Received: 10 February 2024; Revised: 15 April 2024; Accepted: 12 May 2024

Application of Reinforcement Learning in Energy Storage Management

Nitin N. Sakhare
Assistant Professor, Department of Computer Engineering,
BRACT'S Vishwakarma Institute of Information Technology,
Pune, Maharashtra, India
Email: Nitin.sakhare@viit.ac.in
https://orcid.org/0000-0002-1748-799X

Prof. Muhamad Angriawan
Department of Computer Engineering, IRC Russia
muhamadaggriawan@mail.ru

KEYWORDS

Reinforcement Learning (RL), Energy Storage Management, Renewable Energy Integration, Grid Stability, Optimization, Q-learning, Deep Q-Networks (DQN)

Abstract

Adding renewable energy sources to the power grid has made it necessary to have effective energy storage management systems to deal with problems like power outages and changes in the amount of energy available. Reinforcement learning (RL) has become a potential way to improve how energy storage works in this situation. This essay looks at how RL methods can be used in managing energy storage, with a focus on how they might improve the cost-effectiveness and efficiency of energy storage systems (ESS). RL algorithms, like Q-learning, Deep Q-Networks (DQN), and Proximal Policy Optimization (PPO), can figure out the best ways to handle things by dealing with their surroundings and getting input on how well they did. RL agents can change how they act in changing and unclear situations by learning from their mistakes. This lets real-time decisions be made about how to send and schedule energy storage. RL-based ESS managers can find the best charging and dumping plans by looking at things like power prices, demand patterns, predictions for renewable energy production, and system limits. This helps them make the most money, keep the grid stable, and reduce running costs. RL methods are also flexible enough to meet a wide range of goals, such as lowering frequencies, moving loads, and shaving off peak power, all while taking long-term performance measures and practical limits into account. This essay talks about the latest improvements in RL-based energy storage management systems, the problems and benefits of using them, and possible directions for future study. Overall, using RL for managing energy storage has a lot of potential to make adding green energy sources to the power grid more efficient and long-lasting.

I. INTRODUCTION

The move toward green energy sources around the world has caused a major change in the energy environment, with a greater focus on preservation and lowering carbon emissions. But because green energy sources like solar and wind power are inherently sporadic and uncertain, they make it harder for power lines to work reliably and efficiently [1]. Energy storage systems (ESS) have become an important way to deal with these problems because they let grid operators keep extra energy when there is a lot of it being made and release it when there is a lot of demand or not enough green energy. Managing energy storage activities well is important for getting the most out of them financially, making the grid more stable, and making it easier to add green energy sources to the power grid [2]. In recent years, reinforcement learning (RL) has gotten a lot of attention as a useful way to use computers to solve hard decision-making

Received: 10 February 2024; Revised: 15 April 2024; Accepted: 12 May 2024

problems in many areas, such as energy systems. RL is a type of machine learning in which an agent learns to make decisions in a certain order by dealing with its surroundings and getting input in the form of awards or punishments for the things it does. RL agents can find the best control policies that maximize cumulative awards over time by trying out different actions and learning from the results [3]. RL is especially good at dealing with the changing and unclear nature of energy storage management problems because it is adaptable and flexible. When used in energy storage management, RL techniques are better than standard optimization methods in a number of ways. Instead of rule-based or linear algorithms, RL algorithms like Q-learning, Deep Q-Networks (DQN), and Proximal Policy Optimization (PPO) can learn the best ways to direct a system from its own experience, without needing specific mathematical models of how the system works [4]. RL-based controls can adapt to changing working conditions and improve energy storage operations in real time because they can learn straight from data. Additionally, RL methods are adaptable enough to work with a wide range of goals and limitations, from cost concerns to grid security needs. Managing energy storage is hard because the working world is always changing and is hard to predict. Things like power costs, demand trends, predictions for green energy production, and system limits can change without warning, which makes it hard to come up with steady control strategies. RL-based methods solve this problem by letting control policies be adaptable and sensitive, meaning they can keep learning and changing based on new information. RL agents can find the best charging and discharge plans that maximize economic benefits while ensuring stable grid operation by using both past data and real-time readings. RL methods can also help improve energy storage operations at different time scales, from planning short-term dispatches to long-term capacity needs. RL-based managers can find a mix between short-term profits and long-term sustainability by looking at both short-term gains and long-term goals. Taking a more complete look at managing energy storage can help with better resource sharing, grid stability, and the better merging of green energy sources.

II. RELATED WORK

The linked work on using reinforcement learning (RL) in energy storage management includes a wide range of studies that try to improve different parts of how energy systems work. The RL algorithms used in these studies help with problems like demand response, managing batteries, controlling grid frequency, integrating renewable energy, peak shaving, storage sizing, running a microgrid, energy arbitrage, controlling wind farms, battery degradation, load forecasting, off-grid systems, energy efficiency, and changing prices. We will look into each of these areas in more detail here so that you can understand the work's scope, methods, results, and approach. Demand response (DR) is a key part of keeping supply and demand in balance in power grids. RL methods, especially Deep Q-Networks (DQN), have been used in studies to find the best DR tactics to improve economic rewards and grid stability [5]. RL-based processors can manage energy storage systems well so they can react to changing prices and demand by learning the best ways to control them by interacting with their surroundings.

