Federal Office
for Information Security

Reinforcement Learning Security in a Nutshell

Document history

Version	*Date*	*Editor*	*Description*
1.0	22.08.23	Dr. Thomas Jung	

Federal Office for Information Security
P.O. Box 20 03 63
53133 Bonn
ki-kontakt@bsi.bund.de
Internet: https://www.bsi.bund.de
© Federal Office for Information Security 2023

Table of Contents

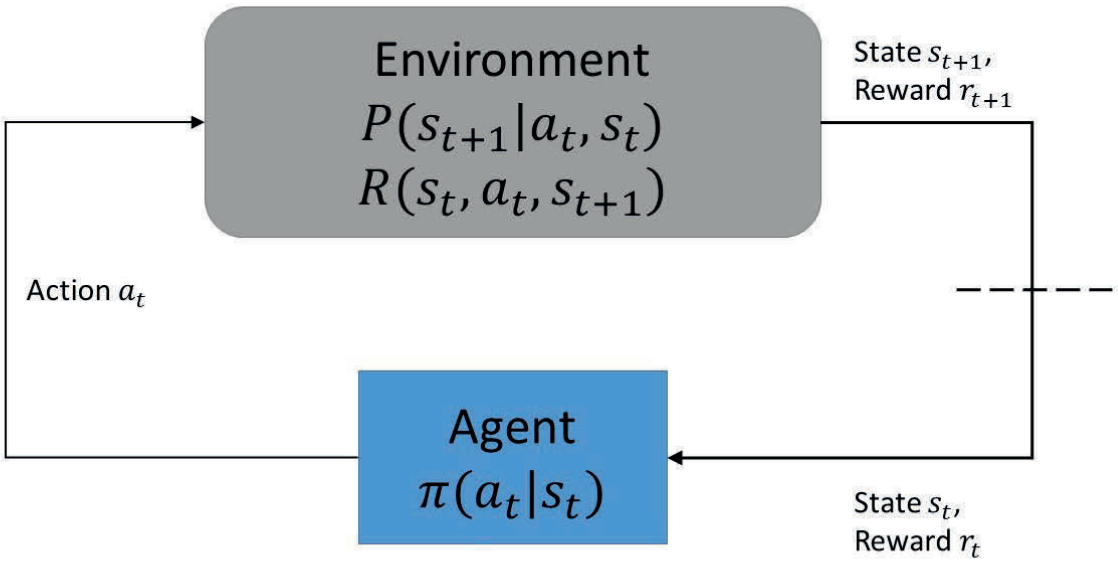
1 Introduction 4
1.1 Concept of Reinforcement Learning 4
1.2 Outline 5
2 Importance of Classical IT Security 6
3 Reward Minimization Attacks 7
3.1 Robustness 7
3.2 Attacks at Training Time 7
3.3 Attacks at Deployment 8
3.4 Defences 8
4 Policy Injection 9
4.1 Attacks at Training Time 9
4.2 Attacks at Deployment 9
4.3 Defences 10
5 Privacy Attacks 11
5.1 Attacks at Deployment 11
5.2 Defences 11
6 Conclusion 13
Bibliography 14

1 Introduction

In this whitepaper, we give a compact overview of possible attacks on Reinforcement Learning systems as well as corresponding defences. The document is targeted at developers of artificial intelligence systems and experts assessing the security of such systems, especially those focused on Reinforcement Learning. Its goal is to sensitize for possible attack vectors and to present possible defences, without going into too much technical detail. The document can be used as a substantive basis for a risk analysis.

1.1 Concept of Reinforcement Learning

Reinforcement Learning is, like Supervised and Unsupervised Learning, a popular technique in Machine Learning. It can be interpreted as a sort of feedback loop between an acting agent and its environment. This is typically modeled as a fully or partially observable Markov Decision Process. Such a process is defined by a state space S containing all possible states of the system, an action space A containing every action, a transition function P, which can be probabilistic and describes how a state changes under a given action, and finally a reward function R, defining the rewards for state-action pairs. In this document, we will sometimes denote the agent as the model, while system means the complete feedback loop.
In image 1, this feedback loop is schematically pictured, with the transition function P, the reward function R and π denominating the agent's policy (the dashed line denotes a jump in time, see index of the states and rewards). The goal of Reinforcement Learning is to find a policy, that performs optimally, or close to optimally, in this environment. This means maximizing the return, which is usually a weighted sum of all rewards $\sum_t \gamma_t r_t$, with rewards r and weights γ. Implied in this is the fact, that Reinforcement Learning learns strategies that have temporal cohesion, contrary to typical classifiers. Famous applications include chess engines, robotics or autonomous driving.



