature [66].

The groups of artificial intelligence/multiagent systems, theoretical computer science, and "classical" game theory are among the many that are interested in solving games and/or learning how to play them. Although there are exceptions, it seems that these communities are primarily headed in distinct directions and that they are not usually aware of the work being done in the other areas. The work in this paper should be of interest to all of these communities: the community of artificial intelligence/multi-agent systems because of the useful properties of fictional play; the community of theoretical computer science because of the formal approximation guarantees; and the community of traditional game theory because the algorithm under study is a common one that seems practical for people to use.

V. Characteristics of Reinforcement Learning

The traits of reinforcement learning are covered in this section.

• **No Supervisor:** Reinforcement learning lacks a pre experience, in contrast to supervised learning [67]. There isn't a supplied example of a result or a targeted variable in it. It only learns from itself.

• **Sequential Decision Making:** Sequential Decision Making is a strategy or algorithm that uses the dynamics of the outside world to make decisions. It won't stop on its own and pushes back some of the problem until it can't be solved.

• **Time as a key element:** Time is always a key element in reinforcement learning.

• **Feedback:** Feedback is delayed and takes time. It never happens right away.

• **Data on subsequent actions:** The agent's activities determine the subsequent data that it gets.

VI. Challenges of Reinforcement Learning

As was already discussed, reinforcement learning employs the feedback mechanism to ensure the optimal actions are taken. As a result, it may be used to solve a variety of complicated issues and has found use in several fields [68]. However, it encounters several difficulties when applying what it has learned in a virtual environment to actual world issues.

- **Simulation:** Setting up the simulation environment, which is heavily reliant on the job

at hand, is one of the largest obstacles in reinforcement learning. In games like Chess, Go, or Atari, setting up the simulation environment is quite easy when the model needs to become superhuman.

- **Autonomous Car:** Building a realistic simulator is essential before allowing the autonomous vehicle drive on the road when it comes to a model that is able to operate one. The Reinforcement Learning model must determine how to brake or prevent a collision in a secure setting where the risk of losing even 100 automobiles is negligible.
- **Training Environment:** The challenging part is transferring the model from the training environment to the actual world. Another significant problem is scaling and modifying the neural network that controls the bot. There is no other method to communicate with the network but by using the rewards and punishments mechanism.
- **Optimum:** Reaching a local optimum, when the agent executes the task as-is but not in the necessary manner, is another significant difficulty. A good example is a "jumper" that jumps like a kangaroo instead of strolling as was expected of it.
- **Learning to Remember:** An observation seldom catches the entire environment condition that decides the appropriate course of action for many real-world jobs. An agent must consider both past and present observations in such partially viewable situations in order to choose the appropriate course of action. Consider an intelligent agent at work who assists a member of the customer care team in resolving a client issue, for instance [69]. The human employee could inquire about a billing matter from the client. That client may have a landline, a cell phone, and an online account. When a human asks, "What's the outstanding amount on the account?" the agent must remember the sequence of the discussion in order to recognize the account that the person is talking to. But understanding a solid policy is difficult when you have to remember everything that was said in a dialogue. Humans communicate in a topic-hopping manner, switching topics and looping back. While some information is extraneous, other information is crucial. Consequently, the task is to learn a condensed representation that only retains the essential data.
- **Limited Agent Freedom:** In practice, even when the task is clearly defined (with explicit reward functions), a key challenge is that it is frequently not possible to allow the agent to interact sufficiently freely and appropriately in the actual environment due to safety, financial, or time constraints.
- **Reality Gap:** There may be circumstances when the agent can only engage with a false simulation of the environment rather than the real world. The disparity between the useful real-world domain and the learning simulation is known as the reality gap.
- **Limited Observations:** In some circumstances, it may no longer be feasible to get fresh observations (e.g. the batch setting). Such situations might occur, for instance, in clinical trials, weather-dependent jobs, or trading marketplaces like stock markets [70].
- **Neural Network:** Modifying and scaling the neural network [71] that directs the agent presents another difficulty. Because rewards and punishments are the sole means of communication with the network, it is complicated. The major challenge associated with this is that this could lead to catastrophic forgetting or in other words, this might cause some old knowledge to get erased as it acquires new knowledge.

VII. The Benefits and Drawbacks of Reinforcement Learning

In this section, we are discussing benefits and drawbacks of reinforcement learning.