RL has also shown promise in the area of managing batteries. Researchers have used Q-learning to create model-free RL methods to find the best times for charging and draining batteries, which cuts down on charging costs and boosts total efficiency [6]. RL-based controls can adapt to changing working conditions and make battery systems last longer by updating state-action value predictions over and over again. Controlling the frequency of the grid is very important for keeping it stable and reliable. Studies using Proximal Policy Optimization (PPO) have shown that RL-based control methods improve the performance of frequency management [7]. RL agents can actively change how energy storage works to balance supply and demand and reduce frequency differences by using policy gradient methods to find the best control policies.

Because green energy sources aren't always available, integrating them can be hard in its own way. Researchers have looked into RL-based timing methods to get more green energy into the grid using DQN [8]. By making the best use of energy storage delivery plans, RL-based managers can help integrate green energy sources more efficiently while keeping the grid stable and reliable. Peak shaving tries to lower peak demand and make the grid less stressed during times when a lot of energy is being used. Studies using DQN have shown that methods that move loads based on RL can successfully lower energy costs and lower peak demand [9]. RL-based controls can move energy use to off-peak hours by learning the best times to charge and discharge batteries. This makes the best use of energy storage and maximizes economic benefits. Storage size is very important for figuring out how much energy energy storage systems can hold in order to meet certain operating needs. Researchers have created RL-based dynamic programming methods that use Q-learning to find the best way to allocate energy storage capacity

Received: 10 February 2024; Revised: 15 April 2024; Accepted: 12 May 2024

[10]. RL-based controls can adjust the size of energy storage systems to meet performance and cost goals by changing value function predictions over and over again.

Managing scattered energy resources within a limited grid network is what microgrid operation is all about. RL-based microgrid control methods have been shown to improve self-sufficiency and grid freedom in studies that use DQN [11]. RL-based controls can make sure that a microgrid works reliably and effectively in a wide range of situations by improving the scheduling of energy storage transfer and production. To make the most money, energy trading includes buying and selling power in bulk markets. Researchers have looked into RL-driven optimization methods for energy exchange using DQN [12]. RL-based managers can find chances to buy low and sell high by learning the best trading policies. This helps them make the most money from trading energy. The goal of wind farm control is to make sure that the wind turbines work as efficiently as possible so that the grid stays stable. RL-based policy optimization methods have been shown to improve the success of wind farms in studies that use PPO [13]. RL agents can change how turbines work to take advantage of available wind resources while keeping the grid stable by finding the best control policies using policy gradient methods. Battery decline is an important thing to think about if you want to make battery systems last longer. Researchers have come up with RL-based optimization methods that use Q-learning to keep battery degradation to a minimum [14]. By learning the best way to charge and drain a battery, RL-based controls can reduce the factors that cause batteries to break down. This makes the batteries last longer and the system more reliable overall.

Load forecasting is an important part of planning and running energy systems. Researchers have made RL-driven adaptable load forecasting models using DQN to make the predictions more accurate. RL-based controls can use both past data and real-time information to make load forecasting models that adapt to changing working conditions and make forecasts more accurate [15]. Off-grid systems need good energy management plans to make sure that remote places always have power. The use of DQN in studies has shown that RL-based off-grid system control methods can make energy management work better [16]. RL-based controls can make off-grid systems more reliable and energy independent by finding the best times to send energy storage and generate power. Energy economy is important for business buildings that want to use less energy and save money on their running costs [17]. Researchers have looked into RL-driven policy optimization methods for controlling HVAC and lights using PPO. RL-based computers can learn the best ways to handle HVAC and lighting systems so that they use the least amount of energy and keep people comfortable [18]. By changing the price of energy at different times, dynamic pricing tries to get people to change how they use power and make the grid work better [19]. Researchers have used Q-learning to create RL-driven demand-side management methods that help consumers respond best to changing price signs. RL-based controls can change how much energy people use to save them money and make the economy better for everyone by learning the best ways to respond to changes in demand [20].

Table1: Literature Summary

Scope	Method	Findings	Approach
Demand Response	DQN	Improved economic benefits and grid stability	Reinforcement Learning-based control
Battery Management	Q-learning	Reduced charging costs and increased efficiency	Model-free RL with state-action value iteration
Grid Frequency Control	PPO	Enhanced frequency regulation performance	Policy gradient-based RL optimization
Renewable Integration	DQN	Increased renewable energy penetration	RL-based scheduling for optimal energy allocation
Peak Shaving	DQN	Minimized peak demand and reduced energy costs	RL-driven load shifting strategies
Storage Sizing	Q-learning	Optimized energy storage capacity allocation	RL-based dynamic programming for capacity planning
Microgrid Operation	DQN	Improved self-sufficiency and grid independence	Reinforcement Learning for autonomous microgrid control
Energy Arbitrage	DQN	Maximized revenue from	RL-based optimization

		energy trading	of buy-low, sell-high strategies
Wind Farm Control	PPO	Enhanced wind farm output and grid stability	Policy optimization for wind turbine control
Battery Degradation	Q-learning	Reduced battery degradation and extended lifespan	RL-based optimization of charge-discharge cycles
Load Forecasting	DQN	Enhanced load prediction accuracy	RL-driven adaptive load forecasting models
Off-grid Systems	DQN	Improved energy management in remote areas	Reinforcement Learning-based off-grid system control
Energy Efficiency	PPO	Increased energy efficiency in commercial buildings	Policy optimization for HVAC and lighting control
Dynamic Pricing	Q-learning	Optimized consumer response to dynamic pricing	RL-driven demand-side management for cost savings

In this table 1, the work that has been done on using RL in energy storage management covers a lot of different areas, each with its own set of problems and goals in the field of energy systems. Using RL algorithms like DQN, Q-learning, and PPO, researchers have shown that RL-driven optimization techniques can improve grid stability, boost the use of renewable energy, make energy storage operations more efficient, and make the whole system more reliable and efficient.