1: Reinforcement Learning Scheme

Security for Reinforcment Learning has, of course, an overlap with general AI security. Many concepts are similar and some of the general attacks also apply to Reinforcement Learning systems. While this document is intended to be self contained, for a more detailed and thourough understanding it is advisable to also read an overview on general AI specific attacks and defences, which is for example given in the detailed publication of BSI (BSI, 2022) or the more concise publication (BSI, 2023).

1.2 Outline

In the following, we categorize three different types of attacks, which are defined by the goal an attacker aims to reach.

- Attacks to minimize the return. These attacks aim to decrease the performance, which is measured by the return, and thereby the usefulness of the system.
- Policy injection attacks, in contrast, do not care as much about the return but rather aim to make the agent follow a specific policy. A relaxed form of this attack is given, when the attacker just aims for the system to reach a specific state.
- Attacks that try to extract information from the system. On the one hand, this can be critical data that is used in the training, like e.g. in healthcare applications. On the other hand, the models themselves might get copied, which can hold economical value.

For each of these attack types, we present possible defences.

Note, that there is no guarantee that the presented defences work against every possible attack and are suited for every possible situation. The document merely provides a broad overview about some of the possibilities, and special care has to be taken when applying these suggestions in a real world application. The document is intended as a basis to start thinking about security risks of Reinforcement Learning and mitigate them. It is important to stress, that the application of defences might have negative consequences on the performance of the agent and hence careful trade offs have to be made.

To get a deeper understanding of this important topic, we advise the reader to read further into the relevant literature. Surveys about security in Reinforcement Learning, such as for example (Demontis, et al., 2022) or (Lei, et al., 2023), serve as an additional starting point and give more technical details on certain topics. Note, that this is an actively researched field, and new findings are frequent.

2 Importance of Classical IT Security

Reinforcement Learning has some unique attack angles that we will present in this guide. However, Reinforcement Learning agents are by design always embedded into a larger IT system. At the very least, it possesses an interface with the environment. Typically, a Reinforcement Learning system includes

- the action interface between the agent and the environment, i.e. the output of the agent is sent as the input for the environment,
- the state interface , i.e. an output of the environment is sent as an input for the agent, and
- the reward interface, also an output of the environment that is sent as an input for the agent.

Because of this, a holistic security approach is essential, and classical IT security is an important factor in the security of AI systems that can provide an additional layer of defence.

Especially important is the security of the aforementioned interfaces, which needs to be taken into account when analysing the overall security of a Reinforcement Learning system. This can prevent e.g. man-in-the-middle attacks by denying access to unauthorized users. Changes in the in- and output of the AI agent can be fatal, so firewalls, strict user management and physical access management can be an important part of the security measures.
These measures are also useful techniques against privacy attacks. A sophisticated defence against attacks to extract the model by making queries is useless, if an attacker can get access to the model data and copy them from the hardware. Good documentation and the logging of system operations to check for anomalies in usage or system behaviour are useful security measure, too. An emergency plan in case of an attack or corruption of data should be in place.

In addition, public models can contain backdoors or other intrinsic security risks, so one needs to be cautious when using pretrained models. This also applies for training environments or reward functions, which are used in training.

Overall, classical IT security is a wide field, adresses many topics and we will not go into detail here. In the IT-Grundschutz (BSI, 2022) there can be found in-depth recommendations by the BSI. Often, AI systems are embedded in a cloud environment, which then also makes measures of Cloud security important. Similar to classical IT security, the BSI provides guidelines for Cloud security (BSI, 2020), and even specifically tailored for AI (BSI, 2021).

3 Reward Minimization Attacks

The first class of attacks we want to look at are attacks that aim at causing a poor agent performance. Thus, they may in practice be considered as attacks on the availability of the system, as it is not usable when minimizing the return.

Some examples of such attacks are:

- A Reinforcement Learning agent is used to manage cybersecurity tools. During training, the reward is large when an intrusion is defended. By minimizing the rewards, an attacker can influence the agent such that the system allows intrusion.
- An autonomous robot is controlled by Reinforcement Learning. The reward is coupled to desirable behaviour, such as staying on the correct path or doing planned procedures. By changing the camera inputs of the robot, a fatal misbehaviour is provoked that can destroy the robot and damage the surroundings. This trajectory of states has, of course, an overall low return.

