Deep Reinforcement Learning-Assisted Electric Load Supply Optimization During Disasters

Md. Zahidul Islam
New York University
Email: mi2502@nyu.edu

Panayiotis Christou
New York University
Email: pc2442@nyu.edu

Syed Mohammed Ali Hussaini
New York University
Email: sh6978@nyu.edu

***Abstract*—Extreme events such as natural disasters and cyberattacks cause outages and failures in electric power grids, disrupting consumer access to electricity. In such conditions, strategic decisions, including system reconfiguration, become essential to restore power to critical loads (e.g., hospitals and military bases). Reinforcement Learning (RL) emerges as a crucial tool for supporting sequential decision-making capabilities that can be employed for electric grid reconfiguration in the aftermath of failures. This report utilizes state-of-the-art deep RL algorithms to reconfigure the post-disaster electric grid with the aim of maximizing load supply and minimizing system loss and operational constraint violations. Focusing on a real-world IEEE-standard electric distribution system, we applied RL algorithms such as Proximal Policy Optimization (PPO), Advantage Actor-Critic (A2C), and Trust Region Policy Optimization (TRPO) to facilitate reconfiguration decisions in the system. The simulation results on the IEEE standard 34-bus test system demonstrate the effectiveness of the proposed RL-based system reconfiguration framework in maximizing electric load supply during disasters.**

I. Introduction

High-impact natural disasters such as hurricanes, earthquakes, and floods pose severe challenges to electric power grid resilience. These events can lead to extensive power outages, disrupting essential services and causing significant economic and social impacts. Traditional grid management approaches in disaster scenarios primarily rely on predefined emergency response plans and manual control. While these strategies have been the backbone of disaster response for decades, they are often inadequate in handling the dynamic and complex nature of modern power systems affected by catastrophic events.The limitations of these conventional methods become particularly apparent when dealing with the unpredictability and rapid changes in power demand and supply patterns caused by such disasters [1]. Consequently, there is a pressing need for more advanced, flexible, and efficient solutions to ensure uninterrupted power supply and quick restoration of services post-disaster.

Reinforcement Learning (RL), a branch of machine learning, offers a powerful solution for optimizing electric power grid resilience in the face of natural disasters. RL is a type of machine learning where an agent learns to make decisions by performing actions in an environment to achieve certain objectives. Through trial and error, the agent learns from the outcomes of its actions, continuously improving its strategy to maximize a reward signal. This ability of RL algorithms to dynamically adapt to unpredictable scenarios enables real-time decision-making and optimization of electric load distribution, which is crucial in managing the complexities associated with disaster-induced power grid disruptions. RL, including deep Learning capabilities, demonstrates the potential of solving many critical applications in electric grid management, as listed below:

- Automated load balancing and demand response in real-time.
- Predictive maintenance for preempting component failures.
- Dynamic rerouting of power to maintain critical infrastructure.
- Sustainable integration of renewable energy sources.

This project seeks to address the challenges in electric power grid management by leveraging the capabilities of deep reinforcement learning algorithms. Our approach aims to bring a paradigm shift in disaster-induced grid management, focusing on real-time adaptability, predictive analytics, and automated decision-making to enhance grid resilience and response effectiveness. It differs from traditional methods by employing advanced RL algorithms like Proximal Policy Optimization (PPO) [2], Advantage Actor-Critic (A2C) [3], and Trust Region Policy Optimization (TRPO) [4]. Applied to an IEEE-standard electric distribution system, these algorithms are tested under various disaster scenarios. The aim is to minimize unserved load and system losses while ensuring grid stability, addressing the core issue of rapid and effective restoration of electric services post-disaster [5].

