Towards Explainable Attacker-Defender Autocurricula in Critical Infrastructures

Eric MSP Veith and Torben Logemann
Carl von Ossietzky University Oldenburg
Research Group Adversarial Resilience Learning
Oldenburg, Germany
Email: {eric.veith,torben.logemann}@uol.de

***Abstract*—Agent systems have become almost ubiquitous in smart grid research. Research can be roughly divided into carefully designed (multi-) agent systems that can perform known tasks with guarantees, and learning agents based on technologies such as Deep Reinforcement Learning (DRL) that promise real resilience by learning to counter the unknown unknowns. However, the latter cannot give guarantees regarding their behavior, while the former are limited to the set of problems known at design time. In this paper, we present work in progress towards explaining strategies learned in autocurriculum settings in Critical National Infrastructures (CNIs), such as the power grid. We show how our equivalent representation of DRL policies allows to study agent behavior and ascertain learned strategies for resilient CNI operation.**

Keywords*—adversarial resilience learning; agent systems; reinforcement learning; explainable reinforcement learning; resilience; power grid

I. Introduction

Over the last years, agent systems and especially Multi-Agent Systems (MASs) [1]–[4] have emerged as one of the most important tools to facilitate management of complex energy systems. As swarm logic, they can handle numerous tasks, such as maintaining real power equilibria, voltage control, or automated energy trading [5]. The fact that MASs implement proactive and reactive distributed heuristics allows to analyze their behavior and give certain guarantees, a property that has helped in their deployment.

However, modern energy systems have also become valuable targets. Cyber-attacks have become more common [6], [7], and establishing local energy markets, although being an attractive concept of self-organization, can also be exploited, e. g., through artificially created congestion [8]. Attacks on power grids are no longer carefully planned and executed, but also learned by agents, such as market manipulation or voltage band violations [9]. Thus, carefully designing software systems that provide protection against a widening field of adversarial scenarios has become a challenge, especially considering that (interconnected) Cyber-Physical Systems (CPSs) are inherently exploitable due to their complexity [10].

Learning agents, particularly those based on DRL, have gained traction as a potential solution: If a system faces *unknown unknowns*, a learning agent can devise strategies against it. In the past, researchers have employed DRL-based agents for numerous tasks related to power grid operations, such as voltage control [11]. Especially the approach to use DRL for vulnerability assessment, cyber security attack mitigation, and general resilient operation have gained traction among researchers in the recent years [12]–[16]. In general, DRL constitutes an attractive family of algorithms as it is at the core of many noteworthy successes, such as MuZero [17], with modern algorithms such as Twin-Delayed DDPG (TD3) [18], Proximal Policy Gradient (PPO) [19], and Soft Actor Critic (SAC) [20] having proved to be able to tackle complex tasks.

While the scientific corpus agrees that DRL-based agents are a valuable topic of research in terms of cyber-security in CNIs, their effectiveness can only be stated in a manner that is (1) indirect and (2) case-based. Indirect, because there is no direct method available that would ascertain a DRL agent's policy. Publications offer analysis of rewards and simulation states; however, it is well known that optimizing a metric (i. e., maximizing the reward) is not necessarily the same as solving the problem behind it. Second, many publications lack long-term simulations, but consider certain well-described scenarios. Thus, a DRL-based agent's ability to generalize is inferred, but not entirely proven.

eXplainable Reinforcement Learning (XRL) [21] promises to fill this gap at least partially. However, the most common techniques, such as saliency maps, give only indirect interpretation and are useful for experts in the DRL domain, but not for practitioners in CNIs. Recent approaches to convert a DRL agent's policy network into a rule-based representation, e. g., as decision tree [22], will satisfy the outlined requirements. In a recent publication, we have presented an equivalent transformation of a DRL agent's policy network into a compressed decision tree, called *NN2EQCDT* [23]. We have also argued that such an equivalent representation should be a default module in any modern architecture for learning agents in CNIs and presented the Adversarial Resilience Learning (ARL) agent architecture in this regard [24].