Closely related to this type of attack is the notion of robustness, which we will briefly explain in the following subsection. Afterwards, we provide an overview of the attacks separated by the life cycle phase they happen at, i.e. if the attacks occur during training phase or operation. Lastly, we outline possible defences. The subsequent chapters follow a similar structure.

3.1 Robustness

Robustness is a broad concept that occurs, for example, in control theory. It describes how well a system performs under unexpected circumstances. Classically, it is used to describe how a model reacts to general errors and measurement uncertainties. These are naturally occuring phenomena, but it is easy to see how this concept can also be useful in the context of cybersecurity. A system that is robust against natural phenomena is probably more robust against tampering from an attacker than a model that is not robust against these phenomena.

As we will see in section 3.4, most defence mechanisms against reward minimization aim to increase the robustness of the Reinforcement Learning model. However, a drawback of robustness is that it typically comes with a price. A model that produces an optimal return in training is usually not robust and increasing robustness often leads to suboptimal performances.

In the following, only an intuitive understanding of robustness is required. For a more detailed definition and a method to estimate robustness, which can be useful to find weaknesses of the model, we refer the reader to (Korkmaz, et al., 2023). The source focuses on Deep Reinforcement Learning, i.e. Reinforcement Learning that uses Deep Neural Networks, which are one of the most important types of Reinforcement Learning.

3.2 Attacks at Training Time

Attacks at training time focus on changing some value of the Markov Decision Process to sabotage the training. Due to the nature of the process, changing any value, e.g. the action or the communicated state, can cascade through the feedback loop and have a profound influence. Typical attack vectors are:

- Changing the reward function by direct access is the most obvious attack. For example, just changing the sign of the reward function typically leads to a considerably worse policy.
- Changing the actions or the observed states might also lead to a worse performance. In order to explicitly minimize the rewards, white box knowledge of the system is beneficial. However, even without detailed knowledge, attackers might achieve a decrease in return.
- If the attacker has access to the training environment, a change in the transition function can lead to similar effects as outlined before.

A potential method an attacker might use to implement the aforementioned changes involves an adversarial agent during training to sabotage the system.

3.3 Attacks at Deployment

Since training is sometimes performed in a secure or isolated setup, the attack surface during deployment might be larger.

In this phase, the attacker needs to overrule or trick an existing policy. This can be done, similar to the methods seen in the last subsection, by altering or influencing the actions, the observed states or the transition probabilities. The rewards do not play a role in the operation phase anymore, as they are only used to train the Reinforcement Learning model.

The attacks in this category are typically a type of adversarial attack. Adversarial attacks, generally speaking, aim to change the output of the AI agent by changing the input, see (BSI, 2023). Often, the change is unnoticeable to the human eye, e.g. in the form of a slight noise. As in other artificial intelligence models, gradient methods can often be used to find adversarial examples, see e.g. (Evasion attacks against machine learning at test time, ECML PKDD, 2013), (Adversarial Attacks on Neural Network Policies, 2017) or (Characterizing Attacks on Deep Reinforcement Learning, 2022).
In the case of white box knowledge, the attacker has direct access to the gradient of the agent. It is however often possible, even with black box knowledge only, to approximate the gradient, e.g. with the use of a so-called shadow model, that imitates the deployed model (Chen, et al., 2020).

3.4 Defences

The following defensive measures might be used against such attacks:

- Attack detection allows a user to fend off an attack, or at least prevent a model from doing harm by shutting it off. For this, the time continous nature of Reinforcement Learning can be used. As there is often temporal cohesion, the next state can, to some degree, be predicted from the history. This is especially true for any type of state that can be interpreted as ‚visual', e.g. provided by a camera. If the observed state is, in some appropriate sense, too far away from the expected one, an attack might be assumed. If an attack or unusual situation is detected in this way, one possible reaction is to switch the actual observation with the predicted state. Such an approach might increase the effort for an attacker, which is forced to alter the relevant history of the states. This is shown e.g. in (Tekgul, et al., 2021), (Xiong, et al., 2022)
- One of the most prevalent methods to defend against such attacks is to make the model more robust by adversarial training. For this purpose, the aforementioned attacks during training time can be intentionally deployed in a controlled manner, see e.g. (Robust Deep Reinforcement Learning with Adversarial Attacks, 2018)
- Besides adversarial training, methods of robust optimization can help to create more robust policies, as can be seen in e.g. (Wang, 2022). With this ansatz, instead of searching for the policy that creates the greatest return, one looks at the worst case under an attack and chooses the policy that performs best in that situation. Therefore, one can effectively cap the damage.

