energies

MDPI

Article

Deep Reinforcement Learning-Based Approach for Autonomous Power Flow Control Using Only Topology Changes

Ivana Damjanović *, Ivica Pavić, Mate Puljiz and Mario Brcic

Faculty of Electrical Engineering and Computing, University of Zagreb, 10000 Zagreb, Croatia
* Correspondence: ivana.damjanovic@fer.hr

Abstract: With the increasing complexity of power system structures and the increasing penetration of renewable energy, driven primarily by the need for decarbonization, power system operation and control become challenging. Changes are resulting in an enormous increase in system complexity, wherein the number of active control points in the grid is too high to be managed manually and provide an opportunity for the application of artificial intelligence technology in the power system. For power flow control, many studies have focused on using generation redispatching, load shedding, or demand side management flexibilities. This paper presents a novel reinforcement learning (RL)-based approach for the secure operation of power system via autonomous topology changes considering various constraints. The proposed agent learns from scratch to master power flow control purely from data. It can make autonomous topology changes according to current system conditions to support grid operators in making effective preventive control actions. The state-of-the-art RL algorithm—namely, dueling double deep Q-network with prioritized replay—is adopted to train effective agent for achieving the desired performance. The IEEE 14-bus system is selected to demonstrate the effectiveness and promising performance of the proposed agent controlling power network for up to a month with only nine actions affecting substation configuration.

Keywords: power system control; autonomous topology control; artificial intelligence; deep reinforcement learning

check for updates

Citation: Damjanović, I.; Pavić, I.; Puljiz, M.; Brcic, M. Deep Reinforcement Learning-Based Approach for Autonomous Power Flow Control Using Only Topology Changes. *Energies* **2022**, *15*, 6920. https://doi.org/10.3390/en15196920

Academic Editor: Juri Belikov

Received: 26 August 2022
Accepted: 19 September 2022
Published: 21 September 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Electricity is the driving factor of the modern world, and concerns for continued supply and efficient use are becoming more important. Power systems are among the most complex systems ever designed by humans [1], and they require planning and operating in a manner for economic and reliable operation. In recent years, the power system has evolved and faced changes. As the demand for electrical energy continues to increase, the power system is expands and becomes more complex. Low carbon policy and market deregulation have led to significant integration of renewable energy resources with very ambitious targets for the incorporation of more renewable generation in the future. Transmission lines and transformers are often congested and this could lead to network splitting and blackout. Named changes pose challenges to the traditional way of power system control and provide an opportunity for the application of artificial intelligence technology in the power system.

In line with the increasing concerns regarding power systems, quite a number of review works in the literature have presented a detailed description of the problems and notable solutions, including reinforcement learning approaches [2–5]. They survey the application of RL in various fields—such as energy management, demand response, electricity market, operational control, cyber-security, and so on. The most recent work in reviewing reinforcement learning for selective key applications in power systems is presented in [6]. The authors comprehensively analyze the research status of reinforcement learning in the power system field. Studies [2–6] arrive at the conclusion that RL is not

envisioned to completely replace existing model-based methods, but rather it serves as a viable alternative for specific tasks. RL and other data-driven methods are promising when the models are too complex to be useful or when the problems are intrinsically hard to model, such as human-in-loop control. Additionally, this subject is new and still under development and needs many more studies.

Preliminary studies on the application of RL methods in power system control have been presented in [7] where RL was considered in a variety of electric power system control and decision problems for different operating states of the power system. For systems in a normal operating state, [8,9] study the using of RL for optimizing reactive power flows and voltage control. For a disrupted system, [10] proposes a method based on RL for preventing cascading failure and blackout by acting on the generation output. Study [11] considers the possibilities of applying RL for restoration of power systems. A large amount of research focuses on generation dispatch [12,13], voltage control [14,15], or demand response [16,17]. Among them, several studies have exploited a less costly method for congestion management and cascading failure prevention with great potential—namely, grid topology reconfiguration [18–20]. Authors from [21] propose an open-source simulator Grid2Op for power grid management with an RL framework. This simulator offers an opportunity for exploring RL application in power system control. Several authors held a series of competitions on this topic and [18–20] are studies completed as a result of participating in competitions. Study [21] provides a report which confirms aroused interest in the field of application of RL in power system control. Article [18] presents a cross-entropy method, a simple deep RL approach for a 1-week long control of an IEEE 14-bus test system with only busbar splitting. They used a simple feed-forward neural network with two hidden layers, each with 300 neurons, and they reduced action space on 98 unitary actions. The authors in [19] tackled the problem using dueling deep Q-networks (DQN), pretraining, and guided exploration. It took 1200 episodes to train the dueling DQN agent with guided exploration to control the IEEE 14-bus network for 288 timesteps (1 day) continuously. Experiments in [20] were conducted on three power grids—IEEE 5-bus, IEEE 14-bus, and L2RPN WCCI 2020 grid (with 36 substations). They implemented three baselines, considering only bus assignment action to control the power network for 3 days (864 timesteps). One of the observed algorithms was dueling DQN, and the authors report that agent performance is slightly better with their methodology than a do-nothing agent, because it remained in the local optimum.

However, these complicated RL methods are employed with additional learning techniques and can manage the power network for up to several days. Our method simply implements an algorithm from the open-source RL library and does not require additional learning techniques. In this paper, it is demonstrated that by thoroughly selecting action space and reduced observation space, the proposed agent successfully manages the power network without expert help for up to a month. Furthermore, it is a fast-converging method with learning convergence in about 1000 episodes. Accordingly, the remainder of the paper is organized as follows; an overview of reinforcement learning theory is presented in Section 2. The section briefly discusses the deep Q network algorithm and its extension with improved performance used in this research. Section 3 presents a detailed analysis of the IEEE 14-bus test system, datasets, and objective of the research. Section 4 provides an easy-to-understand and simple RL approach. It summarizes all frameworks and tools used in this research and considers thoroughly selected actions based on expert knowledge, reduced observation space, and simple reward. In combination with the selected RL algorithm, the proposed agent significantly outperforms similar algorithms available in the literature. Section 5 presents the training and testing results. Finally, Section 6 concludes the paper.

2. Reinforcement Learning Theory—DQN

This paper considers a sequential decision-making process in which an agent interacts with an environment over discrete timesteps [22]. The environment is modeled by states

s_t from set S, and the agent selects the action a_t from set A. To the selected action, the environment responds to the agent by a reward $r_t = r(s_t, a_t)$ and produces the succeeding new state s_{t+1}. Discounted return is defined by $G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$ and the agent seeks to maximize its value. A factor γ is a constant that can take values between 0 and 1, determining the relative value of delayed versus immediate reward.