7.1 Benefits

It provides long-term outcomes that are very challenging to get and helps solve exceedingly complicated issues that traditional procedures are unable to tackle. This model mimics the way humans learn and hence exhibits excellence in every move. The model has the capacity to learn from mistakes and make corrections. Therefore, there is extremely little probability of making the same mistake again. It gains knowledge by experience; therefore a dataset is not required to direct its activities. It opens up the possibility for a thoughtful analysis of the situation-action relationship and generates the best behavior in a specific environment that maximizes performance. The model can correct the errors that occurred during the training process. Once an error is corrected by the model, the chances of occurring the same error are very less. It can create the perfect model to solve a particular problem. Robots can implement reinforcement learning algorithms to learn how to walk. In the absence of a training dataset, it is bound to learn from its experience. When interacting with the environment is the only way to learn about it, it might be helpful. Algorithms for reinforcement learning keep the balance between exploitation and exploration. Exploration involves testing new things to see if they are superior to what has already been tried. Exploitation is the practise of doing what has historically produced the best results. These balances are not achieved by other learning techniques.

7.2 Drawbacks

An overload from excessive reinforcement may taint the outcomes. Instead of basic problems, reinforcement learning [72] is favored for addressing difficult ones. It involves a lot of processing and a lot of data. High maintenance costs. The dimensionality curse severely restricts reinforcement learning for actual physical systems. The term "curse of dimensionality" refers, according to Wikipedia, to a number of phenomena that emerge when data is organized and analyzed in high-dimensional settings but do not do so in low-dimensional environments like the three-dimensional physical space encountered in daily life. Real-world sample's curse is another another drawback. Think about the situation of robot learning, for instance. The hardware for robots is typically quite costly, subject to deterioration, and in need of meticulous upkeep. The expense of fixing a robot system is high. Using reinforcement learning to solve straightforward issues is not recommended.

VIII. Limitations of Reinforcement Learning

Reinforcement Learning (RL) is capable of reaching a high level of proficiency. Since then, RL has drawn a lot of interest because of its capacity to excel at certain tasks like gaming and NLP. However, it has several drawbacks compared to recommendation systems, as with every machine learning [73] method. These include RL is not very generic. It frequently finds it difficult to adjust when new features or choices are made. On combinatorial decision spaces, RL does not scale well. Therefore, RL struggles to manage the volume of potential configurations when we have a large number of decisions to make, such as proposing a large number of movies on Netflix's home screen. There may be issues if the RL algorithm is archived for an offline recommendation [74]. Low signal-to-noise ratio data cannot be handled by RL. RL is an extremely potent model that can deduce complex relationships and rules from the data. RL will fit the noise if there are noisy characteristics. Long temporal spans are problematic for RL. Similar to the previously mentioned concerns, there are several

possibilities to fit noise while trying to optimize a long-term decision, which might cause RL to over fit if given a challenging optimization [75] assignment.

IX. Uses of Reinforcement Learning

Reinforcement learning may be used to solve any real-world issue where an agent must interact with an ambiguous environment in order to achieve a certain objective. The following are a few reinforcement learning uses.

9.1 Games

The capacity to play video games is typically seen as a result of intelligence, and Deep RL has been successfully applied to a wide variety of games with, surprisingly enough, only the input of pixels and no knowledge of the game's dynamics. RL has defeated specialists in a variety of fields, whether playing board games with perfect [76] information like go and chess or games with imperfect information like poker. It has also left its imprint on a number of video games, including old Atari games and difficult internet games like DOTA. The AI can work with human players to defeat opponents even in complicated multiplayer environments. It is crucial to keep in mind that RL may be extremely sample inefficient; for example, it took RL 45,000 years of gameplay simulation before it learned how to play DOTA.

9.2 Recommendation System

Recommendation systems [77] may benefit from the use of RL since it can handle complicated information structures. RL has been effectively incorporated into Alibaba and Taobao's e-commerce platforms. Facebook uses RL to determine which alerts are pertinent to send to its users. All of this is based on the straightforward and understandable principle of supporting the content, which is likely to increase user engagement.

9.3 Robots or Humanoids

At least in simulated situations, RL has demonstrated promising results in a variety of motor task automation and navigation applications. The creation of robots that can interact with humans like people do so is still a long way off in the future. The development of task-specific robots for pertinent industrial [78] applications has been accomplished, nonetheless. The future of RL in the area of industrial applications is bright. A startup specializing in deep RL for industrial systems was recently bought by Microsoft.

9.4 Natural language Processing

Reinforcement of Natural Language Processing As a result of learning's capacity to specialise in particular activities that it repeats repeatedly, goal-oriented chatbots have benefited significantly from NLP. These chatbots [79] are designed to respond to directed questions on a certain area. For instance, it may assist consumers with ticket booking, finding a reservation, and other tasks like text production and machine translation.