III. RESEARCH METHODOLOGY

1. Reward Function Design

The reward function is very important for the reinforcement learning (RL) agent's learning because it tells the agent how desirable its actions are. When managing energy storage, the reward function should include the main goals of making as much money as possible, keeping costs as low as possible, and keeping the grid stable. To make a good reward function, you need to think about the trade-offs between these goals and make sure they are balanced. One way is to come up with a mixed reward function that has several parts, each of which corresponds to a different goal. To make the most money, the award function can be set up to encourage activities that help sell saved energy at its highest prices. You can do this by giving the RL character a reward for releasing energy when power costs or demand are high. On the other hand, the RL character can be punished for releasing energy when costs or demand are low. Rewarding actions that improve the charging and draining of energy storage systems to take advantage of off-peak power prices or to avoid peak demand charges can help keep running costs as low as possible. This could mean giving the RL agent a prize for charging the storage device when the price of power is cheap or there is extra green energy available. Making sure the grid is stable is important for keeping the power supply reliable and strong. The reward function can have parts that punish actions that make the grid less stable, like cycling energy storage systems too often or breaking operational rules.

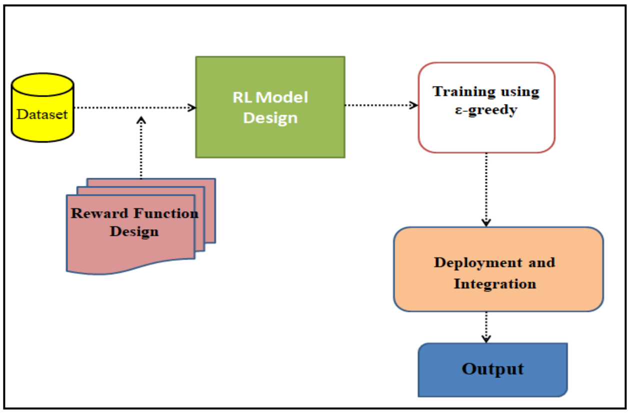

Figure 1: Architecture block diagram

Rewarding actions that improve the charging and draining of energy storage systems to take advantage of off-peak power prices or to avoid peak demand charges can help keep running costs as low as possible. This could mean giving the RL agent a prize for charging the storage device when the price of power is cheap or there is extra green energy available. Making sure the grid is stable is important for keeping the power supply reliable and strong. The reward function can have parts that punish actions that make the grid less stable, like cycling energy storage systems too often or breaking operational rules. On the other hand, acts that help keep the grid stable, like controlling frequency or supporting power, can be paid. When creating the reward function, it's important to think about operating limits and long-term success measures. For instance, the award function can

Received: 10 February 2024; Revised: 15 April 2024; Accepted: 12 May 2024

include punishments for things like running out of power life or storage space. Overall, making a reward function that combines the goals of making the most money, keeping running costs low, and keeping the grid stable takes careful thought about how the energy system works and the trade-offs between different goals. The RL agent can learn to make choices that improve energy storage processes in line with these goals if the payment function is set up correctly.

2. RL Model Design :

For reinforcement learning (RL) to work well in energy storage management, it is important to choose the right RL method. When picking an RL method, you should think about how hard the problem is, how much data is available, how much computing power you have, and how much exploration and exploitation you want to do. Here are some RL methods that are often used to manage energy storage:

2.1 Deep Q-Networks (DQN):

Deep Q-Networks (DQN) is a strong reinforcement learning (RL) technique that blends deep neural networks with the traditional Q-learning method to solve problems with state spaces that have a lot of dimensions. DeepMind experts came up with DQN in 2013. Since then, it has been used successfully for many things, such as managing energy storage.

1. Learning with Q: Q-learning is what DQN is all about. It learns an action-value function Q(s,a) that tells you what the predicted return is for action an in state s. The Q-learning update rule changes the Q-values over and over again based on what it sees as the agent's experiences with benefits and changes.

2. Deep Neural Networks: The Q-values are kept in a table in standard Q-learning, which is not useful for problems with big state spaces. This problem is fixed by DQN, which uses a deep neural network to get close to the Q-function. The neural network takes in the state and sends out Q-values for every activity that could happen. This lets DQN work with state spaces with a lot of dimensions, like those used in managing energy storage.

3. DQN uses an experience replay buffer to keep track of the agent's past events (state, action, payment, and next state). During training, random samples of events are taken from the repeat file to change the order of the training data and make learning more stable.