In terms of future rewards that can be expected, value functions are involved to estimate how good it is for an agent to be in a given state. For an agent behaving according to a stochastic policy π, the value function $v_\pi(s)$ and of the action value function $q_\pi(s, a)$ are defined as follows:

$$v_\pi(s) = \mathbb{E}_\pi[G_t|\ S_t = s] = \mathbb{E}_\pi\left[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+2} + \cdots \middle|\ S_t = s\right] \tag{1}$$

$$q_\pi(s, a) = \mathbb{E}_\pi[G_t|\ S_t = s, A_t = a] = \mathbb{E}_\pi\left[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+2} + \cdots \middle|\ S_t = s, A_t = a\right] \tag{2}$$

The optimal action-value function is defined $q_*(s, a) = \max_\pi q_\pi(s, a)$, for all $s \in S$, $a \in A(s)$. It obeys an important identity known as the Bellman equation:

$$q_*(s, a) = \mathbb{E}[R_{t+1} + \gamma \max_{a'} q_*(s', a')], \text{ for all } s \in S\ ,\ a \in A(s) \tag{3}$$

Deep Q Networks (DQN)

The basic idea behind many reinforcement learning algorithms is to estimate the action-value function (Q-function) by using the Bellman equation as an iterative update. Q-learning [22,23] is an off-policy temporal difference control algorithm defined by:

$$Q(S_t, A_t) \leftarrow\ Q(S_t, A_t) + \alpha\left[R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t)\right] \tag{4}$$

where α represents the learning rate. Such value iteration algorithms converge to the optimal action-value function independent of the policy being followed [22].

In practice, it is common to use a function approximator to estimate the action-value function, such as a neural network, $\hat{q}(s, a, \boldsymbol{w}) \approx q_\pi(s, a)$. Weights $\boldsymbol{w}$ of the neural network function approximator represent the mapping from states to Q-values. A Q-network can be trained by adjusting weight updates by the following:

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t + \alpha\left[R + \gamma \max_a \hat{q}(S_{t+1}, a, \boldsymbol{w}_t) - \hat{q}(S_t, A_t, \boldsymbol{w}_t)\right] \nabla \hat{q}(S_t, A_t, \boldsymbol{w}_t) \tag{5}$$

where $\boldsymbol{w}_t$ is the vector of the network's weights, A_t is the action selected at time step t, and S_t and S_{t+1} are respectively the preprocessed observation input to the network at timesteps t and $t + 1$. The gradient in (5) was computed by backpropagation.

DQN is selected as a fundamental DRL algorithm to train an agent for power system control. However, several limitations of this algorithm are known. Comprehensive research of DQN extensions is in [24]. To overcome overestimation of an algorithm, double DQN is proposed [25]. To generalize learning across actions and to improve policy evaluation in the presence of many similar-valued actions without imposing any change to the underlying reinforcement learning algorithm, Dueling DQN is presented [26]. Prioritized experience replay lets RL agents remember and reuse experiences from the past in order to replay important transitions more frequently, and therefore learn more efficiently [27]. Thus, double dueling DQN with prioritized replay is selected as the baseline model in this work.

3. Test System Description

The modified IEEE 14-bus test system shown in Figure 1 was chosen to assess the performance of the proposed RL agents for safe power network management in significantly disturbed operating conditions.

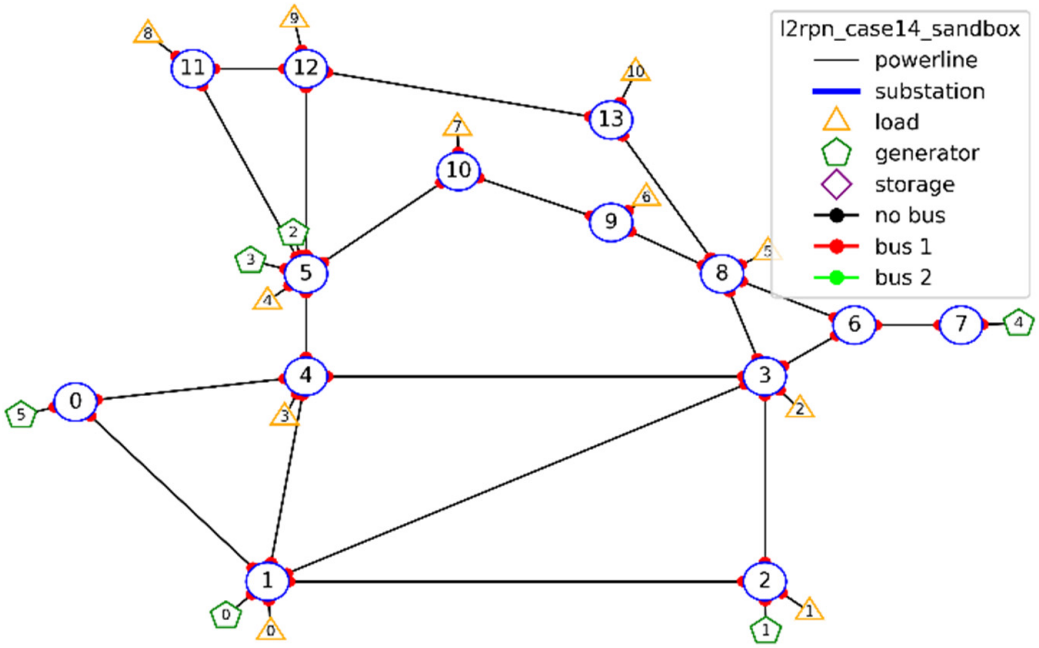
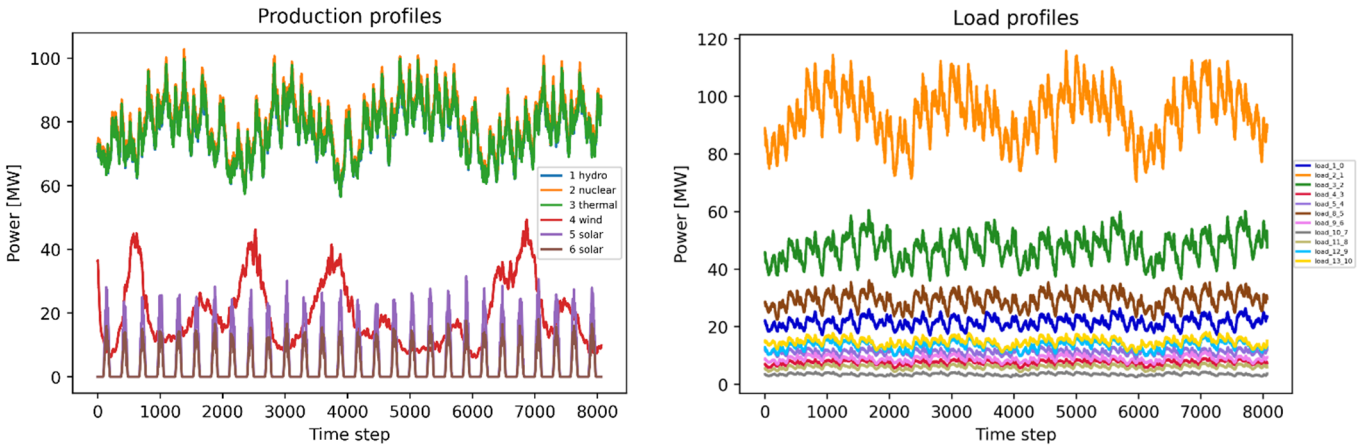


Figure 1. The adopted IEEE 14 bus test system.

Its model includes 14 buses (blue circles), 20 branches, 11 loads (orange circles), and 6 generators (green circles). Generation includes hydro, nuclear, thermal, wind, and two solar power plants to represent the current energy mix. The power grid model is available under the name "l2rpn_case14_sandbox" in Python open-source module Grid2Op [28]. The module also comes with a dataset representing a realistic time series of operating conditions. The dataset for the IEEE 14-bus test system contains 1004 monthly scenarios wherein each represents 28 continuous days in 5-min time intervals. Each scenario includes pre-defined load variations and generation schedules, shown in Figure 2, which are representative of the French grid [21]. The distribution of injections was restricted to be representative of the winter months, over which peak loads are observed [21].