II. Methodology

A. Overview of Reinforcement Learning

Reinforcement Learning (RL) is a branch of machine learning where an agent learns to make decisions by interacting with its environment. The learning process of RL is shown in Fig. 1. The goal is to learn a policy, dictating the agent's actions, to maximize some notion of cumulative reward. In RL, the key components include the agent, the environment, action space, observation space, and reward. The agent takes actions in the environment, which then responds with new states (observations) and rewards. This learning process, often modeled as a Markov Decision Process (MDP), allows the agent to develop a strategy to achieve the best outcome,

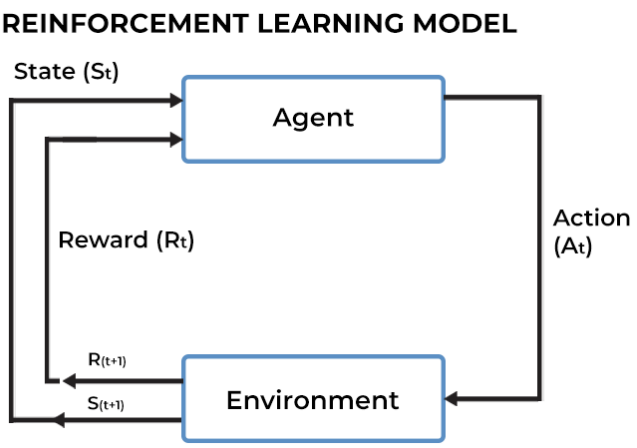

Fig. 1. Illustration of the Reinforcement Learning model [7].

typically defined by the highest cumulative reward over time [6].

1) Environment

The environment in RL represents the world in which the RL agent operates. In our case, the environment is the simulated electric grid system, which includes various components such as power lines, transformers, and loads. The environment provides feedback to the agent in response to its actions. The system also contains some switches along the power lines, which can be controlled to open or close the switches. The power lines become active, when the switches of the lines are closed. An IEEE-standard test system is simulated using OpenDSS, an open source software, which represents the environment in the report [8].

2) Agent

In the context of RL, an agent is the decision-maker or learner. In our project, the agent is the entity responsible for making decisions about electric load distribution during disasters. It interacts with the electric grid (environment), learns from its state, and takes actions to optimize the power supply. Basically, the agent learns to reconfigure the system by closing or opening the tie switches in the electric system, which facilitates load supply during disasters by finding alternative paths between generations and loads.

3) Action Space

The action space defines the set of possible actions that the agent can take. In the electric grid management scenario, actions could include adjusting the power flow, switching on/off certain grid components, or re-routing electricity to different areas. The action space is crucial as it determines the agent's capacity to influence the environment. In the report, the action space is multi-binary representing the status of the switches along the power lines.

4) Observation Space

Observation space refers to the set of all possible states in which the environment can exist. For our electric grid system, this includes the status of different grid components, current load demands, voltage magnitudes of the buses, power flows of the branches, and any other relevant parameters that describe the state of the electric system at any given time. The observation space is obtained by running power flow algorithms in OpenDSS software. The selection of variables in observation space is crucial to ensure stable operation of the electric grid.

5) Reward

The reward in RL is a signal that the agent receives from the environment after taking an action. It indicates the effectiveness of the action. In this project, the reward could be based on criteria such as the minimization of unserved load, efficiency in power distribution, or maintaining grid stability. The agent's objective is to maximize cumulative rewards over time, guiding it towards the most effective strategies for load management during disasters.

B. Choice of Algorithms

For this project, we have selected Proximal Policy Optimization (PPO), Advantage Actor-Critic (A2C), and Trust Region Policy Optimization (TRPO) based on their distinct characteristics and suitability for our problem domain.

1) Proximal Policy Optimization (PPO)

PPO, introduced by Schulman et al., is known for its effectiveness and simplicity. It strikes a balance between ease of implementation and sample efficiency, making it suitable for a wide range of problems, including those with high-dimensional action spaces. PPO's objective is to maintain a balance between exploration (trying new actions) and exploitation (using known information), ensuring stable and consistent learning [9]. Its main function can be expressed as:

$$L^{PPO}(\theta) = \mathbb{E}\left[\min(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)\right] \tag{1}$$

where $r_t(\theta)$ is the probability ratio of the new policy to the old policy, $\hat{A}_t$ is an estimator of the advantage function at time t, and ϵ is a hyperparameter denoting the clipping range.