In this paper, we present an approach to explaining and validating DRL autocurricula in CNIs, such as the power grid. Previous publications have indicated that employing competing agents can lead to faster learning and robust strategies, and we have presented our ARL methodology to take advantage of this [12]. In ARL, two agents (often dubbed "attacker" and "defender") work with an inversible reward function: The defender aims to operate the CNI in a resilient manner, the attacker aims to destabilize the CNI. The competition improves the sample efficiency of the agents, which also learn more robust strategies. As the goal of the ARL research is to develop an actually deployable defender, an extended architecture (the

ISBN: 978-1-68558-113-8

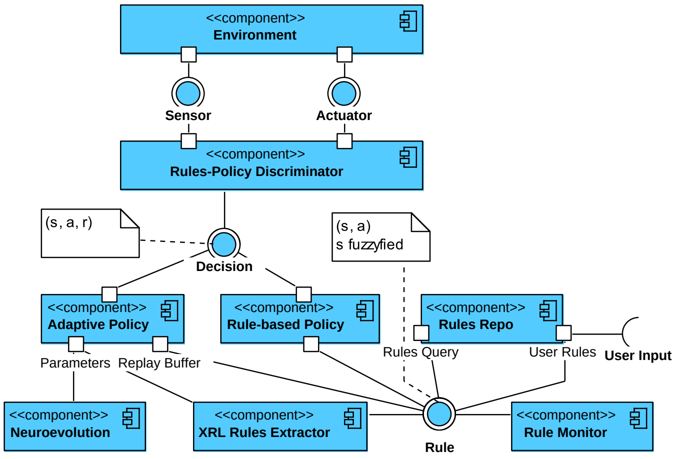

Figure 1. Simplified components view of the ARL agent architecture.

ARL architecture) has been created. In this paper, we will outline how the generation of an equivalent representation of a policy network can be integrated in an agent architecture and provide the first steps towards explaining DRL autocurriculae for resilient operation of smart grids.

The remainder of this work-in-progress paper is structured as follows: Section II gives a concise summary of our NN2EQCDT algorithm and its integration into the ARL agent architecture. In Section III, we then present a scenario that we explain using NN2EQCDT. Section IV offers a discussion of our approach and the experiment's results. Finally, we outline the next steps in Section V.

II. A Self-Explaining Deep Reinforcement Learning Agent

The concept of the ARL agent assumes two parallel policies: An *Adaptive Policy* that is based on DRL, and a *Rules-based Policy* that works on a decision tree. When an agent observes the environment, the *Discriminator* chooses between the two policies based on a trust value. Both policies are queried, and in their *Decision*, they give the action and the reward value they expect from executing the action. The Discriminator checks both proposals against its internal world model and chooses the one whose reward deviates the least from the reward the world model returns. Then, each policy's trust value is modified according to a Linear Time-Invariant (LTI) system:

$$pt1(y, u, t) = \begin{cases} u & \text{if } t = 0 \\ y + \dfrac{u - y}{t} & \text{otherwise,} \end{cases} \tag{1}$$

where y signifies the current trust value of the respective policy module and u is the reward the world model yielded for the policy's decision proposal. The Discriminator's world model is based on data provided by the CNI operator.

The truest approach also means that the adaptive policy will naturally be trusted for situations not covered by rules, but is able to gain more trust to yield innovative strategies over the course of the agent's existence, while the LTI ensures that mistakes do not immediately void the trust.

Whenever the DRL policy retrains, the new policy network is transformed into a new decision tree using the *XRL Rules Extractor*, which implements our NN2EQCDT algorithm [23]. Figure 1 depicts the component architecture of the ARL agent, while Figure 2 shows the procedure described.