(**a**) (**b**)

Figure 2. (**a**) Production profiles over a typical month. (**b**) Load profiles over a typical month.

3.1. Objective and Conditions of the Power System Control

The main objective is to create an agent that can operate the power grid successfully for as many scenarios as possible using only topology adjustment actions. As in the real-world, the agent must respect several operational constraints to make sure the power grid operates properly. A power system blackout will occur if hard constraints are violated:

1. System demand must be fully served;
2. No generator or load may be disconnected;
3. No electrical islands are formed due to topology changes;
4. AC power flow must converge.

When the power in a line increase above its thermal limit, the line becomes overloaded. It can stay overloaded for 10 min (two timesteps) before it becomes disconnected. If the overload is too high (above 200% of the thermal limit), the line becomes disconnected immediately. This can lead to a cascading failure if other lines become overloaded due to power flow redistribution. The thermal limits of the lines are shown in Table 1. Overloaded lines can be recovered after 50 min (10 timesteps).

Table 1. Line thermal limits.

Line	Thermal Limit (A)	Line	Thermal Limit (A)
0	541	10	442
1	450	11	641
2	375	12	840
3	636	13	156
4	175	14	664
5	285	15	235
6	335	16	119
7	657	17	179
8	496	18	1986
9	827	19	1572

The agent's task is to manage congestion in the power system and prevention of cascading failures and blackouts. For this purpose, we analyze the usage of only grid topology reconfiguration actions.

Available topological actions in the simulator are:

- Reconnecting/disconnecting a line;
- Changing the substation configuration.

Only one substation can be modified per timestep, and every substation can be sectioned into two sections. A 'cooldown time' is 15 min which needs to be respected before a switched line or node can be reused for action.

3.2. Analysis of Scenarios

The dataset for the test system contains 1004 monthly scenarios. Load flow is computed for each scenario with the grid topology being fixed to base topology and with power system control constraints as mentioned in Section 3.1. Results from Figure 3 show, for intervals of timesteps, the number of scenarios with corresponding timesteps before blackout. These results will be a baseline for the evaluation of RL agents since they demonstrate how long the power network can be maintained without any interventions/actions.

The mean number of timesteps in this analysis is 1089 or less than 4 days. A minimal number of timesteps is 3 (15 min) and this occurred in nine scenarios. A maximal number of timesteps is 8064, which corresponds to a successfully managed grid through the entire month and this is recorded in three scenarios.

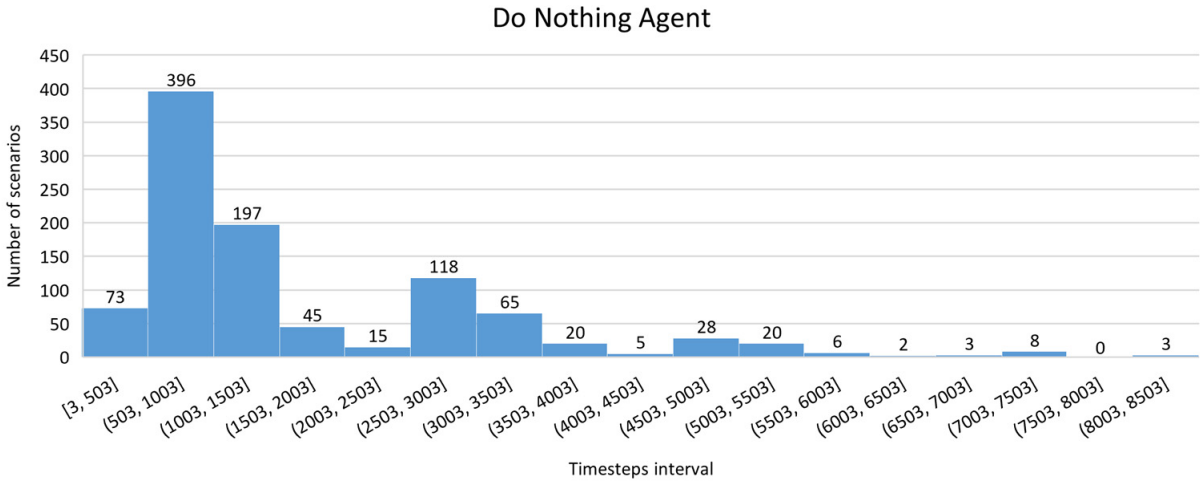


Figure 3. Number of scenarios with associated timesteps for which the power network was operated before the blackout.

To analyze all available data within each scenario—i.e., to go through all of the timesteps in the available dataset—the rules of the power system control are set using 'soft' parameters. The parameters set in this way ensure that lines are not disconnected due to overflow. An overloaded branch is with a loading $\geq$95% of the thermal limit. The highest number of overloads is recorded on branches 9 and 17, while branches 4 and 7 are less frequently overloaded. The average number of overloads on branch 9 per month is 421 timesteps, slightly more than 35 h (about one and a half days). The average number of overloads of branch 17 is 394 timesteps or slightly less than 33 h. Maximal amounts of overload and percentage of timesteps that are overloaded in each scenario are shown in Figure 4.

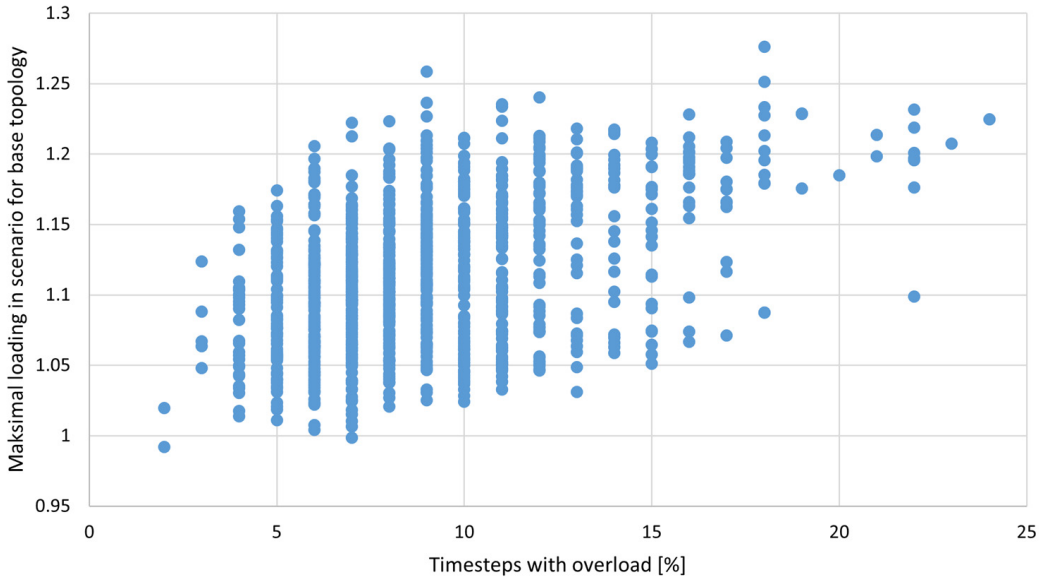


Figure 4. Analysis of highest loading and percentage of timestep with loading of $\geq$0.95 for each of 1004 scenarios.

4. Reinforcement Learning Model

This study intends to apply a state-of-the-art RL approach in an easily applicable way which can serve as a baseline for further algorithm usage for more complex tasks. This paper considers RL algorithms from the standard RL framework RLlib [29], which makes results easily reproducible and verifiable. We consider thoroughly selected actions (based on expert knowledge), reduced observation space, and simple reward.