To effectively use the last two methods, one needs to define a suitable search range. That is, a maximal effort one expects the attacker to put into the attack, as well as a minimal deviation that is interesting. The importance of this is e.g. apparent in attacks on the states. There, one would search the whole state space for worst cases if no maximal effort is defined. If no minimal threshold is chosen, attacks are detected when the observed state is just barely different from the predicted state.

All of the described methods increase robustness but typically decrease optimal performance. An analysis of this trade off is necessary to find the acceptable amount of optimality with highest possible robustness.

4 Policy Injection

Policy injection attacks aim to implement a specific given policy. We also consider a weaker form of policy injection, where the attacker does not want to inject a completely new policy, but rather just aims for the system to reach a specific state.

Policy injections differ from the attacks in the previous section, because they do not necessarily imply minimization of return. One could argue that reward minimization is a form of policy injection, implementing the policy for reward minimization. However, since the used attacks tend to differ, we separate the two attack classes.

Examples for this type of attack are:

- An autonomous driving system that is based on Reinforcement Learning is working correctly in most cases, and hence no problem is noticed. Then in deployment, the agent is presented with a traffic sign that has a sticker on it. This sticker was malevolently introduced in the training as a trigger, and the agent reacts to it by leaving the road and crashing the car.
- A Reinforcement Learning agent is set up to manage a production line. By altering the states, an attacker changes the policy. The policy still reaches the goal, but the costs are increased, maybe by higher energy consumption.

4.1 Attacks at Training Time

In order to make the agent follow a specific policy at operation time, a backdoor approach can be taken: In the training, the attacker tries to place a trigger, which the agent learns. If the agent is presented with the trigger during operation, it will follow the specified policy.

- One way to implement such a backdoor is via Reward Hacking, i.e. changing the reward function in the training phase. In contrast to reward minimization, a deeper knowledge of the state and action space is necessary to create a specific policy. If the attacker only aims at reaching a specific state, raising the reward for actions reaching this state can already be enough.
- One can also force the agent to learn a specific policy by changing state observations, actions or transition probabilities. This however requires a deep understanding of the dependencies between these values and their interdependencies. This is in general only possible with full white box knowledge of the system. Therefore, these attacks are not as easy to carry out as backdoor attacks with Reward Hacking.

4.2 Attacks at Deployment

What makes attacks at deployment difficult is the existence of an already established policy that needs to be, in some way, overruled.

- If the attacker can set the actions, the injection of a policy is trivial. Therefore, this needs to be prevented by classical IT security measures, like access control.
- If the attacker can only slightly change the actions, but not set them freely, a policy injection attack is still possible. For this, white box knowledge, in particular about how the action is influencing the environment and thereby the states and through that the agent, is necessary. This is shown in (Provably Efficient Black-Box Action Poisoning Attacks Against Reinforcement Learning, 2021).
- The same holds, if the attacker is able to change the state observations or the transition probabilities. To get knowledge about the deployed policy, the ansatz of a shadow model can also be used in this sort of attacks.

If we look at the somewhat weaker attack that aims to reach at a certain state, then the attacker can get by with black box knowledge more often. If there is some sort of predictability in the system, or if the attacker can change the transition probabilities, the system might be tricked into reaching the goal state of the

attacker. This type of attack is sometimes called ‚Enchanting Attack' (Tactics of Adversarial Attack on Deep Reinforcement Learning Agents, 2017).

4.3 Defences

The defence mechanisms we saw in the previous section increasing the robustness of the system can also help against the presented policy injection attacks during deployment.

Against backdoor attacks, we will present some more specific defences here.

- One possibility is to use a projection method. There, the basic assumption is that the triggers, which are special states, are usually placed outside of common states, so they are not detected easily. If that is the case, one can create a subspace from a clean collection of common states, that by assumption do not contain a trigger, and project observations on this subspace, eliminitating the trigger in the process (Provable Defense against Backdoor Policies in Reinforcement Learning, NeurIPS, 2022). This also works with high-dimensional state spaces, but the sample complexity grows.
- There have also been methods proposed, that try to detect the triggers. In (Backdoor Detection and Mitigation in Competitive Reinforcement Learning, 2023), an agent is trained to look for states that lead to a drop in the return of the Reinforcement Learning system. If such states are detected, a backdoor is assumed. Then, an unlearning step might be possible to sanitize the system. This uses the observation that often states near the trigger also lead to drops in the return. Alternatively, if this is not possible or if one is unsure if every backdoor is detected, the system can be replaced with a new one, trained on clean data.