2) Advantage Actor-Critic (A2C)

A2C is an algorithm that combines two key components: the Actor, which decides the action to take, and the Critic, which evaluates the action. This separation helps in stabilizing the training process. A2C is known for its effectiveness in continuous action spaces, making it a viable option for complex environments like electric grid management during disasters [10]. A2C updates both policy (actor) and value (critic) networks. The loss function for A2C can be described as:

$$L^{A2C} = \mathbb{E}\left[-\log \pi(a_t|s_t)A(s_t, a_t) + \lambda(V(s_t) - V_t^{target})^2\right] \tag{2}$$

where $\pi(a_t|s_t)$ is the policy's probability of taking action a_t in state s_t, $A(s_t, a_t)$ is the advantage at state s_t and action a_t, $V(s_t)$ is the value estimate, V_t^{target} is the target value, and λ is a coefficient balancing the two terms.

3) Trust Region Policy Optimization (TRPO)

TRPO is designed to take the largest possible improvement step on a policy while ensuring the new policy is not too far from the old policy. This approach ensures robustness and reliability, especially in environments with high-dimensional action spaces and complex dynamics, which are typical in electric grid systems [11]. TRPO ensures policy updates do not diverge significantly using a trust region constraint. Its optimization approach can be formulated as:

$$\max_{\theta} \mathbb{E}\left[\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)} A^{\pi_{\theta_{old}}}(s_t, a_t)\right] \tag{3}$$

subject to:

$$\mathbb{E}\left[D_{KL}(\pi_{\theta_{old}}(\cdot|s_t)||\pi_\theta(\cdot|s_t))\right] \leq \delta \tag{4}$$

where D_{KL} denotes the Kullback-Leibler divergence, and δ is a predefined threshold for the divergence.

III. Approach

A. Grid Simulation Using OpenDSS

To create an accurate and dynamic model of the electric grid, we utilized OpenDSS, a versatile and powerful simulation tool developed by the Electric Power Research Institute (EPRI) [8]. This enabled us to replicate a wide range of grid configurations and conditions, including various component types and network topologies. By simulating different types of natural disasters, such as severe weather events and infrastructure failures, we could assess how these scenarios impact the grid's performance and stability.

B. Neural Network Architecture and Custom Policies

Leveraging the capabilities of Stable Baselines3, we developed custom neural network policies tailored to the unique challenges of electric grid data. These policies were constructed with multiple layers, including convolutional and recurrent layers, to effectively process spatial and temporal aspects of grid data. Activation functions such as ReLU (Rectified Linear Unit) and tanh (hyperbolic tangent) were used to introduce non-linearity, enabling the network to model complex relationships and dependencies within the data.

C. Feature Extraction

In our model, the feature extraction component is responsible for transforming raw observations into a format that is more amenable for the neural network to process. We have utilized two different network architectures for the 'Features Extractor' as shown in Figure 2: a Multilayer Perceptron (MLP) and a Convolutional Neural Network (CNN).

An MLP is a class of feedforward artificial neural network that consists of at least three layers of nodes: an input layer, a hidden layer, and an output layer. MLP utilizes a backpropagation algorithm for training the network and can distinguish data that is not linearly separable.