The NN2EQCDT algorithm works according to Figure 3. The weight and bias matrices $\boldsymbol{W}_i$ and $\boldsymbol{B}_i$ from the Feed-Forward Deep Neural Network (FF-DNN) model are processed layer by layer. These are used to compute rules that are used to add subtrees to the overall Decision Tree (DT). From the second layer, when multiplying the weight and bias matrices, it is necessary to take into account the position of the node to which the generated subtree will be attached. This is done by applying the slope vector $\boldsymbol{a}$ to the current weight matrices. It represents the node position of the connection, since it is the vector of choices according to the Rectified Linear Unit (ReLU) activation function along the path from the root to the connection node.

When adding a node of a newly created subtree to the overall tree, each path from the root to the node in question is checked for satisfiability. If there can be no input so that its evaluation of the DT that takes this path, the node in question and thus further subtrees are not added to keep the size of the DT dynamically small.

Finally, the last checks are converted into expressions, and the DT can be further compressed by removing unnecessary checks, since they are evaluated the same for all possible inputs.

III. Example of Application

The ability to compress policies is important for an effective operation of the hybrid DRL/rules-based agent. Not only inference, but also analysis of extracted rules (e. g., changes with regards to previous iterations) takes advantage of a small tree. Considering that the ARL agent will run on edge devices that are memory- and CPU-constrained, the ability to compress the tree becomes an important feature of the algorithm. As a first step in our work in progress, we experimentally tested how an extrated decision tree is dependent on the size of the policy network, even if the strategy the agent has learned is seemingly simple.

To test this, we constructed a power grid with a simple linear branch feeder. From the 110 kV/20 kV transformer extends a branch with four nodes:

1) an inverter (Photovoltaics, PV), controlled by a "attacker" agent
2) an inverter (PV), controlled by a "defender" agent
3) an independent hospital
4) an independent wind park.

True to the ARL autocurriculum setting, we provided two largely invertable objectives to the agent, both of which targeted the voltage band. The attacker's task was to violate the voltage band, whereas the defender should keep it within acceptable boundaries. We used a bell-shaped curve centered at $1.0\,\mathrm{pu}$. The defender maximum reward was at $1.0\,\mathrm{pu}$, while the attacker used the inverted curve, with maximum reward at $V < 0.8\,\mathrm{pu}$ and $V > 1.1\,\mathrm{pu}$, respectively. Consider the reward function:

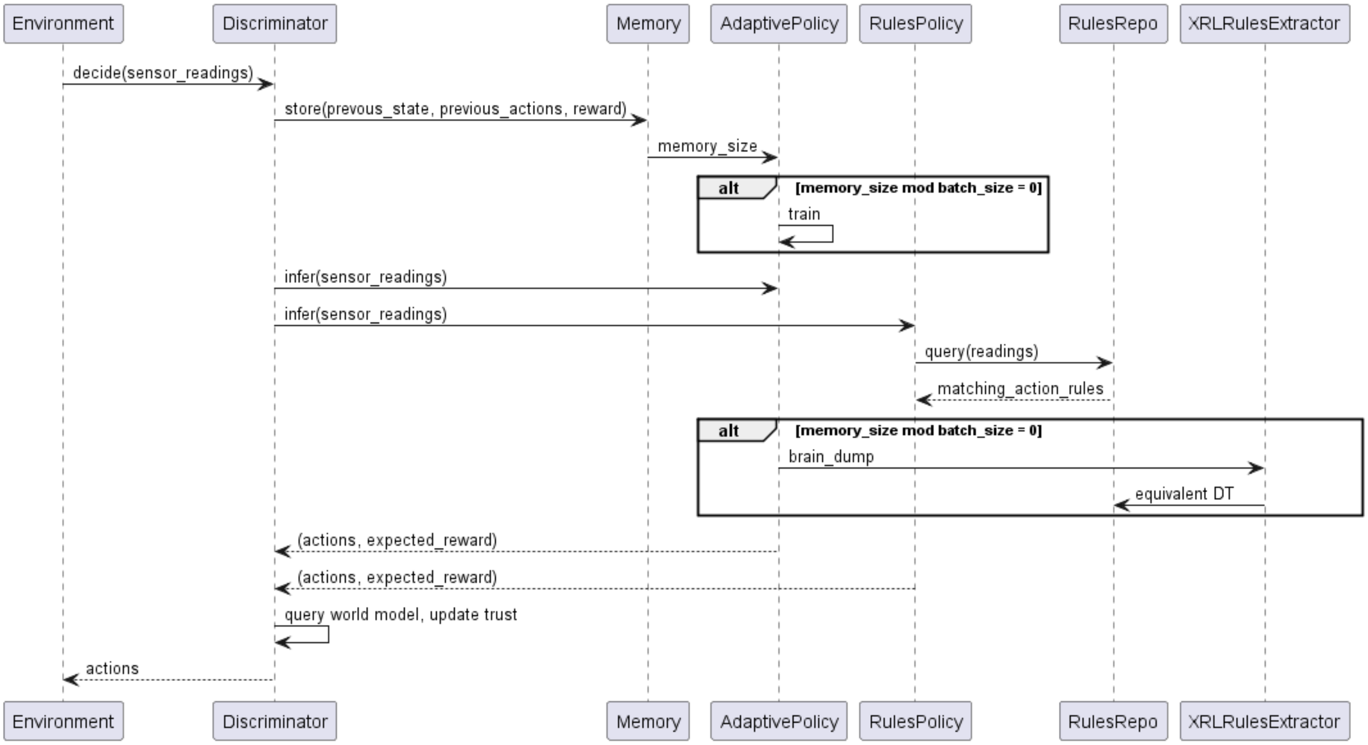

Figure 2. Activity diagram for training and self-explaining of an ARL agent.

```
$\hat{\boldsymbol{W}} = \boldsymbol{W}_0$
$\hat{\boldsymbol{B}} = \boldsymbol{B}_0^\top$
$rules = \text{calc\_rule\_terms}(\hat{\boldsymbol{W}}, \hat{\boldsymbol{B}})$
$T, new\_SAT\_leaves = \text{create\_initial\_subtree}(rules)$
$\text{set\_hat\_on\_SAT\_nodes}(T, new\_SAT\_leaves, \hat{\boldsymbol{W}}, \hat{\boldsymbol{B}})$
for $i = 1, \dots, n-1$ do
    $SAT\_paths = \text{get\_SAT\_paths}(T)$
    for $SAT\_path$ in $SAT\_paths$ do
        $\boldsymbol{a} = \text{compute\_a\_along(SAT\_path)}$
        $SAT\_leave = \text{SAT\_path}[-1]$
        $\hat{\boldsymbol{W}}, \hat{\boldsymbol{B}} = \text{get\_last\_hat\_of\_leave}(T, SAT\_leave)$
        $\hat{\boldsymbol{W}} = (\boldsymbol{W}_i \odot [(\boldsymbol{a}^\top)_{\times k}])\hat{\boldsymbol{W}}$
        $\hat{\boldsymbol{B}} = (\boldsymbol{W}_i \odot [(\boldsymbol{a}^\top)_{\times k}])\hat{\boldsymbol{B}} + \boldsymbol{B}_i^\top$
        $rules = \text{calc\_rule\_terms}(\hat{\boldsymbol{W}}, \hat{\boldsymbol{B}})$
        $new\_SAT\_leaves =$
        $\text{add\_subtree}(T, SAT\_leave, rules, invariants)$
        $\text{set\_hat\_on\_SAT\_nodes}(T, new\_SAT\_leaves,$
        $\hat{\boldsymbol{W}}, \hat{\boldsymbol{B}})$
$\text{convert\_final\_rule\_to\_expr}(T)$
$\text{compress\_tree}(T)$
```

Figure 3. NN2EQCDT algorithm for generating equivalent representation of DRL policy networks.