4.1. Description of Frameworks and Tools Used for This Research

In this section, all tools and frameworks used for this research will be mentioned.

Grid2Op (grid to operate) is an open-source python framework used primarily as a testbed platform for sequential decision making in the world of power system [28]. The simulator can simulate the operation of a power network of any size and characteristic in discrete timesteps. It can simulate cascading outages, where overloaded branches are switched off and the calculation is still carried out considering the following timestep operating conditions. Grid2Op has datasets for several test networks of different sizes and complexities. Grid2Op comes with a machine learning environment that has all the necessary elements for reinforcement learning and which is compatible with the OpenAI Gym framework [30].

OpenAI Gym is an open-source Python library for developing and comparing reinforcement learning algorithms by providing a standard API to communicate between environments and algorithms [30]. It provides an easy API to implement custom environments. The purpose of converting the Grid2op environment into a Gym environment is to make it possible to use it with standard RL frameworks.

Ray is an open-source framework that provides a simple API for building distributed applications and it is a widely used platform for reinforcement learning [31]. Ray can greatly speed up training and make it easier to begin with deep reinforcement learning. Ray is packaged with libraries for accelerating machine learning workloads. A wide range of state-of-the-art algorithms are available through RLlib and can be easily accessed and modified. Ray provides foundations for parallelism and scalability which are simple to use and allow Python programs to scale anywhere from a personal computer to a large cluster.

RL applications can be quite computationally and memory intensive and often need to scale out onto a cluster. A cluster allows using significant computational resources in demanding data processing. Due to that, the learning process for this research was carried out on the high-performance computing (HPC) cluster "Isabella".

4.2. Observation Space, Action Space, and Reward

In agent-environment interaction, as the main parts of reinforcement learning, one should take care of observation space, action space, and reward signal. The goal of investments in the analysis of mentioned parts of RL is to make the agent work. Many analyses were taken before the results become noticeable and worth mentioning. As the result of the analysis in this paper, proper action selection was crucial in reaching the main goal which is secure power system control.

In this study, only topology actions are considered. The goal is to operate the power network as long as possible, and for that reason it is necessary to keep power flow in lines at a desirable level, so they do not become overloaded. Due to this requirement, actions regarding changing the status of power lines are not considered because it is preferable to keep them all connected, as they initially are. Actions regarding busbar splitting are carefully considered. For the RL agent, we focus on the final substation configurations. The number of actions depends on the number of elements connected to the substation and can be calculated as 2^{n-1} where n is the total number of elements connected to the substation. The number of illegal actions—such as disconnecting of load or generator—can be calculated as $2^B - 1$, where B is the number of generators and loads on the substation. Analysis of busbar configurations is in Table 2. If illegal actions are ejected, the overall number of actions for busbar splitting for the observed test network is 179 (with added action for do nothing).

Table 2. Number of configurations for each substation.

Substation	Number of Elements	Configurations	Number of Generators and Loads	Illegal Configurations
0	3	4	1	1
1	6	32	2	3
2	4	8	2	3
3	6	32	1	1
4	5	16	1	1
5	7	64	3	7
6	3	4	0	0
7 *	2	-	1	-
8	5	16	1	1
9	3	4	1	1
10	3	4	1	1
11	3	4	1	1
12	4	8	1	1
13	3	4	1	1
		200	22	
Overall configurations			178	

* Substation 7 cannot be split into sections because there is only one line and one generator connected.

The presented number of actions is too big for this power network. Driven by expert knowledge, the power network does not have such a big number of topologies that would satisfy power network control requirements. Further action selection is based on the basic principle of n − 1 security in network planning states that say if a component should fail or be shut down in a network operating at the maximum forecast levels of transmission and supply, the network security must still be guaranteed. This leads to incorporating the following constraints:

1. A minimum number of elements that must be connected on each substation is 2;
2. At least 2 of the elements connected to a substation must be a line.

With the two above criteria, the number of potential substations for splitting is halved. Since only one of these topology actions can be chosen at each timestep, the final selection of actions would consider only actions that lead to desirable power network topologies. The depth of changes—i.e., number of substations that are split at the same time—is selected to be $\leq$3 for this power network. Thus, only three substations are considered for the final selection of potential topologies. Besides selected actions presented, in the action space actions are added that set the initial topology of substations (every bus element is connected to the same bus section) and action for 'do nothing' which results in a final action space with only nine actions shown in Table 3.

An observation is a description of the power grid perceived by an agent. The observation space with all available observations at the current timestep has 368 features as shown in Table 4. We applied reduction on the observation space, so it contains voltages and currents at both sides of branches, the capacity of each powerline, and the topology vector that for each object (load, generator, ends of a powerline) gives on which bus this object is connected in its substation. Reduced observation space contains 157 features and selected attributes are bolded in Table 4.

Table 3. Selected action space and topologies.

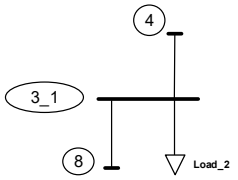
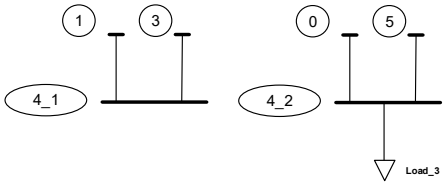
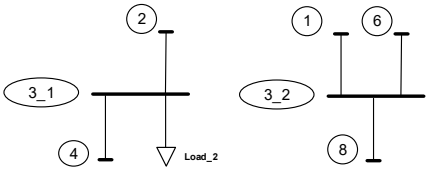
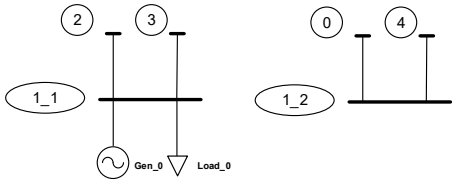
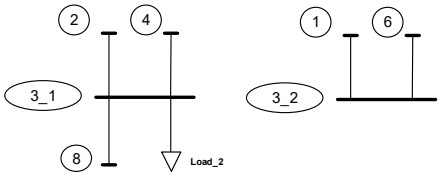
Action	Topology	Action	Topology
Action 1		Action 4	
Action 2		Action 5	
Action 3		Action 6	Initial topology substation 3
		Action 7	Initial topology substation 4
		Action 8	Initial topology substation 1
		Action 9	Do nothing

Table 4. Main observation attributes.

Attribute	Name	Type	Size
Date	year, month, day, hour_of_day, minute_of_hour, day_of_week	int	6
Active power	gen_p, load_p, p_or, p_ex	float	57
Reactive power	gen_q, load_q, q_or, q_ex	float	57
Voltage	gen_v, load_v, **v_or**, **v_ex**	float	57
Current	**a_or**, **a_ex**	float	40
Capacity of powerline	**rho**	float	20
Topology vector	**topo_vect**	int	57
Status of powerline	line_status	bool	20
Timesteps since a powerline is in overflow	timestep_overflow	int	20
Number of timesteps the substation is unavailable	time_before_cooldown_sub	int	14
Number of timesteps the powerline is unavailable	time_before_cooldown_line	int	20

The input layer of the neural network has the size of observation space 157. The size of the output layer corresponds to the size of the action space (9). There are also two hidden layers, each with 128 neurons.

