Multi-Environment Training Against Reward Poisoning Attacks on Deep Reinforcement Learning

Myria Bouhaddi and Kamel Adi
Computer Security Research Laboratory, University of Quebec in Outaouais, Gatineau, Quebec, Canada

Keywords: Deep Reinforcement Learning, Adversarial Attacks, Reward Poisoning Attacks, Optimal Defense Policy, Multi-Environment Training.

Abstract: Our research tackles the critical challenge of defending against poisoning attacks in deep reinforcement learning, which have significant cybersecurity implications. These attacks involve subtle manipulation of rewards, leading the attacker's policy to appear optimal under the poisoned rewards, thus compromising the integrity and reliability of such systems. Our goal is to develop robust agents resistant to manipulations. We propose an optimization framework with a multi-environment setting, which enhances resilience and generalization. By exposing agents to diverse environments, we mitigate the impact of poisoning attacks. Additionally, we employ a variance-based method to detect reward manipulation effectively. Leveraging this information, our optimization framework derives a defense policy that fortifies agents against attacks, bolstering their resistance to reward manipulation.

1 INTRODUCTION

Reinforcement Learning (RL) has garnered significant attention in recent years due to its remarkable ability to solve complex decision-making problems through continuous agent-environment interaction, leading to the development of optimal action selection policies (Sutton and Barto, 2018). Deep Reinforcement Learning (DRL), an amalgamation of reinforcement learning and deep learning, has emerged as a powerful tool for handling high-dimensional state spaces and complex task selection policies. Several DRL algorithms, including Deep Q-Networks (DQN) (Mnih et al., 2015), Trust Region Policy Optimization (TRPO) (Schulman et al., 2015), and Asynchronous Advantage Actor-Critic (A3C) (Greydanus et al., 2018), have been developed to efficiently tackle challenging real-world problems.

DRL has made significant contributions in diverse fields, including robotics, healthcare, and finance. In robotics, DRL enables the development of autonomous robots capable of learning tasks like grasping, walking, and manipulation. In healthcare, it optimizes treatment plans for patients with chronic conditions by leveraging patient data, improving treatment outcomes. In finance, DRL aids in designing automated trading systems that make intelligent real-time decisions based on market data. These examples highlight the wide-ranging applicability of DRL in addressing complex real-world problems.

However, the security of DRL systems has become a critical concern, as they are vulnerable to adversarial attacks (Kiran et al., 2021; Behzadan and Munir, 2017a). Even small perturbations can significantly impact performance (Zhang et al., 2021b), and attacks on one policy can be transferred to others (Huang et al., 2017). Poisoning attacks, specifically, manipulate reward signals during the learning process, thereby influencing the behavior of the agent. Compromised DRL systems pose risks such as economic losses, injuries, and even potential loss of life, especially in critical domains like autonomous cars and drones.

While security challenges in supervised and unsupervised learning have been extensively studied (Akhtar and Mian, 2018), the security implications of DRL demand significant attention. Ensuring robustness against attacks is crucial for the safe deployment of DRL systems in critical applications. Addressing these security challenges is paramount for successful real-world implementation.

This paper focuses on the problem of reward poisoning in DRL, where attackers alter rewards to manipulate the agent's policy, necessitating the development of defense mechanisms. We propose a robust RL algorithm that can detect and defend

Bouhaddi, M. and Adi, K.
Multi-Environment Training Against Reward Poisoning Attacks on Deep Reinforcement Learning.
DOI: 10.5220/0012139900003555
In *Proceedings of the 20th International Conference on Security and Cryptography (SECRYPT 2023)*, pages 870-875
ISBN: 978-989-758-666-8; ISSN: 2184-7711

against reward tampering. Our approach involves training agents in diverse environments to minimize the impact of poisoning, employing variance-based techniques for detection, and enhancing resilience through adversarial training. Our main contributions include a rigorous formulation of the poisoning attack as an optimization problem, providing insights into the attacker's objectives and enabling exploration of strategies for detection and mitigation. Additionally, we propose a novel approach that mitigates reward manipulation attacks, leveraging multiple environments, variance-based detection, and adversarial training. Our experimental results demonstrate the effectiveness of our approach in enhancing the robustness of agent-based systems against adversarial attacks.