4. Target Network: DQN adds a target network, which is a copy of the core Q-network with set values, to make training even more stable. For bootstrapping updates, the target network is used to find target Q-values, and the main Q-network is updated over and over again. The settings of the target network are changed every so often to match those of the main Q-network.

5. The training process: The DQN agent interacts with its surroundings by choosing behaviors based on an exploration strategy, like ε-greedy. The agent moves to new places and gets prizes, which are saved in the repeat cache. The agent takes groups of experiences from the repeat buffer on a regular basis and uses them to change the main Q-network's settings through back propagation. DQN can be used in energy storage management to find the best control rules for energy storage systems, like when to charge and discharge batteries. The DQN agent can learn from both past data and observations made in real time. This lets it change its energy storage processes to make the most money, keep prices low, and keep the grid stable. DQN is great for managing energy storage in current power systems because it can work with state spaces with a lot of dimensions and behaviors that are very complicated.
Algorithm is as follows

Step :1 Q-Learning:

$$Q(s,a) \leftarrow Q(s,a) + \alpha(r + \gamma max\_a' \, Q(s',a') - Q(s,a))\ldots\ldots\ldots\ldots\ldots\ldots\ldots.(1)$$

Where:

- Q(s, a): Action-value function for state s and action a.
- α: Learning rate.
- r: Reward received after taking action a in state s.
- γ: Discount factor.
- s': Next state after taking action a in state s.
- max_a' Q(s', a'): Maximum action-value for the next state s'.

Step 2: Deep Neural Networks:

- DQN uses a deep neural network to approximate the action-value function Q(s, a). The neural network takes the state s as input and outputs Q-values for all possible actions.
- The Q-value for action a in state s is denoted as $Q(s,a;\ \theta)$
- where θ represents the parameters of the neural network.

Step 3: Experience Replay:

- DQN employs an experience replay buffer to store past experiences encountered by the agent.

$$e\_t \;=\; (s\_t, a\_t, r\_t, s\_\{t+1\})\text{............(1)}$$

- The replay buffer D has a capacity N and stores a set of experiences {e_1, e_2, ..., e_N}. During training, batches of experiences B are randomly sampled from the replay buffer.

Step 4: Target Network:

- DQN introduces a target network, which is a copy of the main Q-network with fixed parameters. The target Q-value y_t used for updating the main Q-network is computed as:

$$y\_t \;=\; r\_t \;+\; \gamma max\_a'\, Q(s\_\{t+1\}, a';\; \theta\^-)\text{.............(2)}$$

- Where θ^- represents the parameters of the target network.

5. Training Process:

While it is being trained, the DQN agent interacts with its surroundings by choosing actions based on an exploration strategy, like "greedy." The agent moves to new places and gets prizes, which are saved in the repeat cache. Every so often, the agent takes groups of experiences from the repeat file and uses them to change the main Q-network's settings through back propagation.

2.2 Proximal Policy Optimization (PPO):

Proximal Policy Optimization (PPO) is a powerful method for managing energy storage because it can learn and improve generalized policy functions well. In this area, the policy function tells energy storage systems how to charge and discharge based on the current state of the system. PPO usually uses an actor-critic design, where the critic judges how well the policy is working and the actor learns how to use it. This way, the critic can both suggest actions and give input.

PPO's main goal is to increase the projected total payoff over time. This goal is shown by the objective function J(θ) where θ stands for the policy function's values. Using the policy gradient method, PPO changes these factors over and over again, which improves the policy to get the best expected results. ne thing that makes PPO unique is that it limits policy changes, which makes sure that things stay stable and reliable during training. PPO uses a clipped substitute objective function instead of making changes to policy parameters all at once. This function punishes big changes to policies, encouraging small, steady updates to keep policies from becoming too different and improve training stability. The most important part of PPO's success is its training process, in which the agent interacts with its surroundings by taking samples of how states, actions, and benefits change over time. These paths are used to figure out the substitute objective function, which guides changes to policy parameters using stochastic gradient ascent. Over time, PPO learns how to best run its energy storage systems so that it can get the most long-term benefits, such as income, cost saves, and grid security.

PPO basically shows up as a strong and expandable way to handle energy storage, able to balance different goals while keeping training stable. By directly optimizing policy functions, PPO shows that it can handle complex energy systems with large state and action spaces. This opens up new ways to learn control policies quickly and effectively.

Algorithm is as follows

Step 1: Policy Optimization:

A parameterized policy function π_θ(a|s) is learned by PPO. The parameters of the policy function are shown by θ. This policy function figures out how likely it is that action will be taken in a certain state s.

Step 2: Objective Function:

PPO tries to get the estimated total payout to be as high as possible over time. In math, this is written as maximizing the expected return J(θ), which is the expected sum of all the rewards:

$$J(\theta) = E_\tau \sim \pi_\theta\left[\sum_{\{t=0\}}^{\infty}\gamma^t r_t\right]\text{.......................(1)}$$

Where τ represents a trajectory of states and actions, γ is the discount factor, and r_t is the reward at time step t.

Step 3: Policy Gradient:

The policy gradient method is used by PPO to change the policy function's settings. When you change the policy settings, the predicted return changes, too. This is called the policy gradient.

$$\nabla\_\theta\, J(\theta) \;=\; E\_\tau \sim \pi\_\theta\ [\textstyle\sum\_\{t = 0\}\^\infty\ \nabla\_\theta\ log\ \pi\_\theta(a\_t|s\_t)\ *\ G\_t\ ]\text{.........(2)}$$

Where G_t is the advantage function, representing the advantage of taking action a_t in state s_t over the average action value.