9.5 Finance

As a result of learning's capacity to specialise in particular activities that it repeats repeatedly, goal-oriented chatbots have benefited significantly from NLP. These chatbots are designed to respond to directed questions on a certain area. For instance, it may assist consumers with ticket booking, finding a reservation, and other tasks like text production and machine translation.

9.6 AlphaGo

The Chinese board game Go, which dates back 3,000 years, is one of the most difficult strategic games. Due to the fact that there are 10270 different board configurations that may be used, which is a huge increase over the [80] number of chess boards, it is more difficult. The best human Go player was defeated in 2016 by AlphaGo, a Go agent with real-world roots. It gained expertise by playing thousands of games with expert players, much like a human player would. The most recent RL-based Go agent has an advantage over human players in that it can learn by competing against itself.

9.7 Autonomous Driving

An autonomous driving system must perform multiple perception and planning tasks in an uncertain environment. Some specific tasks where RL finds application include vehicle path‧planning and motion prediction. Vehicle path planning requires several low and high-level policies to make decisions over varying temporal and spatial scales. Motion prediction is the task of predicting the movement of pedestrians and other vehicles, to understand how the situation might develop based on the current state of the environment.

9.8 Reinforcement Learning in Manufacturing

Producing products that can fulfill our fundamental requirements and wants is the main goal of manufacturing. With their own RL solutions for packaging and quality testing, Cobot Manufacturers (or Manufacturers of Collaborative Robots that can do diverse production jobs with a workforce of more than 100 people) are assisting many enterprises. Without a doubt, their utilization is accelerating the process of producing high-quality items that can firmly reject unfavorable client feedback. Additionally, the performance of the product and the sales margin are both improved by the absence of negative feedback.

9.9 Reinforcement Learning in Broadcast Journalism

Attracting likes and views and following reader behavior is significantly easier with different forms of reinforcement learning. Besides, recommending news that meets the frequently-changing tastes of readers and other web users might potentially be attained since journalists can now be provided with an RL-based system that maintains an eye on intuitive news content as well as the headlines. Look at the additional benefits that Reinforcement Learning is providing readers all around the world.

9.10 Reinforcement Learning in Marketing

Marketing is the process of advertising and then selling goods and services, whether they are under your own brand or those of another. Finding the correct audience to advertise to in order to generate more returns on the investment you or your business are making is a problem in and of itself. And that is one of the reasons businesses spend money managing numerous marketing efforts digitally. Real-time bidding supports the core competencies of RL, as well as those of smaller and bigger businesses like yours.

9.11 Datacenters Cooling

Today, AI can assist us in addressing some of the most difficult physical issues the world has ever seen, such energy use. Large-scale commercial and industrial systems like data centers have a high energy consumption to keep the servers operating, with the entire globe on the cutting edge of virtualization and cloud-based applications. In comparison to the current PID controllers, this method of using a Reinforcement Learning agent with little to no prior knowledge may effectively and securely govern conditions on a server floor. The properties such as

temperatures, power, set points, etc. in the data acquired by hundreds of sensors within the data centers are provided to be utilized to train the deep neural networks for datacenter cooling. Deep Q-learning Network (DQN) [81] based approaches are frequently employed to overcome this barrier because standard machine learning algorithms find it difficult to directly solve this problem owing to the absence of diverse datasets.

9.12 Traffic Light Control

A popular data-driven method for adaptive traffic signal regulation is reinforcement learning. These models are trained with the intention of developing a policy that, given the present state of the traffic, operates the traffic signal [47] optimally. The decision-making process must be dynamic and dependent on the volume of traffic coming in from various directions [82] at various times of the day. Due to this non-stationary nature, the usual method of controlling traffic appears to have its limitations. Furthermore, the policy cannot be applied to a junction with y lanes after it has been trained for an intersection with x lanes.

9.13 DTRs (Dynamic Treatment Regimes)

DTRs entail making a series of healthcare decisions that are specific to a patient based on their medical history and conditions throughout time, including the kind of therapy, drug doses, and appointment scheduling. The algorithm that generates treatment choices uses this input data to give the patient's surroundings its ideal state. The problematic part is that patients [83] with chronic, long-lasting illnesses like HIV acquire medication resistance, necessitating a gradual drug transition and emphasizing the significance of the treatment order. Physicians may turn to prior studies, scholarly reviews, and analyses when they need to modify a course of treatment for a specific patient [84]. For many ICU situations, the relevant use-case data might not be accessible. Additionally, a large number of patients admitted to ICUs may be too unwell to participate in clinical studies. Other approaches, such as sizable observational data sets, are required to support ICU clinical choices. One machine learning technique called reinforcement learning (RL) is particularly suited for ICU settings because of the dynamic nature of critically sick patients.