$$g\left(x = \frac{\sum_{i=1}^{|\boldsymbol{V}|} V_i}{|\boldsymbol{V}|}, A, \mu, C, \sigma\right) = A\cdot\exp\left(-\frac{(x-\mu)^2}{2\sigma^2} - C\right), \tag{2}$$

where $\boldsymbol{V}$ are voltages at the observed "victim buses" to which the hospital and the wind park are connected. The parameters A, μ, C, and σ shape the curve, so that we define:

$$\begin{aligned} reward_{attacker}\left(x = \frac{\sum_{i=1}^{|\boldsymbol{V}|} V_i}{|\boldsymbol{V}|}\right) = \\ g(x, A = -12.0, \mu = 1.0, C = -10.0, \sigma = -0.05) \\ + g(x, A = -12.0, \mu = 0.83, C = 0.0, \sigma = 0.01) \\ + g(x, A = -12.0, \mu = 1.16, C = 0.0, \sigma = 0.01) \end{aligned} \tag{3}$$

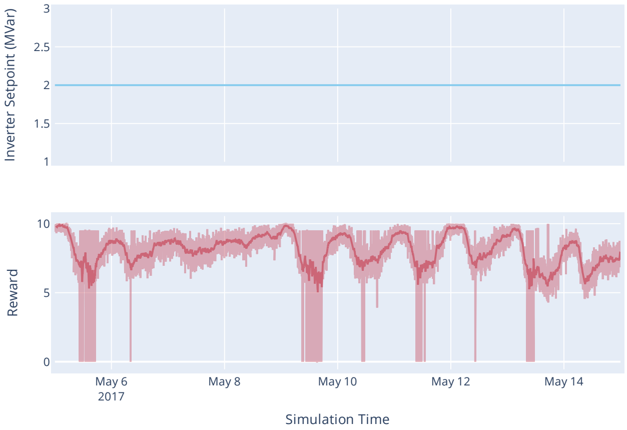

Figure 4. Setpoint and reward of the defender agent

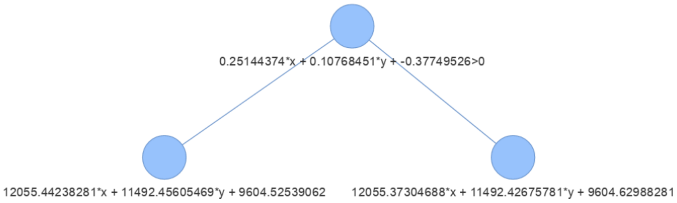
0.25144374*x + 0.10768451*y + -0.37749526>0
12055.44238281*x + 11492.45605469*y + 9604.52539062
12055.37304688*x + 11492.42675781*y + 9604.62988281

Figure 5. Decision tree as an equivalent representation of the agent's q-control policy

$$reward_{defender}\left(x = \frac{\sum_{i=1}^{|\boldsymbol{V}|} V_i}{|\boldsymbol{V}|}\right) = g(x, A = 10.0, \mu = 1.0, C = 0.0, \sigma = 0.032) \quad (4)$$

From Figure 4, we can see the setpoints and rewards of the defender agent. From these values alone, we can deduce that they have learned a very simple strategy (namely, one setpoint). This is expected in the simple scenario. We provided both agents with a larger-than-necessary policy network (a FF-DNN with $[2, 8, 8, 1]$ neurons).

IV. Discussion

Even if the number of neurons in the policy network of the agents seems low compared to many Deep Neural Networks (DNNs), such a network would already suffer from co-adaptation. However, Figure 5 shows that the resulting DT contains only the single setpoint strategy over the range of perceived voltage levels. Moreover, when calculating the invariants of the DT and, thus, compressing it even further, it collapses to one node that exactly represents the simple learning strategy.

We can conclude that our NN2EQCDT algorithm is able to extract a reasonable representation even if the policy network is larger than needed. This is especially important considering that, as seen in Figure 1, the policy network is evolved through neuroevolution. We cannot assume that it is always of an appropriate minimal size, since the neuroevolutionary algorithm is not automatically fed size constraints based on the agent's memory.