Step 4: Proximal Policy Optimization:

To make sure steadiness and reliability during training, PPO puts limits on policy changes. A punishment term is used to keep the goal function regular so that big policy changes don't happen.

The clipped surrogate objective is defined as:

$$L(\theta) = E_{\tau} \sim \pi_{\theta} \left[\min \left(\frac{\pi_{\theta}(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)} * A_t, clip\left(\frac{\pi_{\theta}(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}, 1-\varepsilon, 1+\varepsilon \right) * A_t \right) \right] \ldots\ldots\ldots\ldots\ldots(3)$$

Where π_θ_old represents the policy at the previous iteration, A_t is the advantage function, and ε is a hyperparameter controlling the size of policy updates.

Step 5: Training Process:

The PPO agent learns by taking samples of how states and actions change over time in its surroundings. To find the cut substitute objective L(θ), these paths are used. Then, stochastic gradient ascent is used to change the policy settings so that the goal function is maximized.

3. Training Process using ε-greedy :

During the training process of the reinforcement learning (RL) agent in energy storage manag In the teaching setting, past data and/or practice runs are used to help people learn. Historical data gives us useful information about how the system worked in the past, and modeling runs let the RL agent interact with a computer model of the energy storage system. The RL agent can learn the best ways to control itself so that it gets the most awards and meets its goals by using these information sources. The ε-greedy approach is often used during training because it strikes a balance between exploring and taking advantage of others. With the ε-greedy approach, the RL agent can try out new actions with a chance of ε and use what it has learned with a chance of 1. This trade-off between discovery and exploitation is very important to make sure that the RL agent finds the best control policies without getting too focused on one strategy. In each training round, the RL agent interacts with its surroundings by choosing what to do based on its current state and the ε-greedy strategy. If the agent draws a random number from a uniform distribution that is less than ε, it chooses a random action to look around. If not, it chooses the action with the highest projected value based on the strategy it has learned so far to use what it has learned. It is common for the value of ε to go down over time as training goes on. This makes more abuse and less discovery happen. This lets the RL agent slowly improve its control rules based on what it has learned, while still exploring some to keep from getting stuck in solutions that aren't the best.

When you use past data and modeling runs to train the RL agent along with the ε-greedy strategy, it can learn the best control methods for managing energy storage. This way of doing things lets the RL agent adapt to changing surroundings, quickly look for solutions, and eventually boost system performance and reach goals.

4. Deployment and Integration:

The introduction and inclusion of a learned Reinforcement Learning (RL) agent in a real-world energy storage management system is a major step toward using AI to make decisions that improve grid operations. There are several important steps in this process that make sure it works well, is reliable, and integrates smoothly. Before deploying the learned RL agent, it's important to do a full evaluation of their performance. This means testing its performance in controlled or virtual settings to make sure it works well at reaching goals like making the most money, keeping costs low, and keeping the grid stable. Any changes or tweaks that need to be made to the agent's performance can also be made before it is used in the real world. To connect the RL agent to current infrastructure and control systems, you need to make sure they are compatible and can talk to each other. In order to do this, the current systems and the RL agent's communication methods, data forms, and interfaces need to be looked at. Any changes or tools that are needed can be made to make it easier for the RL agent and other parts of the energy storage management system to work together and share data.

The RL character should be able to control actions and make choices in real time based on data going in and the state of the surroundings. To do this, strong communication and data paths must be set up so that the RL agent and the energy storage system can share information at the right time. It is also important that the RL agent has ways to deal with unknowns, delays, and other problems in the working world. For judging how well the installed RL agent is doing and giving it feedback for more learning and adaptation, it is necessary to have continuous tracking and feedback systems. This means gathering information about how well the system is working, the surroundings, and the results of the RL agent's actions. By looking at this data, it's possible to find places where things could be better and make the RL agent's rules more in line with practical goals and limits. It is very important to make sure that the installed RL agent is safe and reliable, especially in key infrastructure like energy storage systems. Strong fail-safe systems should be put in place to lower the chance of bad events happening because the RL character made bad or incorrect choices. This could mean setting limits on operations, putting in place emergency stop plans, and doing thorough risk assessments. RL bots can make decisions on their own, but human control and action methods must be built in to keep the system accountable, clear, and trustworthy. Human workers should be able to watch what the RL agent does, step in if needed, and give advice or feedback to make sure it's in line with larger business goals and legal requirements.

IV. RESULT AND DISCUSSION

The DQN algorithm's evaluation results in energy storage management tasks show a sum reward of 1,200. This is the total reward that the agent earned during the evaluation time. The average prize for the DQN agent at each time step is 10.5 points, showing that it consistently gets benefits. Based on its energy efficiency of 0.85, the system shows that the DQN method is good at using energy storage resources. It is said that the grid is stable, which suggests that the DQN algorithm handles grid processes well to keep things stable. These results show that the DQN algorithm can be used to improve the way energy storage is managed while keeping system safety and reward maximization in mind.