5 Privacy Attacks

Privacy attacks are very different in nature to the previously described attacks. They do not disturb the functionality of the system, but rather want to extract some sort of information from the Reinforcement Learning system. This can be e.g. information about the reward function, about the training environment or the policy itself. Often, this information is critical, especially protected or of economical interest. It is therefore important to protect the system from leaking information to others. Examples for information leaks are:

- A Reinforcement Learning agent is used to assist in medical diagnosis. It was trained on data from real patients. This data is highly sensitive, and an attacker could try to reconstruct patient information through querying the system.
- The navigation of a robot is done via Reinforcement Learning. It was trained in a special parcour, which was built after extensive and costly testing. An attacker might extract a map of this training ground from the behaviour of the robot.
- An agent was trained to perform on a very high level, so it is the best on the market. A competitor queries the model and by imitating it gets a good model with low effort.

5.1 Attacks at Deployment

We only present attacks during deployment, as during training time, classical IT security is most relevant, and there are no Reinforcement Learning specific attacks during training we know of. At deployment phase, the system has gathered all the information during training and has become vulnerable to extraction.

- In the previous sections, a possible use of a stolen policy has been shown. There, a shadow model that imitates the true model has been used to get gradients or other information in a black box setting. A way to get this shadow model is via imitation learning (Chen, et al., 2020). Here, the attacker trains a model on queries or observations of the target system, with the goal to act the same. This works better, if the basic architecture of the target is known, but can also be done with black box knowledge.
- If the attacker has knowledge about the reward function and the state and action space, the reconstruction of the transition probabilities is possible. This means, it is possible to extract information about the training environment. In (How You Act Tells a Lot: Privacy-Leaking Attack on Deep Reinforcement Learning, AAMAS, 2019) an actual map of the training room a robot was trained in was reconstructed.
- Attacks on the reward function can also leak important information and should not be ignored. Even when the policy is trained in a privacy-preserving manner, the reward function might still be vulnerable. (Prakash, et al., 2021) show, how this can be done with an inverse Reinforcement Learning approach. There, based on the optimal policy, the reward function is reconstructed with linear programming, i.e. one finds what reward function leads most likely to the policy at hand. Note that inverse Reinforcement Learning presents a whole class of methods, and other methods in addition to linear programming exist.

5.2 Defences

The general concept of the typical defence mechanisms discussed in literature is built on Differential Privacy (Differential Privacy, ICALP, 2006). Formulaic, for a stochastic mechanism M and any two training sets U, U‘ that only differ in one data point, this is written as

$$P[M(U)] \leq e^{\epsilon} P[M(U')] + \delta$$

for some non-negative parameters ϵ, δ. P denotes here the probabilities of different results of the mechanism, not the transition function. For smaller ϵ and δ it gets harder to tell what dataset the mechanism M was applied to.

Differential Privacy is typically implemented by adding stochastic noise to different values. These values can include:

- network parameters,
- rewards,
- value functions,
- loss function,
- training observations or
- transition probabilities.

In order to apply the noise at every time step, a suitable stochastic process with fitting parameters must be chosen. Adding the noise makes it unclear, if information one extracted is a real datapoint or one added through noise, thereby protecting the real data.

The drawback of this approach is, that it adds a computational overhead and can slow down the training and can make it computationally harder. Furthermore, if unique datapoints are important, this approach can lower the quality of the agent. Moreover, finetuning of hyperparameters, in particular the size of noise, can be challenging.

Of course, classical IT security is especially important for privacy attacks. If an attacker is able to gain access to the model data or the training environment, no extraction algorithm is necessary and the attacker can directly pull the information from the system.

6 Conclusion

Reinforcement Learning has some specific vulnerabilities that we presented here. We categorized them into three attack classes and presented possible defences. Note, that we do not claim to be comprehensive.
All defences can help mitigate attacks, but they do not guarantee complete security. Furthermore, they do affect the performance of the model. The defence mechanisms may lead to suboptimal policies, a trade off users should be aware of. Effects like that should be carefully assessed in a risk reward analysis. That the performance can decrease when using defence mechanisms becomes apparent, when we see that some of the defences can also be used as intentionally placed attacks during training.
Additionally, the defence strategies often introduce additional computation time and effort, which should be included in the considerations.
As some applications for Reinforcement Learning are critical, attacks can present substantial dangers to the privacy of protected data as well as to the integrity and availability in crucial sectors, e.g. autonomous driving or health applications. Hence, defence mechanisms should often not be neglected, and the drawbacks can be well worth.