Finally, we used a simple reward function that just counts the number of timesteps the agent has successfully managed to perform. It adds a constant reward for each timestep successfully handled.

4.3. Algorithm Selection and Hyperparameters Used

Due to the nature of the selected observation space and action space type which are both discrete, as well as algorithm simplicity, DQN and its derivatives are selected for analysis. The deep Q Network algorithm and its derivatives observed for this research are from the standard RL library RLlib. They implement all of the DQN improvements evaluated in [24]. DQN framework is shown in Figure 5. Among DQN derivatives, the best performing algorithm is dueling double DQN with prioritized replay and it is used in this research.

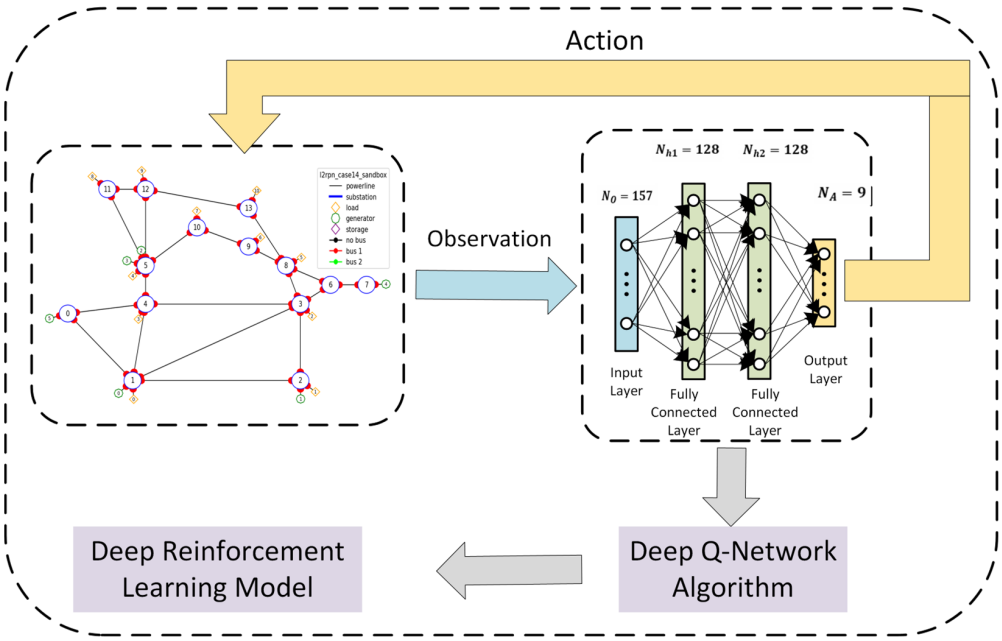


Figure 5. DQN framework.

The hyperparameters used in this research for training RL agent are default parameters set in Ray RLlib for the DQN algorithm, except for the hyperparameters which are tuned as listed in Table 5.

Table 5. Hyperparameters used in this research.

Hyperparameter	Value
model_fcnet_hiddens	[128,128]
model_fcnet_activation	relu
num_workers	4
learning_start	5000
train_batch_size	64
target_network_update_freq	256
learning rate schedule	lr_start = 10×10^{-5} lr_end = 10×10^{-6} lr_time = 100,000
exploration	initial_epsilon = 0.4 final_epsilon = 1/2016 epsilon_timesteps = 100,000

5. Results

In this section, the performance of the proposed approach is analyzed. First, the training phase and results for several agents are presented. Second, evaluation of agent

testing is presented on 100 unseen scenarios. The testing process is described to demonstrate that using the proposed RL agent to control the power network leads to greatly improved performance. A comparison of the results is performed with similar agents from the literature to illustrate the benefits of the proposed approach. Third, the application effects of the proposed agent are shown. Forth, the limitations of the presented algorithm are presented.

5.1. Training Phase

An episode is a sequence of observations, actions, and rewards from an initial state to a terminal state, either succeeding or failing to manage the grid for the duration. For the training phase, we used a set of 800 scenarios available from the dataset for the adopted IEEE 14-bus network used in research from [28]. Mean episode length is tracked during training since the main purpose of the agent is to manage the power network as long as it is possible. During training, early stopping is used to stop training when the agent's performance is satisfactory.

Due to the nature of the selected observation and action space types (both discrete) and algorithm simplicity, DQN and its derivatives are selected for analysis. DQN algorithms considered in this paper are available in the RLlib package named DQN, double DQN, dueling DQN, dueling double DQN, and dueling double DQN with prioritized replay. The training was conducted with 1000 episodes and the results are in Figure 6. According to the results, DDDQN with prioritized replay has the best performance, so further analyses were conducted with that agent.

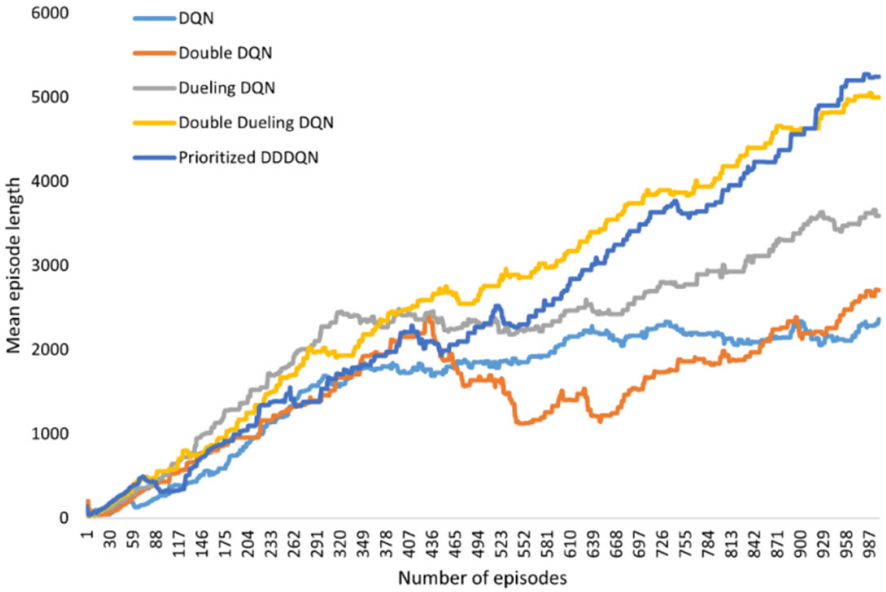


Figure 6. Comparison of the convergence speed between the proposed algorithm and other DQN derivative algorithms.

Figure 7 shows the training progress of the agent for several observation spaces. The yellow curve represents the final agent whose performance will be evaluated in unseen scenarios. Compared to the other agents, the selected agent can control the power network for a longer period (timesteps) in the earlier phases of the training process. The agents' observed metric is monotonically increasing, indicating that the agent is learning. We made our agent act only in hazardous situations when the power flow of a given line is larger than the threshold (in this case 95% of the thermal limit) and the topology of the power grid is not optimal. In this case, training took 1096 episodes.