All data of this experiment is available from [25].

V. Conclusion and Future Work

In this work-in-progress paper, we presented preliminary results of our approach to explain learned strategies of agents in CNIs, which have been obtained in a autocurriculum setting.

In the future, we will expand our approach to more complex scenarios and a comprehensive experimentation regimen in order to show benefits and boundaries of our approach, especially focusing on scalability. We will present an extensive standard benchmarking scenario for our ARL methodology that will be based on a simulated power grid that includes a wide range of Distributed Energy Resources (DERs), consumers/prosumers, and assets the grid operator has access to. We will then show the benefits of the autocurriculum and, especially, our extended ARL agent architecture [24]. Through the steps outlined in this work-in-progress paper, as well our previous publications, we work towards making introspection of learned strategies in CNIs a default.

Acknowledgements

This work was funded by the German Federal Ministry for Education and Research (BMBF) under Grant No. 01IS22071.

References

[1] E. M. Veith, *Universal Smart Grid Agent for Distributed Power Generation Management*. Logos Verlag Berlin GmbH, 2017.

[2] E. Frost, E. M. Veith, and L. Fischer, "Robust and deterministic scheduling of power grid actors," in *7th International Conference on Control, Decision and Information Technologies (CoDIT)*, IEEE, 2020, pp. 100–105.

[3] M. Sonnenschein and C. Hinrichs, "A distributed combinatorial optimisation heuristic for the scheduling of energy resources represented by self-interested agents," *International Journal of Bio-Inspired Computation*, pp. 69–78, 2017, ISSN: 1758-0366. DOI: 10.1504/IJBIC.2017.10004322.

[4] O. P. Mahela *et al.*, "Comprehensive overview of multi-agent systems for controlling smart grids," *CSEE Journal of Power and Energy Systems*, vol. 8, no. 1, pp. 115–131, Jan. 2022, Conference Name: CSEE Journal of Power and Energy Systems, ISSN: 2096-0042. DOI: 10.17775/CSEEJPES.2020.03390.

[5] S. Holly *et al.*, "Flexibility management and provision of balancing services with battery-electricautomated guided vehicles in the Hamburg container terminal Altenwerder," ser. Energy Informatics, SpringerOpen, 2020, pp. 1–20. DOI: https://doi.org/10.1186/s42162-020-00129-1.

[6] J. Styczynski and N. Beach-Westmoreland, "When the lights went out: Ukraine cybersecurity threat briefing," *Booz Allen Hamilton*, vol. 12, pp. 1–86, 2016.

[7] A. Aflaki, M. Gitizadeh, R. Razavi-Far, V. Palade, and A. A. Ghasemi, "A hybrid framework for detecting and eliminating cyber-attacks in power grids," *Energies*, vol. 14, no. 18, p. 5823, Jan. 2021, ISSN: 1996-1073. DOI: 10.3390/en14185823.

[8] T. Wolgast, E. M. Veith, and A. Nieße, "Towards reinforcement learning for vulnerability analysis in power-economic systems," in *DACH+ Energy Informatics 2021: The 10th DACH+ Conference on Energy Informatics*, Freiburg, Germany, Sep. 2021, pp. 1–20.

[9] E. M. Veith, N. Wenninghoff, S. Balduin, T. Wolgast, and S. Lehnhoff, *Learning to attack powergrids with DERs*, 2022. DOI: 10.48550/ARXIV.2204.11352. [Online]. Available: https://arxiv.org/abs/2204.11352.

[10] O. Hanseth and C. Ciborra, *Risk, complexity and ICT*. Cheltenham, UK: Edward Elgar Publishing, 2007.

[11] R. Diao *et al.*, "Autonomous voltage control for grid operation using deep reinforcement learning," in *2019 IEEE Power & Energy Society General Meeting (PESGM)*, Atlanta, GA, USA: IEEE, Aug. 2019, pp. 1–5, ISBN: 978-1-72811-981-6. DOI: 10.1109/PESGM40551.2019.8973924.