Bibliography

Adversarial Attacks on Neural Network Policies. **Huang, Sandy H., et al. 2017.** Toulon : s.n., 2017.

Backdoor Detection and Mitigation in Competitive Reinforcement Learning. **Guo, Junfeng, Li, Ang und Liu, Cong. 2023.** s.l. : arXiv, 2023.

BSI. 2021. *AI Cloud Service Compliance Criteria Catalogue.* 2021.

—**. 2023.** AI Security Concerns in a Nutshell. [Online] 2023. https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/KI/Practical_AI-Security_Guide_2023.html.

—**. 2020.** *Cloud Computing Compliance Criteria Catalogue - C5:2020.* Bonn : s.n., 2020.

—**. 2022.** *IT-Grundschutz-Kompendium.* Bonn : s.n., 2022.

—**. 2022.** Projekt P464. [Online] 2022. https://www.bsi.bund.de/DE/Service-Navi/Publikationen/Studien/Projekt_P464/Projekt_P464_node.html.

Characterizing Attacks on Deep Reinforcement Learning. **Pan, Xinlei, et al. 2022.** Auckland, New Zealand : s.n., 2022.

Chen, Kangjie, et al. 2020. Stealing Deep Reinforcement Learning Models for Fun and Profit. *CoRR.* 2020.

—**. 2020.** Stealing Deep Reinforcement Learning Models for Fun and Profit. *CoRR.* 2020.

Chen, Tong, et al. 2019. Adversarial attack and defense in reinforcement learning-from AI security view. *Cybersecurity.* Volume 2, 2019, Bd. 1.

Demontis, Ambra, et al. 2022. A Survey on Reinforcement Learning Security with Application to Autonomous Driving. *CoRR.* 2022.

Differential Privacy, ICALP. **Dwork, Cynthia. 2006.** s.l. : Springer, 2006.

Evasion attacks against machine learning at test time, ECML PKDD. **Biggio, Battista, et al. 2013.** s.l. : Spinger, 2013.

Explaining and Harnessing Adversarial Examples, ICLR. **Goodfellow, Ian, Shlens, Jonathon und Szegedy, Christian. 2015.** San Diego : arXiv, 2015.

How You Act Tells a Lot: Privacy-Leaking Attack on Deep Reinforcement Learning, AAMAS. **Pan, Xinlei, et al. 2019.** Montreal : s.n., 2019.

Korkmaz, Ezgi und Brown-Cohen, Jonah. 2023. Detecting Adversarial Directions in Deep Reinforcement Learning to Make Robust Decisions. *CoRR.* 2023.

Lei, Yunjiao, et al. 2023. New Challenges in Reinforcement Learning: A Survey of Security and Privacy. *CoRR.* 2023.

Nachname, Vorname. 2019. *Titel eingetragen.* Bonn als Ort : Verleger eingetragen, 2019. ISBN 123456.

Prakash, Kritika, et al. 2021. How Private Is Your RL Policy? An Inverse RL Based Analysis Framework. *CoRR.* 2021.

Provable Defense against Backdoor Policies in Reinforcement Learning, NeurIPS. **Bharti, Shubham Kumar, et al. 2022.** 2022.

Provably Efficient Black-Box Action Poisoning Attacks Against Reinforcement Learning. **Liu, Guanlin und Lai, Lifeng. 2021.** Virtual-Only : s.n., 2021.

Robust Deep Reinforcement Learning with Adversarial Attacks. **Pattanaik, Anay, et al. 2018.** Stockholm : s.n., 2018.

Tactics of Adversarial Attack on Deep Reinforcement Learning Agents. **Lin, Yen-Chen, et al. 2017.** Toulon, France : s.n., 2017.

Tekgul, Buse G. A., et al. 2021. Real-time Attacks Against Deep Reinforcement Learning Policies. *CoRR.* 2021.

Wang, Chao. 2022. Attacking and Defending Deep Reinforcement Learning Policies. *CoRR.* 2022.

Xiong, Zikang, et al. 2022. Defending Observation Attacks in Deep Reinforcement Learning via Detection and Denoising. *CoRR.* 2022.