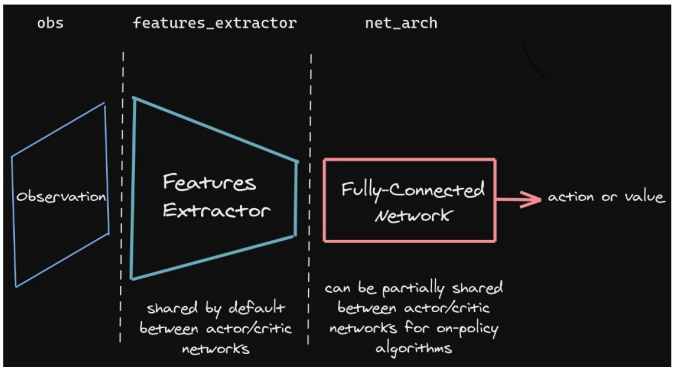

Fig. 2. The architecture of the neural network used in feature extraction. The left part represents the observation input that passes through the feature extractor, which is then inputted into a fully-connected network to produce either an action or a value. Adapted from [12].

A CNN is a deep learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image, and be able to differentiate one from the other. The pre-processing required in a CNN is much lower as compared to other classification algorithms. From the OpenDSS simulations, we extracted a comprehensive set of features representing the state of the electric grid. This included real-time data on load demands, grid component statuses (like switch positions and transformer operations), and power flow metrics across different nodes of the grid. To prepare this data for input into our neural networks, we applied preprocessing steps like normalization to scale the data into a uniform range and encoding techniques to handle categorical variables.

D. Implementation of RL Algorithms

In implementing the PPO, A2C, and TRPO algorithms, we focused on optimizing their configurations to suit our specific use case. This involved adjusting hyperparameters such as learning rates, discount factors, and the size of policy networks to balance learning efficiency and performance. The reward structures were designed to prioritize key objectives such as minimizing unserved load, ensuring supply to critical infrastructure, and maintaining grid stability under stress.

E. Application to Electric Grid Management

Within the OpenDSS environment, our RL models were tasked with managing the grid under various simulated disaster scenarios. The models received real-time state information from the grid and responded with actions aimed at redistributing loads, activating or deactivating grid components, and rerouting power flows. The effectiveness of these actions was continuously evaluated based on their impact on grid stability and load servicing, with the models adapting their strategies over time to improve performance.

IV. Results

A. Algorithm Performance Analysis

The performance of Proximal Policy Optimization (PPO), Advantage Actor-Critic (A2C), and Trust Region Policy Optimization (TRPO) algorithms was assessed under various loading conditions. We focused on their effectiveness in minimizing unserved loads during line faults.

*1) Loading Condition 1 (0.5*Base load)*

Under this lighter load condition, PPO demonstrated a notable edge in performance, with the lowest unserved load.

TABLE I
PERFORMANCE OF RL ALGORITHMS UNDER LOADING CONDITION 1 (0.5*BASE LOAD)

Algorithm	Switch Status	Unserved Load
PPO	[0, 1, 1, 1, 0, 1, 1, 1, 0]	0.1464
A2C	[0, 0, 1, 1, 0, 0, 0, 0, 1]	0.733
TRPO	[1, 0, 1, 1, 1, 1, 1, 0, 1]	0.1483

2) Loading Condition 2 (Base load)

As the load increases to the base level, PPO continues to outperform, but with a reduced margin.

TABLE II
PERFORMANCE OF RL ALGORITHMS UNDER LOADING CONDITION 2 (BASE LOAD)

Algorithm	Switch Status	Unserved Load
PPO	[1, 1, 1, 1, 1, 1, 0, 0, 0]	0.523
A2C	[1, 0, 1, 1, 0, 0, 0, 1, 1]	0.544
TRPO	[0, 0, 1, 0, 1, 1, 1, 1, 1]	0.5553

*3) Loading Condition 3 (1.5*Base load)*

At 1.5 times the base load, the performance dynamics change, with TRPO showing the lowest unserved load.