9.14 Natural Language Processing (NLP)

Due of the intrinsic decision-making capabilities of Reinforcement Learning, it is used in language interpretation. The agent makes an effort to comprehend the sentence's current situation and formulate an action plan that will maximize the value it will bring. Due to the size of the state space and the action space, [85] the issue is complicated. Numerous NLP applications, including text summarization, question answering, translation, conversation production, machine translation, etc., employ reinforcement learning. Agents trained in reinforcement learning may be taught to comprehend a few phrases from a text and utilize that knowledge to reply to queries.

X. CONCLUSION

One of the most popular study areas in the field of contemporary artificial intelligence (AI) is reinforcement learning (RL), and its popularity is only increasing. The primary issue with supervised and unsupervised learning techniques is how much they rely on previously collected data.

The study of decision-making is called reinforcement learning (RL). It involves understanding how to act in a situation to reap the most benefits. Similar to how toddlers explore their surroundings and discover the behaviors that enable them to accomplish a task, this ideal behavior is learnt via interactions with the environment and observations of how it reacts. The

learner must autonomously determine the order of behaviors that maximizes the reward in the absence of a supervisor. This technique of discovery resembles a trial-and-error search. The quality of an activity is determined by both the current benefit it brings in and any potential future rewards. Reinforcement learning is an extremely potent algorithm because it can learn the behaviors that lead to success in an unobserved environment without the assistance of a supervisor. The researchers can choose which RL technique or algorithms to apply for their problem-solving with the aid of this survey study. Despite the fact that these RL techniques provide substantial advantages for learning from one's own experiences without a precise system model.

XI. REFERENCES

[1]. K. D. Stephan, B Notes for a history of the IEEE society on social implications of technology,[IEEE Technol. Soc. Mag., vol. 25, no. 4, pp. 5–14, 2006

[2]. Yusuf Perwej, "The Bidirectional Long-Short-Term Memory Neural Network based Word Retrieval for Arabic Documents", for published in the Transactions on Machine Learning and Artificial Intelligence (TMLAI), Society for Science and Education, United Kingdom (UK), Volume 3, Issue 1, Pages 16 - 27, 2015, DOI: 10.14738/tmlai.31.863

[3]. R. Sutton and A. Barto, Reinforcement Learning: An Introduction, Cambridge, MA, USA:MIT Press, vol. 2, 2015

[4]. A. Coronato, M. Naeem, G. De Pietro and G. Paragliola, "Reinforcement learning for intelligent healthcare applications: A survey", Artif. Intell. Med., vol. 109, 2020

[5]. Sutton, R.S.; Barto, A.G. Finitie Markov Decision Processes. In Reinforcement Learning: An Introduction, 2nd ed.; The MIT Press: Cambridge, CA, USA, pp. 47–68, 2020

[6]. Virvou, M.; Alepis, E.; Tsihrintzis, G.A.; L.C. Machine Learning Paradigms; Springer: Cham, Switzerland, 2020

[7]. Firoj Parwej, Nikhat Akhtar, Yusuf Perwej, "A Close-Up View About Spark in Big Data Jurisdiction", International Journal of Engineering Research and Application (IJERA), ISSN: 2248-9622, Volume 8, Issue 1, (Part -I1), Pages 26-41, 2018, DOI: 10.9790/9622-0801022641

[8]. Yusuf Perwej, "The Ambient Scrutinize of Scheduling Algorithms in Big Data Territory", International Journal of Advanced Research (IJAR), ISSN 2320-5407, Volume 6, Issue 3, Pages 241-258, 2018, DOI: 10.21474/IJAR01/6672

[9]. Yusuf Perwej, "An Optimal Approach to Edge Detection Using Fuzzy Rule and Sobel Method", International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, Volume 4, Issue 11, Pages 9161-9179, 2015, DOI: 10.15662/IJAREEIE.2015.0411054

[10]. R. S. Sutton and A. G. Barto, Reinforcement learning : an introduction, 2nd ed. Cambridge, MA: Mit Press, 2017