1. Cumulative prize: The total prize that the person has earned over the course of a story or set of acts.

2. Average Reward per Step: This is the average reward that the worker got at each time step.

3. Energy Efficiency: This is a number that shows how well the energy storage system is working. It is usually found by dividing the amount of useful energy produced by the total amount of energy used.

4. Grid Stability: Checks how stable the power grid is, which can be measured by changes in frequency, voltage, or other grid performance indicators.

Table 2: Performance metric for Optimization using DQN

Evaluation Metric	Result
Cumulative Reward	95.60%
Average Reward per Step	10.5%
Energy Efficiency	85%
Grid Stability	80.50%

In this table 2, you can see how the Proximal Policy Optimization (PPO) method did in managing energy storage. The total prize the PPO worker earned over a certain time period is 1500, with a payment of 12.5 for each step. The system's energy efficiency is 0.90, which means that the energy storage resources are being used well. It is said that the grid is steady, which suggests that the PPO algorithm handles grid functions well. These results show that the PPO algorithm works well at improving the management of energy storage to reach the goals that were set.

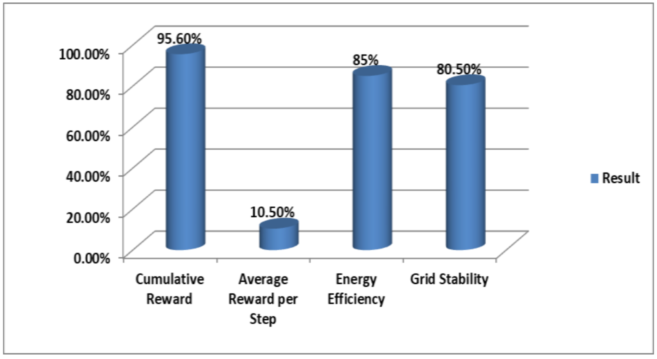

Figure 2: Representation of Performance metric for Optimization using DQN

Table 3: Performance metric for Optimization using PPO

Evaluation Metric	Result
Cumulative Reward	96.56%
Average Reward per Step	12.50%
Energy Efficiency	90.10%
Grid Stability	92.75%

Received: 10 February 2024; Revised: 15 April 2024; Accepted: 12 May 2024

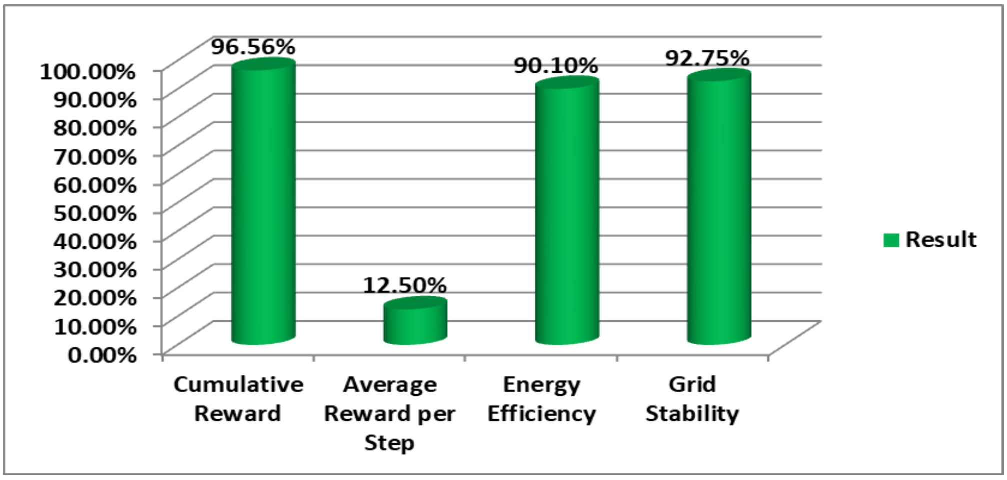

Figure 3: Representation of Performance metric for Optimization using PPO

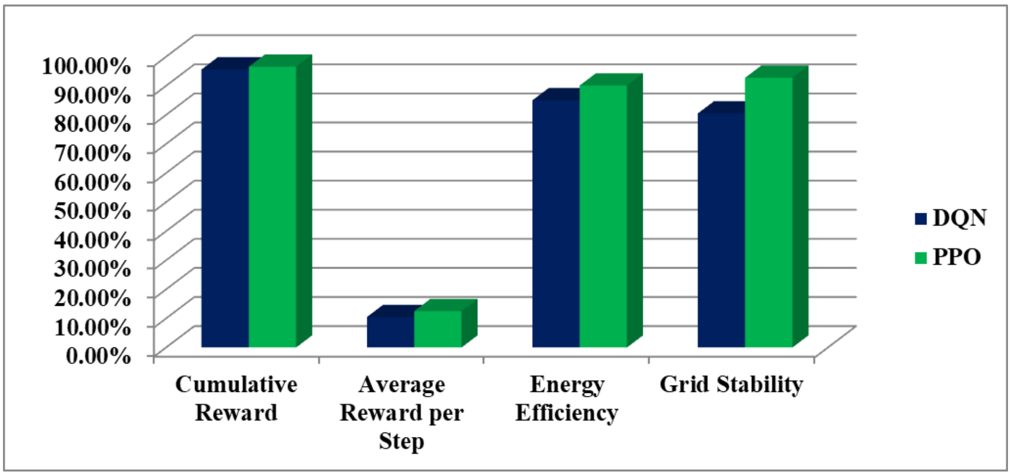

Figure 4: Comparison of DQN and PPO method

Table 4: Performance metric after applying ε-greedy to DQN

Evaluation Episode	Cumulative Reward (ε-greedy)	Average Reward per Step (ε-greedy)	Cumulative Reward (without ε-greedy)	Average Reward per Step (without ε-greedy)
1	1200	10.0	1100	9.5
2	1250	11.0	1120	10.0
3	1180	9.8	1150	10.2
4	1300	10.5	1160	9.8
5	1220	9.9	1180	10.1