TABLE III
PERFORMANCE OF RL ALGORITHMS UNDER LOADING CONDITION 3 (1.5*BASE LOAD)

Algorithm	Switch Status	Unserved Load
PPO	[1, 1, 0, 0, 1, 1, 1, 1, 0]	1.745
A2C	[0, 1, 1, 0, 0, 1, 1, 1, 1]	0.738
TRPO	[1, 1, 0, 1, 0, 1, 0, 1, 0]	0.0572

B. Feature Extractor Network Performance

The performance of different reinforcement learning algorithms using a deeper network for feature extraction was evaluated in Table IV. The focus was on how this change impacts the unserved load and loss.

TABLE IV
ALGORITHM PERFORMANCE WITH FEATURE EXTRACTOR DEEPER NETWORK AND CONVOLUTIONAL NEURAL NETWORK (LINE FAULT 24)

Algorithms	Feature Extractor Deeper Network			Convolutional Neural Network		
	Switch Status	Unserved Load	Loss	Switch Status	Unserved Load	Loss
PPO	[1, 1, 0, 1, 0, 1, 0, 1, 1]	1.8349	0.006	[0, 0, 1, 1, 1, 1, 0, 1, 1]	0.5268	0.0009
A2C	[0, 1, 1, 0, 1, 0, 0, 0, 1]	2.2167	0.0011	[1, 1, 0, 1, 1, 1, 1, 1, 1]	0.047	0.0042
TRPO	[0, 1, 0, 1, 1, 1, 1, 1, 1]	0.5177	0.00008	[1, 1, 0, 1, 1, 1, 1, 1, 1]	0.1329	0.0135

In the deeper network setup, TRPO significantly outperformed PPO and A2C in terms of unserved load, achieving the lowest figure at 0.5177. This result indicates its superior effectiveness in complex feature extraction scenarios. Additionally, TRPO demonstrated the lowest loss, with a value of 0.00008, suggesting not only more efficient load handling but also a more stable learning process. These outcomes highlight the critical role of network depth in managing complex scenarios such as line faults in power grid management.

TABLE V
NEURAL NETWORK DEPTH ANALYSIS FOR PPO (LINE FAULT 24)

Architecture	Loss	Topology Violation	Voltage Violation	Branch Flow Violation	Unserved Energy	Convergence Status	Optimal Configuration
Shallow (2 layers)	0.0212	200	0	0	0.567	0	[1, 0, 1, 1, 0, 1, 0, 1]
Medium (5 layers)	0.00281	200	0	0	0.00845	0	[1, 1, 1, 1, 1, 0, 0, 0]
Deep (7 layers)	0.00249	200	0	0	0.5863	0	[1, 0, 1, 1, 0, 1, 1, 0]

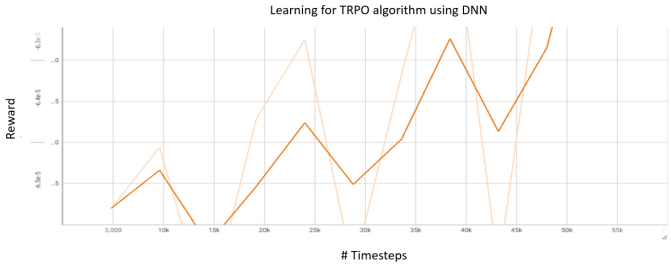

Fig. 3. Learning curve for the TRPO algorithm using a Deep Neural Network (DNN).

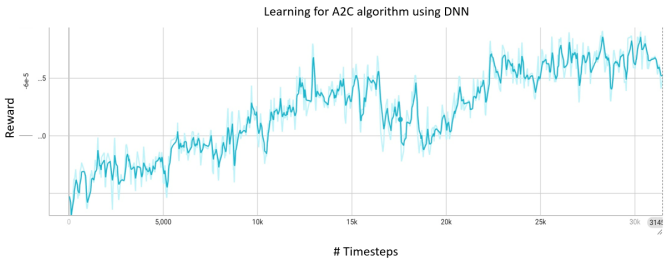

Fig. 4. Learning curve for the A2C algorithm using a Deep Neural Network (DNN).