[11]. Yusuf Perwej, "Unsupervised Feature Learning for Text Pattern Analysis with Emotional Data Collection: A Novel System for Big Data Analytics", IEEE International Conference on Advanced computing Technologies & Applications (ICACTA'22), SCOPUS, IEEE No: #54488 ISBN No Xplore: 978-1-6654-9515-8, Coimbatore, India, 4-5 March 2022, DOI: 10.1109/ICACTA54488.2022.9753501

[12]. Y. Perwej, Ashish Chaturvedi, "Machine Recognition of Hand Written Characters using Neural Networks", International Journal of Computer Applications (IJCA), USA, ISSN 0975 – 8887, Volume 14, No. 2, Pages 6- 9, 2011, DOI: 10.5120/1819-2380

[13]. Wiering, M. A., & Van Otterlo, M., "Reinforcement learning," Adaptation, learning, and optimization, Vol.12, No.3, 2012

[14]. Trivedi, A.; Tripathi, C.M.; Perwej, Y.; Srivastava, A.K.; Kulshrestha, N. Face Recognition Based Automated Attendance Management System. Int. J. Sci. Res. Sci. Technol, 9, 261–268, 2022,

[15]. Sutton, R.S.; Barto, A.G. Finitie Markov Decision Processes. In Reinforcement Learning: An Introduction, 2nd ed.; The MIT Press: Cambridge, CA, USA, 2020; pp. 47–68. Available online: http://incompleteideas.net/book/RLbook2020.pdf (accessed on 17 September 2022).

[16]. Virvou, M.; Alepis, E.; Tsihrintzis, G.A.; Jain, L.C. Machine Learning Paradigms; Springer: Cham, Switzerland, 2020. [CrossRef]

[17]. Coursera. 3 Types of Machine Learning You Should Know. 2022. Available online: https://www.coursera.org/articles/types-ofmachine- learning (accessed on 12-Jan-2023).

[18]. T. F. G. Title, D. R. Learning, E. Telem, G. Mu, F. Advisor, and C. B. Mux, "Bachelor degree thesis," 2018

[19]. M. R. F. Mendonca, H. S. Bernardino, and R. F. Neto, "Simulating human behavior in fighting games using reinforcement learning and artificial neural networks," in Proceedings of the 2015 14th Brazilian Symposium on Computer Games and Digital Enter- tainment (SBGames), vol. 0, pp. 152–159, Piau´ı, Brazil, November 2015

[20]. V. Mnih et al., "Playing Atari with Deep Reinforcement Learning," in Conference on Neural Information Processing Systems, 2013, pp. 1–9.

[21]. Shubham Mishra, Mrs Versha Verma, Nikhat Akhtar, Shivam Chaturvedi, Yusuf Perwej, "An Intelligent Motion Detection Using OpenCV" , International Journal of Scientific Research in Science, Engineering and Technology (IJSRSET), Print ISSN: 2395-1990 , Online ISSN : 2394-4099, Volume 9, Issue 2, Pages 51-63, 2022, DOI: 10.32628/IJSRSET22925

[22]. Panzer, M.; Bender, B. Deep reinforcement learning in production systems: A systematic literature review. Int. J. Prod. Res. 2021, 60, 4316–4341. [CrossRef]

[23]. DeepMind, "About Us." [Online]. Available: https://deepmind.com/about/. [Accessed: 14-Jan-2023]

[24]. G. Tesauro, "TD-Gammon, a Self-Teaching Backgammon Program, Achieves Master-Level Play," Neural Comput., vol. 6, no. 2, pp. 215–219, 1994

[25]. D. Silver et al., "Mastering the game of Go with deep neural networks and tree search," Nature, vol. 529, no. 7587, pp. 484–489, 2016

[26]. C.-S. Lee, M.-H. Wang, S.-J. Yen et al., "Human vs. Computer go: review and prospect [discussion forum]," IEEE Computational Intelligence Magazine, vol. 11, no. 3, pp. 67–72, 2016

[27]. S. Carta, A. Corriga, A. Ferreira, D. R. Recupero, and A. S. Podda, "A multi-layer and multi-ensemble stock trader using deep learning and deep reinforcement learning," 2020

[28]. A. Perwej, K.P. Yadav, V. Sood and Y. Perwej, "An Evolutionary Approach to Bombay Stock Exchange Prediction with Deep Learning Technique", IOSR Journal on Business Management, Vol. 20, No. 12, pp. 63-79, 2018