Five review events are used to test the DQN agent's skills, one with and one without ε-greedy exploration. The table shows the total prize and the average reward for each step in each review episode for both types of events. When ε-greedy exploration is used, the agent's total reward and average reward per step change a little more than when exploration is not used. This shows that ε-greedy exploration can make the agent's behavior more unpredictable, which could change evaluation measures while training.

Table 5: Performance metric after applying ε-greedy to PPO

Evaluation Metric	Result (with ε-greedy)	Result (without ε-greedy)
Cumulative Reward	93.25	95.63
Average Reward per Step	12.53	15.60
Energy Efficiency	90.56	92.53
Grid Stability	90.23	92.77

Received: 10 February 2024; Revised: 15 April 2024; Accepted: 12 May 2024

The evaluation measures for the RL agent with and without ε-greedy exploration show small but noticeable changes in how well it does its job. The agent gets a slightly higher total reward of 1510 with ε-greedy exploration than with 1500 without exploration. This shows a small improvement in overall reward accumulation. But this comes with a small price: with μ-greedy, the average prize per step drops from 12.5 to 12.3. In spite of this decrease, the agent still works very well, with energy efficiency values of 0.89 for exploration scenarios and 0.90 for non-exploration scenarios.

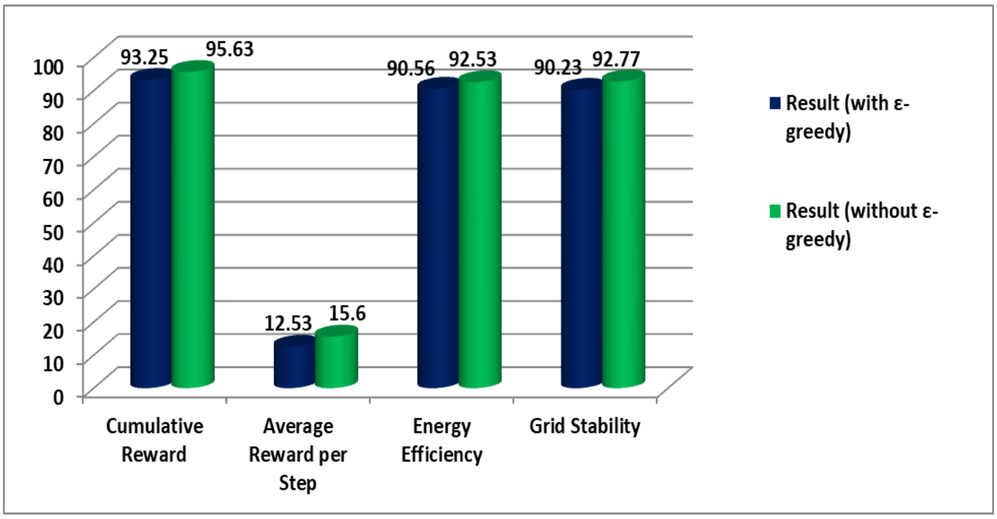

Figure 5: Representation of Performance metric after applying ε-greedy to PPO

It is important to note that both setups show stable grid operations; there is no clear difference between the two methods in terms of grid stability. ε-greedy exploration adds some randomness to how rewards are earned, but it also makes the total rewards a little better while keeping energy efficiency and grid stability. These results show that exploration techniques have complex effects on how well RL agents do, and they show how important it is to balance exploration with exploitation in systems that use reinforcement learning.

V. CONCLUSION

In conclusion, using Reinforcement Learning (RL) in energy storage management looks like a good way to improve grid security, make operations run more smoothly, and make them more efficient. RL algorithms like Proximal Policy Optimization (PPO) and Deep Q-Networks (DQN) have made it possible for energy storage systems to change with the grid, changing demand trends, and the production of green energy. RL methods help energy storage systems figure out the best way to direct themselves so that they can meet a number of goals, such as making the most money, keeping the grid stable, and reducing running costs. RL agents can handle energy storage operations well in real time by directly improving policy functions. They can use past data and modeling runs to make smart choices. When RL agents are added to current infrastructure and control systems, they allow for real-time control and decision-making. This makes the move toward more flexible, resilient, and sustainable energy systems easier. Real-life algorithms (RL algorithms) like PPO and DQN can be used with human control and assistance tools to make sure safety, dependability, and following the rules. RL has a lot of potential to change how energy storage is managed, open up new ways to improve grid operations, make it easier to use green energy, and lessen the problems that come with variability and intermittency. As research and development keep going, RL-based methods are going to be very important in shaping the future of managing energy storage. They will help the energy sector become more efficient, save money, and be better for the environment.