The analysis indicates a nuanced impact of network depth on the PPO algorithm, as seen in Table V. While the medium-depth (5 layers) network achieved the lowest unserved energy, suggesting an optimal balance between complexity and performance, the deep network (7 layers) did not significantly enhance performance, indicating possible overfitting or unnecessary complexity for the given task. The learning curve (i.e., obtained reward per step) of the TRPO, A2C, and PPO algorithms are also shown in Figures 3, 4, and 5, respectively. Although, PPO provides less effective performance with deep neural network as feature extractor, as indicated in Figure 5, it is capable of improving its performance with medium-depth neural network, as seen in Figure 6.

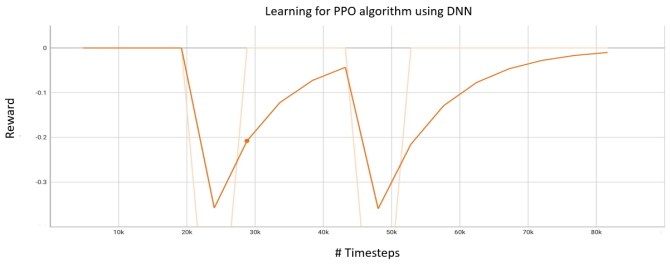

Fig. 5. Learning curve for the PPO algorithm using a Deep Neural Network (DNN) .

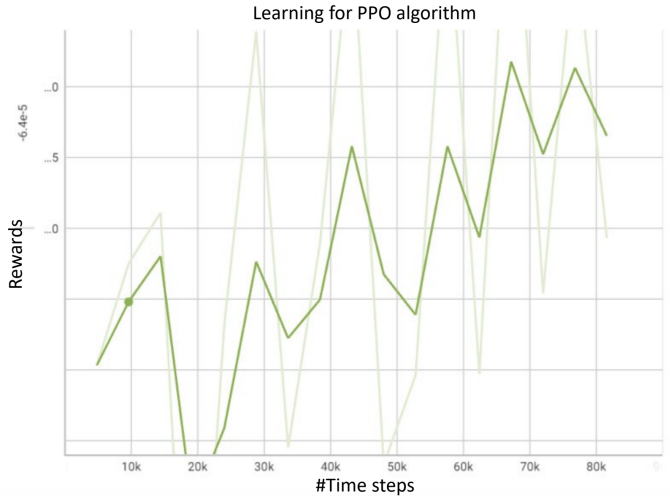

Fig. 6. Learning curve of the PPO algorithm indicating improved performance over time with Medium (5 layers) Neural Network.

C. Comprehensive Fault Analysis

The adaptability of the PPO algorithm was tested under various line fault conditions, each with differing load factors and reported in Table VI. The ability to maintain service with minimal unserved energy and without topology, voltage, or branch flow violations is key to assessing the robustness of the algorithm.

TABLE VI
COMPREHENSIVE FAULT ANALYSIS FOR PPO (ALL 5 LINE FAULTS)