[12] L. Fischer, J. M. Memmen, E. M. Veith, and M. Tröschel, "Adversarial resilience learning—towards systemic vulnerability analysis for large and complex systems," in *ENERGY 2019, The Ninth International Conference on Smart Grids, Green Communications and IT Energy-aware Technologies*, Athens, Greece: IARIA XPS Press, 2019, pp. 24–32, ISBN: 978-1-61208-713-9.

[13] E. Veith, L. Fischer, M. Tröschel, and A. Nieße, "Analyzing cyber-physical systems from the perspective of artificial intelligence," in *Proceedings of the 2019 International Conference on Artificial Intelligence, Robotics and Control*, ACM, Dec. 2019, ISBN: 978-1-4503-7671-6.

[14] Y. Zheng *et al.*, "Vulnerability assessment of deep reinforcement learning models for power system topology optimization," *IEEE Transactions on Smart Grid*, vol. 12, no. 4, pp. 3613–3623, 2021. DOI: 10.1109/TSG.2021.3062700.

[15] T. T. Nguyen and V. J. Reddi, "Deep reinforcement learning for cyber security," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–17, 2021. DOI: 10.1109/TNNLS.2021.3121870.

[16] C. Roberts *et al.*, "Deep reinforcement learning for der cyber-attack mitigation," in *2020 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, 2020, pp. 1–7. DOI: 10.1109/SmartGridComm47815.2020.9302997.

[17] J. Schrittwieser *et al.*, "Mastering Atari, Go, Chess and Shogi by planning with a learned model," pp. 1–21, 2019. arXiv: 1911.08265. [Online]. Available: http://arxiv.org/abs/1911.08265.

[18] S. Fujimoto, H. van Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," *arXiv:1802.09477 [cs, stat]*, Oct. 22, 2018. arXiv: 1802.09477. [Online]. Available: http://arxiv.org/abs/1802.09477 (visited on 08/07/2023).

[19] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," Jul. 19, 2017. arXiv: 1707.06347. [Online]. Available: http://arxiv.org/abs/1707.06347.

[20] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," *arXiv:1801.01290 [cs, stat]*, Aug. 8, 2018. arXiv: 1801.01290. [Online]. Available: http://arxiv.org/abs/1801.01290 (visited on 08/07/2023).

[21] E. Puiutta and E. M. S. P. Veith, "Explainable reinforcement learning: A survey," in *Machine Learning and Knowledge Extraction. CD-MAKE 2020*, Dublin, Ireland: Springer, Cham, 2020, pp. 77–95. DOI: 10.1007/978-3-030-57321-8_5.

[22] C. Aytekin, *Neural networks are decision trees*, Oct. 25, 2022. DOI: 10.48550/arXiv.2210.05189. arXiv: 2210.05189[cs]. [Online]. Available: http://arxiv.org/abs/2210.05189 (visited on 08/07/2023).

[23] T. Logemann and E. M. Veith, "NN2EQCDT: Quivalent transformation of feed-forward neural networks as DRL policies into compressed decision trees," in *Proceedings of the Fifteenth International Conference on Advanced Cognitive Technologies and Applications (COGNITIVE 2023)*, IARIA, IARIA XPS Press, Jun. 2023, pp. 94–100.

[24] E. M. Veith, "An architecture for reliable learning agents in power grids," in *Proceedings of the Thirteenth International Conference on Smart Grids, Green Communications and IT Energy-aware Technologies (ENERGY 2023)*, IARIA, IARIA XPS Press, Mar. 2023, pp. 13–16, ISBN: 978-1-68558-054-4.

[25] T. Logemann and E. Veith, *Towards explainable attacker-defender autocurricula in critical infrastructures: Source code to the paper*, Retrieved: 2023-09-18, 2023. [Online]. Available: https://gitlab.com/arl-experiments/simple-voltage-attack-explainability.

ISBN: 978-1-68558-113-8