[29]. Asif Perwej, Dr. Yusuf Perwej, Nikhat Akhtar, and Firoj Parwej, "A FLANN and RBF with PSO Viewpoint to Identify a Model for Competent Forecasting Bombay Stock Exchange", COMPUSOFT, SCOPUS, International Journal of Advanced Computer Technology, 4 (1), Volume- IV, Issue-I, Pages 1454-1461, 2015, DOI:10.6084/ijact.v4i1.60

[30]. D. Silver et al., "Mastering Chess Shogi by Self-Play with a General Reinforcement Learning Algorithm," London, 2017

[31]. Bellman, R.E. A Markovian Decision Process. J. Math. Mech. 1957, 6, 679–684. [CrossRef]

[32]. van Otterlo, M.; Wiering, M. Reinforcement learning and markov decision processes. In Reinforcement Learning; Springer: Berlin, Germany, 2012; Volume 12. [CrossRef]

[33]. Nikhat Akhtar, Dr. Devendera Agarwal," A Perceptual Evaluation of Optimization Algorithms and Iterative Method for E-Commerce", International Journal of Science and Research (IJSR), ISSN (Online): 2319-7064, Volume 3 Issue 12, Pages 2527 – 2534, 2014

[34]. P. Abbeel, A. Coates, M. Quigley, and A. Y. Ng, "An application of reinforcement learning to aerobatic helicopter flight," Education, vol. 19, p. 1, 2007.

[35]. A. Y. Ng et al., "Autonomous inverted helicopter flight via reinforcement learning," Springer Tracts Adv. Robot., vol. 21, pp. 363–372, 2006.

[36]. S. Carta, A. Ferreira, A. S. Podda, D. Reforgiato Recupero, and A. Sanna, "Multi-DQN: an ensemble of deep Q-learning agents for stock market forecasting," Expert Systems with Applications, vol. 164, Article ID 113820, 2021.

[37]. Luketina, J., Nardelli, N., Farquhar, G., Foerster, J., Andreas, J., Grefenstette, E., & Rocktäschel, T., "A survey of reinforcement learning informed by natural language," arXiv preprint, arXiv:1906.03926, 2019.

[38]. Shobhit Kumar Ravi, Shivam Chaturvedi, Dr. Neeta Rastogi, Dr. Nikhat Akhtar, Dr. Yusuf Perwej, "A Framework for Voting Behavior Prediction Using Spatial Data", International Journal of Innovative Research in Computer Science & Technology (IJIRCST), ISSN: 2347-5552, Volume 10, Issue 2, Pages 19-28, 2022, DOI: 10.55524/ijircst.2022.10.2.4

[39]. Yusuf Perwej, "Recurrent Neural Network Method in Arabic Words Recognition System", International Journal of Computer Science and Telecommunications (IJCST), UK, London Volume 3, Issue 11, Pages 43-48, 2012

[40]. Ahmed, S.H., and Koob, G.F., "Transition to Drug Addiction: A Negative Reinforcement Model Based on an Allostatic Decrease in Reward Function", Psychopharmacology, 180(3), pp. 473-490, 2005

[41]. Sutton S, Barto G. Reinforcement Learning: An Introduction [M], Cambridge, MA, USA: MIT Press, 1998

[42]. q=https://drive.google.com/drive/folders/1lZFy9KDnW8Ru2OD6wFgP1GXpZHf7QAy&sa=D&source=docs&ust=1643043550121570&usg=AOvVaw2ZDr6mtmPCmH6I1W4czIBj

[43]. https://neptune.ai/blog/model-based-and-model-free-reinforcement-learning-pytennis-case-study

[44]. Beom H B. A sensor2based navigation for a mobile robot using fuzzy logic and reinforcement learning [J]. IEEE Trans. on Systems, Man, and Cybernetics, 25(3):464-477, 1995

[45]. M. van Otterlo and M. Wiering, Reinforcement Learning and Markov Decision Processes, Berlin, Heidelberg:Springer Berlin Heidelberg, pp. 3-42, 2012

[46]. S. Coskun and R. Langari, "Predictive Fuzzy Markov Decision Strategy for Autonomous Driving in Highways", 2018 IEEE Conf. Control Technol. Appl. CCTA 2018, pp. 1032-1039, 2018

[47]. Mumdouh Mirghani Mohamed Hassan,Yusuf Perwej, Awad Haj Ali Ahmed, Firoj Parwej," Using Intelligent Transportation Systems the Modern Traffic Safety on the Highway in the Sudan" , International Journal of Computer Science Trends and Technology (IJCST), ISSN 2347 – 8578, Volume 7, Issue 3, Pages 1- 13, May - Jun 2019, DOI: 10.33144/23478578/IJCST-V7I3P1