Line	Factor	Loss	Topology Violation	Voltage Violation	Branch Flow Violation	Unserved Energy	Convergence Status	Optimal Configuration
Normal	0.5	9.38E-04	0	0	0	0.5412	0	[1, 1, 1, 0, 1, 0, 1, 0, 0]
Normal	1	2.66E-03	400	0	0	0.5518	0	[1, 1, 0, 1, 0, 1, 1, 0, 1]
Normal	1.5	7.81E-04	800	0	0	0.5861	0	[0, 0, 0, 0, 0, 0, 1, 0, 1]
L7	0.5	6.47E-05	400	0	0	0.7396	0	[1, 1, 0, 0, 0, 0, 0, 1, 1]
L7	1	8.61E-04	200	0	0	0.5268	0	[0, 0, 1, 1, 1, 1, 0, 1, 1]
L7	1.5	1.01E-03	400	0	0	0.567	0	[0, 0, 0, 1, 0, 1, 0, 1, 1]
L9	0.5	7.71E-04	200	0	0	0.5848	0	[0, 0, 1, 0, 1, 1, 1, 0, 1]
L9	1	7.97E-04	600	0	0	0.5452	0	[0, 1, 1, 0, 1, 1, 1, 1, 1]
L9	1.5	2.53E-03	200	0	0	0.057	0	[1, 1, 0, 1, 1, 0, 1, 0, 0]
L15	0.5	6.44E-05	600	0	0	0.7386	0	[1, 1, 0, 0, 1, 0, 0, 0, 0]
L15	1	6.45E-05	400	0	0	0.7386	0	[1, 1, 0, 0, 1, 0, 0, 1, 0]
L15	1.5	7.68E-04	600	0	0	0.588	0	[0, 0, 1, 1, 0, 0, 1, 0, 0]
L16	0.5	6.45E-05	600	0	0	0.7386	0	[1, 1, 0, 0, 1, 0, 0, 0, 0]
L16	1	6.45E-05	400	0	0	0.7386	0	[1, 1, 0, 0, 1, 0, 0, 1, 0]
L16	1.5	7.68E-04	600	0	0	0.588	0	[0, 0, 1, 1, 0, 0, 1, 0, 0]
L18	0.5	7.94E-04	200	0	0	0.5404	0	[1, 1, 0, 0, 1, 1, 1, 1, 0]
L18	1	6.45E-05	800	0	0	0.7414	0	[0, 0, 1, 0, 0, 0, 0, 1, 0]
L18	1.5	3.96E-03	400	0	0	0.118	0	[1, 0, 1, 1, 1, 1, 1, 1, 0]

The PPO algorithm demonstrated varying degrees of effectiveness, with the best performance observed under normal conditions with a 0.5 factor, indicating its potential for effective load management in less severe fault scenarios. However, as the load factor increased, the unserved energy metric also increased, highlighting areas for further optimization. A typical learning curve for initial fault condition in line 7 is shown in Figure 7, which confirms PPO's convergence capability over time.

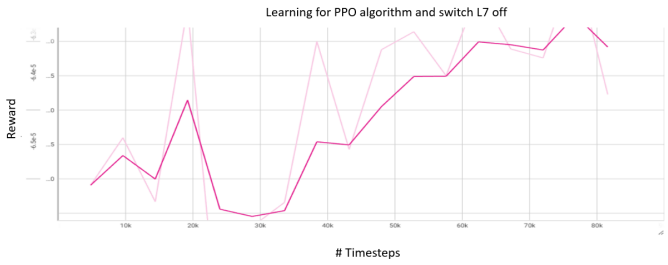

Fig. 7. Learning curve for the PPO algorithm with switch L7 off.

V. CONCLUSION

This study embarked on an exploration of deep reinforcement learning algorithms to enhance the resilience of electric grids during disaster scenarios. By simulating various load conditions and fault instances, we evaluated the efficacy of Proximal Policy Optimization (PPO), Advantage Actor-Critic (A2C), and Trust Region Policy Optimization (TRPO) algorithms. The performance analysis under different load conditions revealed PPO's robustness, especially under lower load scenarios, with TRPO outperforming in scenarios with deeper feature extraction networks. The comprehensive fault analysis further underlined PPO's adaptability, demonstrating effective load management capabilities across a spectrum of fault conditions.

The learning curves for the PPO and A2C algorithms, as visualized in the provided figures, offer insights into the learning dynamics over numerous training steps. PPO's learning trajectory, characterized by its variance in rewards, indicates its explorative learning strategy and potential for optimizing decision-making in real-time power distribution. Conversely, the A2C algorithm showcased a trend towards stability, albeit with a gradual convergence to a lower reward plateau, implying a need for further tuning to enhance its performance under the complex conditions presented by electrical grid disruptions.