2 RELATED WORK

Attacks Against Reinforcement Learning. Reinforcement learning (RL) is susceptible to various types of attacks, with evasion attacks and model poisoning attacks being two prominent categories extensively studied in deep RL (Huang et al., 2017; Kos and Song, 2017; Lin et al., 2017). Evasion attacks aim to induce undesirable behavior in trained policies by finding adversarial examples, while model poisoning attacks manipulate the reward signal during RL training to induce sub-optimal policies. These attacks have significant implications in real-world applications, including the manipulation of pre-trained RL models downloaded by agents.

Previous research has investigated reward poisoning in both batch and online RL settings. In batch RL, attackers can easily modify pre-collected rewards, while online RL poses a greater challenge as rewards need to be modified on-the-fly. Although reward poisoning in online RL has been studied using multi-armed bandits, our focus is on black-box attacks that can target any efficient RL algorithm.

Furthermore, studies have explored reward poisoning in the white-box setting, where attackers have complete knowledge of the underlying Markov decision process (MDP) or learning algorithm. These attacks involve manipulating the reward function using adversarial rewards based on the state and action, independent of the learning process. Notably, (Zhang et al., 2020) developed an adaptive attack that leverages the victim's Q-table, significantly accelerating the attack process.

In contrast to observation perturbation attacks that alter the agent's environment observation during training without changing the actual state or reward (Behzadan and Munir, 2017b; Inkawhich et al., 2019), our poisoning attacks directly modify the actual reward or state of the environment. This differentiation highlights the distinct nature and potential impact of reward manipulation attacks in RL.

Defenses Against Poisoning Attacks. In order to ensure the security of DRL policy training, defense mechanisms are employed to protect against poisoning attacks. The importance of robustness cannot be overstated, as it guarantees the functionality of the system even in the presence of disturbances (Behzadan and Munir, 2017b). Defenses against poisoning attacks can generally be classified into two categories: (1) studies that provide theoretical guarantees for learning under perturbations (Banihashem et al., 2021; Lykouris et al., 2021; Chen et al., 2021; Wei et al., 2022; Zhang et al., 2021a; Wu et al., 2022), and (2) empirical approaches that evaluate the robustness of the system through practical experiments (Behzadan and Munir, 2017b; Behzadan and Munir, 2018; Wang et al., 2020).

However, it is important to note that designing robust DRL algorithms often comes at a cost, as it may compromise the overall performance of the learned policies. Achieving complete robustness is challenging, especially considering the evolving strategies employed by attackers. Hence, relying solely on robustness measures may prove inadequate in ensuring the secure learning of DRL policies. It is essential to explore additional measures and techniques that can enhance the security and reliability of DRL systems against poisoning attacks.

In this context, our work focuses on addressing the issue of data poisoning in reinforcement learning, particularly the manipulation of reward signals to influence policy. We aim to propose innovative solutions that effectively protect DRL policies from such attacks. While robustness is crucial, we acknowledge its potential impact on policy performance. Therefore, we propose a lightweight approach that enhances protection without compromising the overall performance of the system. By complementing robustness measures, we aim to strengthen the security of DRL learning and enhance the reliability of the learned policies.

3 PRELIMINARY

In this section, we will outline the essentials of deep reinforcement learning, including its key components and underlying principles.

Deep Reinforcement Learning. In reinforcement learning (RL), an agent learns an optimal behavior by

sequentially interacting with an environment, known as a Markov Decision Process (MDP), to achieve its objectives through trial and error. The MDP is defined as a tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P}, R, \gamma, \sigma)$, where $\mathcal{S}$ and $\mathcal{A}$ are the state and action spaces, respectively, $\mathcal{P}$ is the transition dynamics that determine the probability distribution of the next state given the current state and action, R is the reward function that maps state-action pairs to scalar rewards, γ is a discount factor that weighs immediate and future rewards and σ is the initial distribution over the states. The training process consists of multiple episodes where each episode is initialized with a state sampled from σ. The agent interacts with the environment at each timestep until the episode ends. It is assumed that every episode is comprised of T distinct timesteps. We assume that $\mathcal{S}$ and $\mathcal{A}$ are finite and discrete sets.