[48]. J. Hu and M. P.Wellman, "Nash Q-learning for general-sum stochastic games," J. Mach. Learn. Res., vol. 4, pp. 1039–1069, 2003

[49]. N. Suematsu and A. Hayashi, "A multiagent reinforcement learning algorithm using extended optimal response," in Proc. 1st Int. Joint Conf. Auton. Agents & Multi agent Syst., Bologna, Italy, July 15-19 2002, pp. 370–377

[50]. M. Bowling, "Multiagent learning in the presence of agents with limitations," Ph.D. dissertation, School of Computer Science, Carnegie Mellon University, Pittsburgh, May 2003

[51]. M. H. Bowling and M. M. Veloso, "Multiagent learning using a variable learning rate," Art. Intell., vol. 136, no. 2, pp. 215–250, 2002

[52]. M. L. Littman, "Friend-or-foe Q-learning in general-sum games," in Proc. 18th Int. Conf. Machine Learning, Morgan Kaufman, pp. 322–328, 2001

[53]. M. L. Littman, "Markov games as a framework for multi-agent learning," in Proc. 11th Int. Conf. Machine Learning, San Francisco, pp. 157–163, 1994

[54]. M. L. Littman and C. Szepesv´ari, "A generalized reinforcement-learning model: convergence and applications," in Proc. 13th Int. Conf. Machine Learning, Bari, Italy, pp. 310–318, 1996

[55]. M. L. Littman, "Value-function reinforcement learning in markov games," J. Cogn. Syst. Res., vol. 2, pp. 55–66, 2001

[56]. E. F. Morales, "Scaling up reinforcement learning with a relational representation," in Workshop Adaptabil. Multi-Agent Syst., Sydney, 2003

[57]. George W. Brown. Some notes on computation of Games Solutions. RAND Corporation Report P-78, April 1949.

[58]. Julia Robinson. An iterative method of solving a game. The Annals of Mathematics, 54(2):296 – 301, 1951

[59]. Yuri V. Matiyasevich. Hilbert's Tenth Problem. MIT Press, Cambridge, Massachusetts, 1993.

[60]. K. Miyasawa. On the convergence of the learning process in a 2 x 2 nonzero sum two-person game. Technical report, Research memo 33, Princeton University, 1961.

[61]. D. Monderer and L. S. Shapley. Fictitious play property for games with identical interests. Journal of Economic Theory, 68:258–265, 1996.

[62]. J. Nachbar. Evolutionary selection dynamics in games: Convergence and limit properties. International Journal of Game Theory, 19:59–89, 1990

[63]. J. Robinson. An iterative method of solving a game. Annals of Mathematics, 54:296–301, 1951

[64]. B. Banerjee and J. Peng. Rvσ(t): A unifying approach to performance and convergence in online multiagent learning. In Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS), pages 798–800, Hakodate, Japan, 2006

[65]. M. Bowling. Convergence and no-regret in multiagent learning. In Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS), pages 209–216, Vancouver, Canada, 2005

[66]. M. Bowling and M. Veloso. Multiagent learning using a variable learning rate. Artificial Intelligence, 136:215–250, 2002

[67]. D. G. Lainiotis, "Sequential Structure and Parameter Adaptive Pattern Recognition Part I: Supervised Learning", IEEE Transactions on Information Theory, vol. IT-16, no. 5, pp. 548-556, September 1970

[68]. Yusuf Perwej, Nikhat Akhtar, Firoj Parwej, "The Kingdom of Saudi Arabia Vehicle License Plate Recognition using Learning Vector Quantization Artificial Neural Network", International Journal of Computer Applications (IJCA), USA, Volume 98, No.11, Pages 32 – 38, 2014, DOI: 10.5120/17230-7556

[69]. Poranki KR, Perwej Y, Perwej A. ," The Level of Customer Satisfaction related to GSM in India", RJSITM., 4(3):32-3, 2015

[70]. Perwej A, Yadav KP, Sood V, Perwej Y,"An evolutionary approach to bombay stock exchange prediction with deep learning technique", IOSR J Bus Manag (IOSR-JBM) 20(12):63–79, 2018

[71]. Yusuf Perwej , Asif Perwej , "Forecasting of Indian Rupee (INR) / US Dollar (USD) Currency Exchange Rate Using Artificial Neural Network", International Journal of Computer Science, Engineering and Applications (IJCSEA), Academy & Industry Research Collaboration Center (AIRCC), USA , Volume 2, No. 2, Pages 41- 52, April 2012, DOI: 10.5121/ijcsea.2012.2204