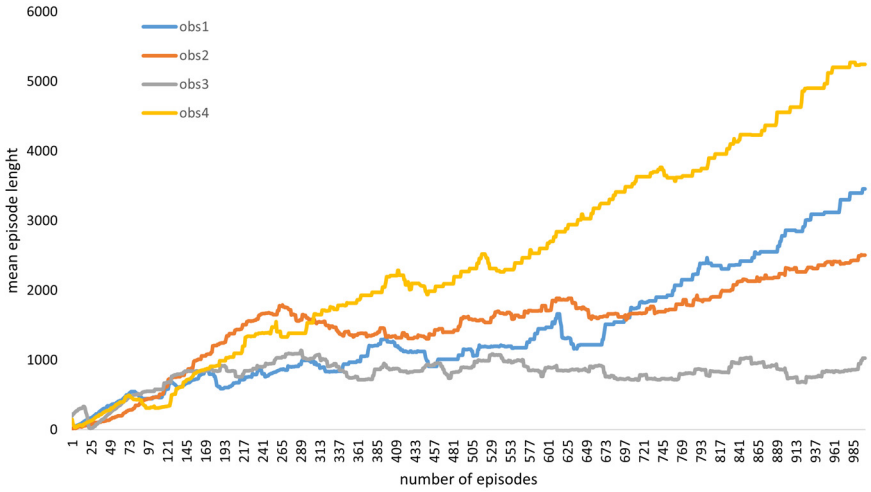


Figure 7. Training progress of the agent monitored by mean episode length for several observation spaces.

5.2. Testing Phase

To evaluate the agent, we used 100 scenarios that were not used in the training phase to see how the agent would perform in unseen scenarios. Evaluation results of the proposed RL agent are compared to the do-nothing agent. The result of the test is in Figure 8. Results for the do-nothing agent illustrate that the plugged-in renewable energy sources will lead to heavy overload in the power system if no control measure is taken. In that case, the system will blackout at an average number of timesteps around 1528 (about 5 days). Using the proposed RL agent to control the power network leads to greatly improved performance. For all scenarios, the proposed RL agent automatically operates the power grid for longer than a day (288 timesteps) without expert help. For only 11 scenarios, it manages the power network for less than a week; and for 62 scenarios, it operates the grid successfully through the entire month (8064, maximum number of timesteps). On average, it controls the power network for 6574 timesteps, or almost 23 days.

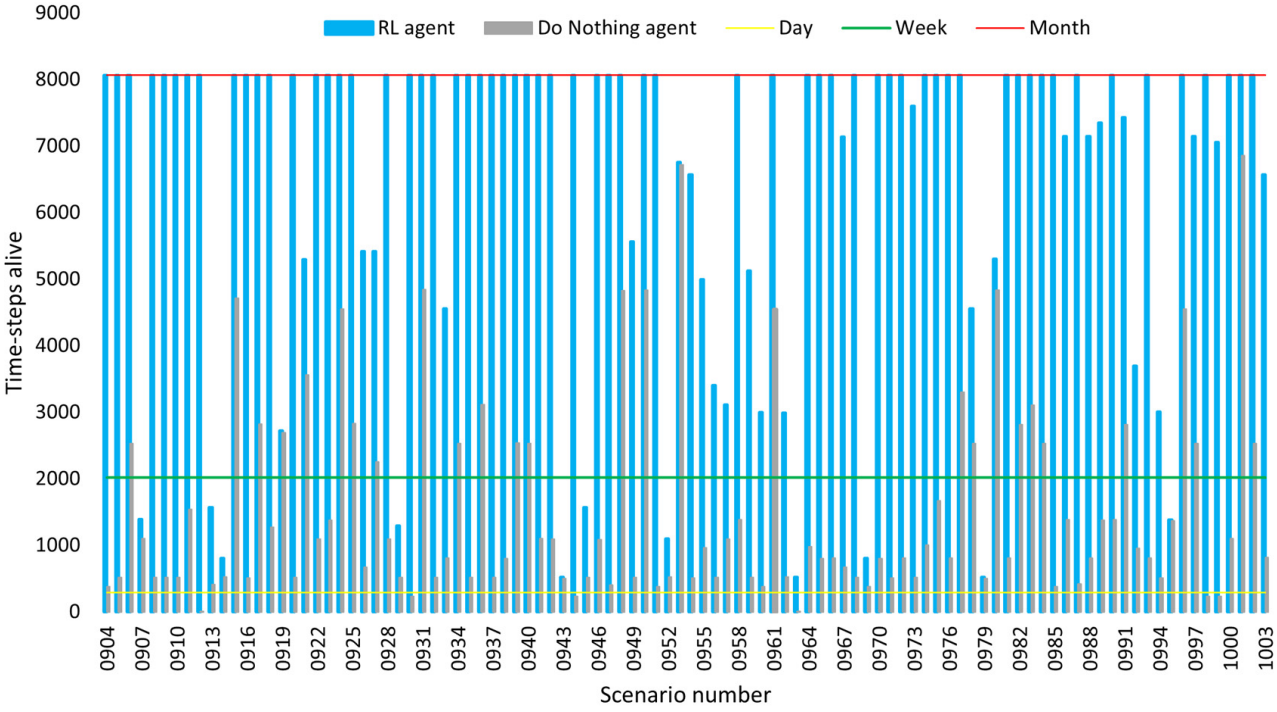


Figure 8. Performance of RL agent in 100 test scenarios.

For a fair comparison, our agent is compared with agents from the literature that aim to control the IEEE 14-bus power network with only topological actions and/or use the same RL algorithm (dueling double DQN). The details for observed agents from the literature and the results of our agent are summarized in Table 6. Besides a much longer period of managing the power network, our approach is much simpler since it does not require additional learning techniques. Observation and action spaces are reduced.

Table 6. Overview of presented approaches from literature and comparison with RL agent proposed in this research.

Algorithm	Additional Learning Techniques	Topology Actions	Observation Space	Action Space	Neural Network Architecture	Autonomously Control Power Network
Cross-entropy method [18]	Importance sampling	Busbar splitting	324	112	Feed-forward [324,300,300,112]	Up to 1 week
Double dueling DQN [19]	Imitation learning, guided exploration, importance sampling	Busbar splitting and line switching	538	251	Batch normalization layer added to the input layer [538,256,128,251,251]	288 timesteps (1 day)
Double dueling DQN [20]	Embedding layers—graph neural networks (GNNs)	Busbar splitting	194	160	Embedding layers (6 GNN block) and DDQN from [19]	Not much better than the do-nothing agent (training scenarios 864 timesteps—3 days)
Double dueling DQN proposed in this research	-	Busbar splitting	157	9	Feed-forward [157,128,128,9]	Up to 1 month (never less than 1 day)

The proposed DRL agent managed the grid in a way that it determines optimal power network configuration and set it at beginning of the episode to prevent cascading failure and blackouts in the smart grid. The agent learned that the initial topology is not optimal, and that optimal topology can be set by a combination of actions 3 and 5. Setting optimal topology at the beginning of the episode made power system control much easier since the sequence of decision-making actions is much shorter in disturbing situations. This approach belongs to system-level control and considers only modifying substation configuration. The control is designed to work in normal operating states of the system and applies control actions to control flows over transmission lines.

5.3. Performance of the Proposed Method

To fully evaluate the performance of the proposed DRL agent, we selected four scenarios with a different share of renewables shown in Figure 9. Cases 2 and 3 are similar in energy profiles, but case 3 is interesting because a blackout occurs in the network in the third timestep if no actions are taken. Case 4 is interesting due to the larger share of energy from renewable resources, where wind and solar energy made up almost 20% of total production.