The neural network depth analysis revealed a critical insight: a medium-depth network strikes the optimal balance between complexity and performance, providing a guiding principle for designing neural network architectures in reinforcement learning applications for power grid management. Additionally, the use of two different feature extractor networks helps to understand the best-suited deep neural network for the reconfiguration task.

In conclusion, the results affirm the promise of deep reinforcement learning in crafting intelligent, adaptive responses to the intricate challenges posed by emergency grid management. Future work will aim to refine these algorithms, considering a broader array of disaster scenarios, to further improve their reliability and efficiency. By continuing to harness the

capabilities of deep reinforcement learning, we edge closer to realizing a resilient electrical infrastructure capable of withstanding the unforeseen challenges of tomorrow.

VI. Code and Resources

For accessing the source code and additional resources used in this project, please visit the following link: https://github.com/mdzahidul-islam/deep_learning_class_project.git.

VII. Acknowledgments

We would like to extend our sincere gratitude to our professor, Prof. Gustavo Sandoval and the Teaching Assistants, for their invaluable guidance and support throughout the duration of this project. Their insights and expertise have been instrumental in shaping our research and approach.

We also acknowledge the contributions of various researchers whose papers and findings provided us with a foundational understanding and inspiration for our work. The insights gained from these studies have been crucial in guiding our methodology and analysis.

Lastly, we thank all those who have directly or indirectly contributed to our research, offering their time, expertise, and resources to aid in our endeavor.

References

[1] J. Schlegelmilch, "Challenges in Traditional Grid Management during Natural Disasters," Journal of Modern Power Systems and Clean Energy, vol. 11, no. 3, pp. 455-462, 2023.

[2] Zhou, Yuhao, et al. "Deriving AC OPF Solutions via Proximal Policy Optimization for Secure and Economic Grid Operation." arXiv preprint arXiv:2003.12584 (2020).

[3] Wang, Yuan, et al. "Recursive Least Squares Advantage Actor-Critic Algorithms." arXiv preprint arXiv:2201.05918 (2022).

[4] "Trust Region Policy Optimization with Optimal Transport Discrepancies," Advances in Neural Information Processing Systems (NeurIPS), 2023. [Online]. Available: https://papers.nips.cc/paper/2023. [Accessed: 15-Dec-2023].

[5] J. Rahman, R. A. Jacob, S. Paul, S. Chowdhury and J. Zhang, "Reinforcement Learning Enabled Microgrid Network Reconfiguration Under Disruptive Events," 2022 IEEE Kansas Power and Energy Conference (KPEC), Manhattan, KS, USA, 2022, pp. 1-6, doi: 10.1109/KPEC54747.2022.9814797.

[6] R. S. Sutton and A. G. Barto, "Reinforcement Learning: An Introduction," MIT Press, 2018.

[7] "What is Reinforcement Learning?" Spiceworks, 2023. [Online]. Available: https://www.spiceworks.com/tech/artificial-intelligence/articles/what-is-reinforcement-learning/. [Accessed: 16-Dec- 2023].

[8] Electric Power Research Institute, "OpenDSS," [Online]. Available: https://www.epri.com/pages/sa/opendss. [Accessed: 15-Dec-2023].

[9] J. Schulman et al., "Proximal Policy Optimization Algorithms," arXiv:1707.06347, 2017.

[10] V. Mnih et al., "Asynchronous Methods for Deep Reinforcement Learning," in Proceedings of the 33rd International Conference on Machine Learning (ICML), 2016.

[11] J. Schulman et al., "Trust Region Policy Optimization," in Proceedings of the 32nd International Conference on Machine Learning (ICML), 2015.

[12] "Custom Policy Networks." Stable Baselines3 Documentation, 2023. [Online]. Available: https://stable-baselines3.readthedocs.io/en/master/guide/custom_policy.html. [Accessed: 16- Dec- 2023].