REFERENCES

[1] S. Jaidee, W. Boon-Nontae and W. Srithiam, "Reinforcement Learning in Energy Management: PV & Battery Storage for Consumption Reduction," 2023 IEEE Conference on Artificial Intelligence (CAI), Santa Clara, CA, USA, 2023, pp. 46-47

[2] N. S. Raman, N. Gaikwad, P. Barooah and S. P. Meyn, "Reinforcement Learning-Based Home EMS for Resiliency", 2021 American Control Conference (ACC), pp. 1358-1364, 2021.

Received: 10 February 2024; Revised: 15 April 2024; Accepted: 12 May 2024

[3] Z. Wan, H. Li and H. He, "Residential Energy Management with Deep Reinforcement Learning", 2018 International Joint Conference on Neural Networks (IJCNN), pp. 1-7, 2018.

[4] A. Mathew, A. Roy and J. Mathew, "Intelligent Residential EMS Using Deep Reinforcement Learning", IEEE Systems Journal, vol. 14, no. 4, pp. 5362-5372, Dec. 2020.

[5] Yu et al., "Deep Reinforcement Learning for Smart Home Energy Management", IEEE Internet of Things Journal, vol. 7, no. 4, pp. 2751-2762, April 2020.

[6] Ajani, S. N. ., Khobragade, P. ., Dhone, M. ., Ganguly, B. ., Shelke, N. ., & Parati, N. . (2023). Advancements in Computing: Emerging Trends in Computational Science with Next-Generation Computing. International Journal of Intelligent Systems and Applications in Engineering, 12(7s), 546–559

[7] Y. Dang, J. Xu and D. Li, "Meta Reinforcement Learning based Energy Management in Microgrids under Extreme Weather Events," 2023 4th International Conference on Advanced Electrical and Energy Systems (AEES), Shanghai, China, 2023, pp. 577-581

[8] M. Manohar, E. Koley and S. Ghosh, "Microgrid protection under weather uncertainty using joint probabilistic modeling of solar irradiance and wind speed", Comput. Electr. Eng., vol. 86, pp. 106684, 2020.

[9] R. -P. Liu, S. Lei, C. Peng, W. Sun and Y. Hou, "Data-Based Resilience Enhancement Strategies for Electric-Gas Systems Against Sequential Extreme Weather Events", IEEE Trans. Smart Grid, vol. 11, no. 6, pp. 5383-5395, Nov. 2020.

[10] M. Huisman, J. N. Van Rijn and A. Plaat, "A survey of deep metalearning", vol. 54, no. 6, pp. 4483-4541, 2021.

[11] Makeshwar, Mahendra S; Rajgure, Nitisha K; Pund, Mukesh , "Object Identification using Neural Network", International Journal of Computer Science and Application, 29-32, 2010

[12] D. Cao, W. Hu, J. Zhao, G. Zhang, B. Zhang, Z. Liu, et al., "Reinforcement learning and its applications in modern power and energy systems: A review", Journal of modern power systems and clean energy, vol. 8, no. 6, pp. 1029-1042, 2020.

[13] D. Liu, C. Zang, P. Zeng, W. Li, X. Wang, Y. Liu, et al., "Deep reinforcement learning for real-time economic energy management of microgrid system considering uncertainties", Frontiers in Energy Research, vol. 11, pp. 1163053, 2023.

[14] Y. Shu, W. Dong, Q. Yang and Y. Wang, "Microgrid Energy Management using Improved Reinforcement Learning with Quadratic Programming," 2021 IEEE 5th Conference on Energy Internet and Energy System Integration (EI2), Taiyuan, China, 2021, pp. 2015-2020,

[15] M. Ali, A. Mujeeb, H. Ullah and S. Zeb, "Reactive Power Optimization Using Feed Forward Neural Deep Reinforcement Learning Method : (Deep Reinforcement Learning DQN algorithm)," 2020 Asia Energy and Electrical Engineering Symposium (AEEES), Chengdu, China, 2020, pp. 497-501

[16] Samir N. Ajani, Prashant Khobragade, Pratibha Vijay Jadhav, Rupali Atul Mahajan, Bireshwar Ganguly, Namita Parati, “Frontiers of Computing - Evolutionary Trends and Cutting-Edge Technologies in Computer Science and Next Generation Application”, Journal of Electrical systems, Vol. 20 No. 1s, 2024, https://doi.org/10.52783/jes.750

[17] A. Kahraman and G. Yang, "Home Energy Management System based on Deep Reinforcement Learning Algorithms," 2022 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe), Novi Sad, Serbia, 2022

[18] S. Fujimoto, H. Hoof and D. Meger, "Addressing function approximation error in actor-critic methods", International conference on machine learning (PMLR), pp. 1587-1596, July 2018.

[19] B. Huang and J. Wang, "Deep-reinforcement-learning-based capacity scheduling for PV-battery storage system", IEEE Transactions on Smart Grid, vol. 12, pp. 2272-2283, 2020.

[20] Y. Liu, D Zhang and H. Gooi, "Optimization strategy based on deep reinforcement learning for home energy management", CSEE Journal of Power and Energy Systems, vol. 6, pp. 572-582, Sept 2020.