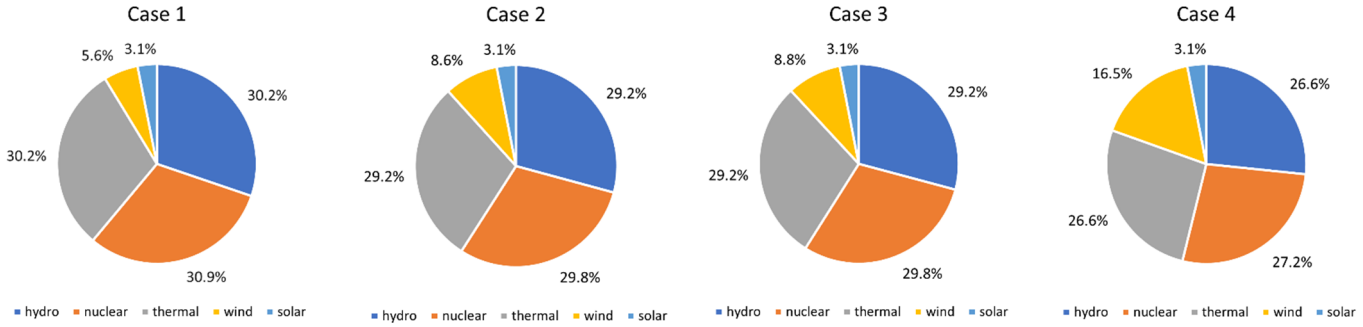


Figure 9. Energy profiles of each case.

To demonstrate the application effects of DRL, we investigate agent control performance for each case. Since the objective of the agent is power flow control to prevent cascading failures and blackouts, we focus on the maximum line capacity usage. We record the maximum line capacity usage for selected cases. A comparison is made between the do-nothing agent and our proposed RL agent. As shown in Figure 10, in all cases the agent controlled the network longer and made the maximum line capacity usage ratio lower than the do-nothing agent. Case 3 was very challenging, wherein overload occurred at the beginning of the scenario and the agent had to respond quickly to eliminate overload and prevent a blackout. This demonstrates the adaptability of our proposed control method.

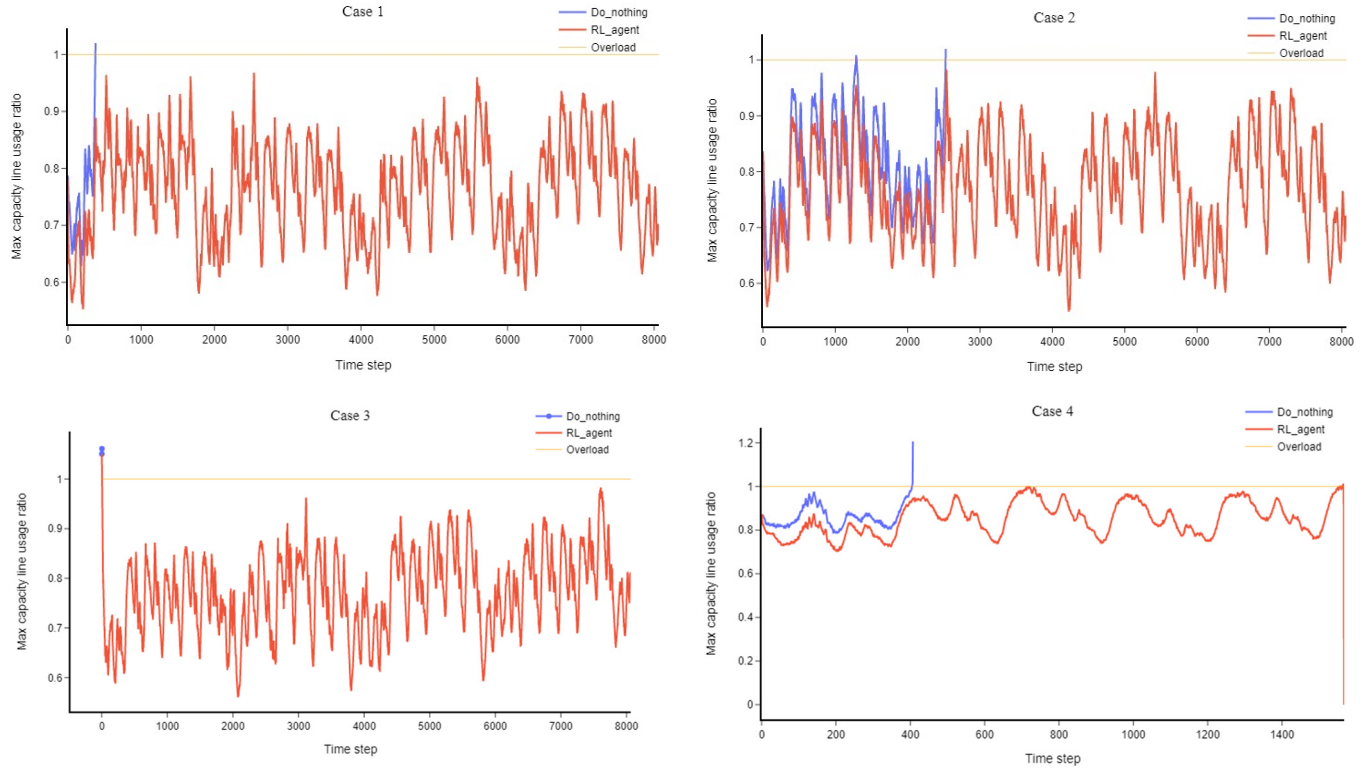


Figure 10. Maximum line capacity usage in the scenarios for the do nothing and DRL agents.

5.4. Limitations of the Work

The limitation of the proposed method is a limitation of the RL algorithm used in this research. DQN algorithms support only discrete actions. This means that future extensions with continuous actions, such as generator redispatch, should be carried out carefully. In this case, to alleviate this problem, continuous actions must be converted into discrete actions. The continuous features should be subjected to techniques such as binning and

clustering that group continuous values into discrete bins, thereby discretizing continuous features into discrete ones.

6. Conclusions

All tools and frameworks used in this research are open-source, and the algorithm is used as it is in the standard RL library without further improvements in the learning process. This is in contrast to previous studies where more complicated RL algorithms are employed and additional techniques such as supervised learning, imitation learning, or guided exploration are needed.

Besides highlighted simplicity and reproducibility, we empirically demonstrated that the presented method significantly outperforms similar agents available in the literature. This work shows the possibility of an intelligent agent that automatically operates the power grid without expert help with only a less costly method—substation reconfiguration. The possibility of an agent controlling a power network for up to an entire month with similar methods is not recorded in literature according to the author's knowledge.

Future work can include the application of RL agents on networks bigger and more constrained than the IEEE 14-bus test system with varied RE penetration. Incorporation of other control variables (such as line switching, transformer tapping control, and generator dispatch) in the RL agent formulation is also recommended. It is also recommended that RL algorithms other than the DQN be explored to improve the overall performance and computational speed of the training. Aside from secure power system management, power system losses should be considered as a further improvement of the proposed method.

Author Contributions: Conceptualization, I.D., I.P., M.P. and M.B.; Data curation, I.D., I.P. and M.B.; Formal analysis, I.D., I.P., M.P. and M.B.; Investigation, I.D. and I.P.; Methodology, I.D., I.P., M.P. and M.B.; Software, I.D. and M.B.; Validation, I.D., I.P. and M.P.; Writing—original draft, I.D., I.P., M.P. and M.B.; Writing—review & editing, I.D., I.P., M.P. and M.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Acknowledgments: This research was performed using resources of computer cluster Isabella based in SRCE—University of Zagreb University Computing Centre.