[72]. L. Busoniu, R. Babuska, B. De Schutter, "A comprehensive survey of multiagent reinforcement learning", IEEE Trans. Syst. Man Cybern., vol. 38, no. 2, pp. 156-172, 2008

[73]. Saurabh Sahu, Km Divya, Dr. Neeta Rastogi, Puneet Kumar Yadav, Dr. Yusuf Perwej, "Sentimental Analysis on Web Scraping Using Machine Learning Method" , Journal of Information and Computational Science (JOICS), ISSN: 1548-7741, Volume 12, Issue 8, Pages 24-29, August 2022, DOI: 10.12733/JICS.2022/V12I08.535569.67004

[74]. Nikhat Akhtar, Devendera Agarwal, "An Efficient Mining for Recommendation System for Academics", International Journal of Recent Technology and Engineering (IJRTE), SCOPUS, Volume-8, Issue-5, Pages 1619-1626, 2020 , DOI: 10.35940/ijrte.E5924.018520

[75]. Yusuf Perwej, "The Ambient Scrutinize of Scheduling Algorithms in Big Data Territory",International Journal of Advanced Research (IJAR), ISSN 2320-5407, Volume 6, Issue 3, Pages 241-258, 2018, DOI: 10.21474/IJAR01/6672

[76]. D. Liu and C. Yang, "A Deep Reinforcement Learning Approach to Proactive Content Pushing and Recommendation for Mobile Users", IEEE Access, vol. 7, pp. 83120-83136, 2019

[77]. Nikhat Akhtar, Devendera Agarwal, "An Influential Recommendation System Usage for General Users", for published in the Communications on Applied Electronics (CAE), ISSN: 2394-4714, Foundation of Computer Science, New York, USA, Vol. 5, No.7, Pages 5 – 9, July 2016, DOI: 10.5120/cae2016652315

[78]. Aboudonia, A., Scianca, N., de Simone, D., Lanari, L., & Oriolo, G. (2017). Humanoid gait generation for walk-to locomotion using single-stage MPC. In 17th IEEE-RAS Int. Conf. on Humanoid Robots, pp. 178–183

[79]. J. Hill, W. R. Ford and I. G. Farreras, "Real conversations with artificial intelligence: A comparison between human-human online conversations and human-chatbot conversations", Computers in Human Behavior, vol. 49, pp. 245-250, 2015

[80]. D. Silver, A Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, et al., "Mastering the game of Go with deep neural networks and tree search", Nature, vol. 529, no. 7587, pp. 484-489, 2016

[81]. C. JCH. Watkins and P. Dayan, "Q-Learning", Mach. Learn, vol. 8, no. 3–4, pp. 279-292, May 1992

[82]. Ankit Kumar, Neha kulshrestha, Yusuf Perwej, Ashish Kumar Srivastava, Chandan Mani Tripathi, "The Assay of Potholes and Road Damage Detection", International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), ISSN : 2456-3307, Volume 8, Issue 1, Pages 202-211, January-February-2022, DOI: 10.32628/CSEIT228135

[83]. Nikhat Akhtar, Saima Rahman, Halima Sadia, Yusuf Perwej, "A Holistic Analysis of Medical

Internet of Things (MIoT)", Journal of Information and Computational Science (JOICS), ISSN: 1548 - 7741, Volume 11, Issue 4, Pages 209 - 222, 2021, DOI: 10.12733/JICS.2021/V11I3.535569.31023

[84]. Yusuf Perwej, Nikhat Akhtar, Neha kulshrestha, Pavan Mishra, "A Methodical Analysis of Medical Internet of Things (MIoT) Security and Privacy in Current and Future Trends", Journal of Emerging Technologies and Innovative Research (JETIR), Volume 09, Issue 1, Pages 346 - 371, 2022, DOI: 10.6084/m9.figshare.JETIR2201346

[85]. Y. Wang, "Natural language processing and applications in machine learning", Modern Chinese, vol. 5, pp. 187-191, 2019

Cite this article as :

Shweta Pandey, Rohit Agarwal, Sachin Bhardwaj, Sanjay Kumar Singh, Dr. Yusuf Perwej, Niraj Kumar Singh, "A Review of Current Perspective and Propensity in Reinforcement Learning (RL) in an Orderly Manner", International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), ISSN : 2456-3307, Volume 9, Issue 1, pp.206-227, January-February-2023. Available at doi : https://doi.org/10.32628/CSEIT2390147
Journal URL : https://ijsrcseit.com/CSEIT2390147