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

References

1. Eremia, M.; Shahidehpour, M. *Handbook of Electrical Power System Dynamics*; Wiley-IEEE Press: Hoboken, NJ, USA, 2013; ISBN 978-1-118-49717-3.
2. Zhang, D.; Han, X.; Deng, C. Review on the research and practice of deep learning and reinforcement learning in smart grids. *CSEE J. Power Energy Syst.* **2018**, *4*, 362–370. [CrossRef]
3. Zhang, Z.; Zhang, D.; Qiu, R.C. Deep reinforcement learning for power system: An overview. *CSEE J. Power Energy Syst.* **2019**, *6*, 213–225. [CrossRef]
4. Cao, D.; Hu, W.; Zhao, J.; Zhang, G.; Zhang, B.; Liu, Z.; Chen, Z.; Blaabjerg, F. Reinforcement learning and its applications in modern power and energy systems: A review. *J. Mod. Power Syst. Clean Energy* **2020**, *8*, 1029–1042. [CrossRef]
5. Yang, T.; Zhao, L.; Li, W.; Zomaya, A.Y. Reinforcement learning in sustainable energy and electric systems: A survey. *Annu. Rev. Control* **2020**, *49*, 145–163. [CrossRef]
6. Chen, X.; Qu, G.; Tang, Y.; Low, S.; Li, N. Reinforcement learning for selective key applications in power systems: Recent advances and future challenges. *IEEE Trans. Smart Grid* **2022**, *13*, 2935–2985. [CrossRef]
7. Glavic, M.; Fonteneau, R.; Ernst, D. Reinforcement learning for electric power system decision and control: Past considerations and perspectives. *IFAC-PapersOnLine* **2017**, *50*, 6918–6927. [CrossRef]
8. Xu, Y.; Zhang, W.; Liu, W.; Ferrese, F. Multiagent-based reinforcement learning for optimal reactive power dispatch. *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.* **2012**, *42*, 1742–1751. [CrossRef]
9. Vlachogiannis, J.G.; Hatziargyriou, N.D. Reinforcement learning for reactive power control. *IEEE Trans. Power Syst.* **2004**, *19*, 1317–1325. [CrossRef]
10. Zarrabian, S.; Belkacemi, R.; Babalola, A.A. Reinforcement learning approach for congestion management and cascading failure prevention with experimental application. *Electr. Power Syst. Res.* **2016**, *141*, 179–190. [CrossRef]

11. Ye, D.; Zhang, M.; Sutanto, D. A hybrid multiagent framework with q-learning for power grid systems restoration. *IEEE Trans. Power Syst.* **2011**, *26*, 2434–2441. [CrossRef]
12. Jasmin, E.A.; Imthias Ahamed, T.P.; Jagathy Raj, V.P. Reinforcement learning approaches to economic dispatch problem. *Int. J. Electr. Power Energy Syst.* **2011**, *33*, 836–845. [CrossRef]
13. Li, D.; Yu, L.; Li, N.; Lewis, F. Virtual-action-based coordinated reinforcement learning for distributed economic dispatch. *IEEE Trans. Power Syst.* **2021**, *36*, 5143–5152. [CrossRef]
14. Duan, J.; Shi, D.; Diao, R.; Li, H.; Wang, Z.; Zhang, B.; Bian, D.; Yi, Z. Deep-reinforcement-learning-based autonomous voltage control for power grid operations. *IEEE Trans. Power Syst.* **2019**, *35*, 814–817. [CrossRef]
15. Lukianykhin, O.; Bogodorova, T. Voltage control-based ancillary service using deep reinforcement learning. *Energies* **2021**, *14*, 2274. [CrossRef]
16. Lu, R.; Hong, S.H. Incentive-based demand response for smart grid with reinforcement learning and deep neural network. *Appl. Energy* **2019**, *236*, 937–949. [CrossRef]
17. Zhang, X.; Lu, R.; Jiang, J.; Hong, S.H.; Song, W.S. Testbed implementation of reinforcement learning-based demand response energy management system. *Appl. Energy* **2021**, *297*, 117131. [CrossRef]
18. Subramanian, M.; Viebahn, J.; Tindemans, S.H.; Donnot, B.; Marot, A. Exploring grid topology reconfiguration using a simple deep reinforcement learning approach. In Proceedings of the 2021 IEEE Madrid PowerTech, Madrid, Spain, 28 June–2 July 2021. [CrossRef]
19. Lan, T.; Duan, J.; Zhang, B.; Shi, D.; Wang, Z.; Diao, R.; Zhang, X. AI-based autonomous line flow control via topology adjustment for maximizing time-series ATCs. In Proceedings of the IEEE Power and Energy Society General Meeting, Montreal, QC, Canada, 2–6 August 2020; Volume 2020.
20. Yoon, D.; Hong, S.; Lee, B.-J.; Kim, K.-E. Winning the L2RPN challenge: Power grid management via semi-markov afterstate actor-critic. In Proceedings of the International Conference on Learning Representations, Addis Ababa, Ethiopia, 26–30 April 2020; pp. 1–17.
21. Marot, A.; Donnot, B.; Romero, C.; Donon, B.; Lerousseau, M.; Veyrin-Forrer, L.; Guyon, I. Learning to run a power network challenge for training topology controllers. *Electr. Power Syst. Res.* **2020**, *189*, 106635. [CrossRef]
22. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*; The MIT Press: London, UK, 2018; ISBN 9788578110796.
23. Watkins, C.J.C.H.; Dayan, P. Q-learning. *Mach. Learn.* **1992**, *8*, 279–292. [CrossRef]
24. Hessel, M.; Modayil, J.; Van Hasselt, H.; Schaul, T.; Ostrovski, G.; Dabney, W.; Horgan, D.; Piot, B.; Azar, M.; Silver, D. Rainbow: Combining improvements in deep reinforcement learning. In Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI-18), New Orleans, LA, USA, 2–7 February 2018; pp. 3215–3222.
25. Hasselt, H. Double Q-learning. In *Advances in Neural Information Processing Systems*; Lafferty, J., Williams, C., Shawe-Taylor, J., Zemel, R., Culotta, A., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2010; Volume 23.
26. Wang, Z.; Schaul, T.; Hessel, M.; Van Hasselt, H.; Lanctot, M.; De Frcitas, N. Dueling network architectures for deep reinforcement learning. In Proceedings of the 33rd International Conference on Machine Learning, ICML 2016, New York, NY, USA, 19–24 June 2016; Volume 4, pp. 2939–2947.
27. Schaul, T.; Quan, J.; Antonoglou, I.; Silver, D. Prioritized experience replay. In Proceedings of the 4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, 2–4 May 2016.
28. Grid2Op. Available online: https://github.com/rte-france/Grid2Op#grid2op (accessed on 1 March 2022).
29. Liang, E.; Liaw, R.; Moritz, P.; Nishihara, R.; Fox, R.; Goldberg, K.; Gonzalez, J.E.; Jordan, M.I.; Stoica, I. RLlib: Abstractions for distributed reinforcement learning. In Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholm, Sweden, 10–15 July 2018; Volume 7, pp. 4768–4780.
30. OpenAI Gym. Available online: https://github.com/openai/gym (accessed on 1 March 2022).
31. Ray. Available online: https://github.com/ray-project/ray (accessed on 1 March 2022).