The agent interacts with the environment sequentially, starting with an initial state s_0, following the distribution σ, and selecting actions based on a policy π. Policies can be generic (stochastic) denoted by $\pi(a|s)$, mapping states to action probabilities, or deterministic denoted by $\pi(s)$. The set of all policies is Π, and deterministic policies are Π^{det}.

The agent's transition to a new state s_{t+1} based on $\mathcal{P}$ and the reward $r_{(s_t,a_t)}$ it receives, reflecting the quality of its decision, leads to the generation of a trajectory $\mathbb{T}$ consisting of state-action-reward triplets. This trajectory captures the agent's interaction with the environment, and at each time step, the agent updates its Q-table, which stores the estimated values of state-action pairs.

In reinforcement learning, the cumulative reward or return is the total reward an agent receives over time. It is computed as the sum of discounted rewards at each timestep, using the factor $\gamma \in [0,1]$. This balances the importance of immediate and future rewards, expressed as $CR = \sum_{t=0}^{T} \gamma^t R(s_t, a_t)$.

We define the state value $\mathcal{V}_\pi(s)$ for a policy π as the expected total return CR from state s under policy π. It is represented by the function $\mathcal{V}_\pi : \mathcal{S} \to \mathbb{R}$ and expressed as $\mathcal{V}_\pi(s) = \mathbb{E}[CR|s = s_t]$, accounting for stochastic environment transitions.

The state-action value function $Q_\pi(s,a)$, also known as the Q-function, extends the definition of the state-value function $\mathcal{V}_\pi(s)$ to state-action pairs. It represents the expected return CR from state s, taking action a, and following policy π. The agent aims to find an optimal policy π^* that maximizes the expected return from all states, given by $\pi^* = \underset{\pi}{\operatorname{argmax}}\ Q_\pi(s,a)$.

The policy score ρ^π quantifies the overall quality of a policy π based on the expected rewards obtained by following the policy over an extended period. It is calculated by considering all possible actions from each state using the Q-values. The score is expressed as $\rho^\pi = \mathbb{E}[(1-\gamma)\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)|\pi, \sigma]$, where the initial state s_0 is sampled from the initial state distribution σ, and subsequent states s_t are obtained by executing policy π in the MDP. The score reflects the expected total return, discounted by a factor of $1-\gamma$.

Deep Reinforcement Learning (DRL) combines deep learning and RL to tackle challenges in learning control policies from high-dimensional raw input data and large state and action spaces. The policy π in DRL is represented by a deep neural network with parameters Θ. Various DRL algorithms, including Deep Q-Network (DQN), Trust Region Policy Optimization (TRPO), and Asynchronous Advantage Actor-Critic (A3C), aim to optimize the policy network by maximizing the expected return.

In Deep Q-learning, the Q-values for actions are approximated based on states, enabling the agent to select the action with the highest Q-value to maximize its reward. This approach has shown success in domains like Go and Atari games.

The policy gradient algorithm directly parameterizes the policy as $\pi_\theta(s,a)$, which takes the state as input and outputs the corresponding action. By maximizing the expected total discounted rewards, represented by the objective function $J(\theta)$, the optimal parameters θ and policy are obtained. The gradient of this objective function can be expressed as the expected product of the gradient of the log of the policy network and the action-value function of the Markov Decision Process.

To address this, the policy gradient algorithm approximates the action-value function $Q_\omega(s,a)$ using a deep neural network Q_ω learned alongside the policy network. This allows the Policy Gradient Theorem to be applied, facilitating the computation of the policy gradient.

4 REWARD POISONING ATTACKS AGAINST DRL

The goals of attacks against machine learning models in agent-based reinforcement learning often involve manipulating the policies of the agents to align them with a specific target policy. This is accomplished by strategically modifying the agent's reward function. In our study, we adopt an attack formulation inspired by previous works such as (Ma et al., 2019) and (Rakhsha et al., 2020). By leveraging these existing approaches, we aim to develop a comprehensive understanding of attack strategies and their implications in the context of agent-based reinforcement learning.

4.1 Threat Model

Our paper tackles the problem of reward poisoning in a white-box setting, where the attacker has complete knowledge of the agent's Markov Decision Process (MDP) environment and Deep Q-Learning algorithm, except for their future randomness. This means that the attacker places themselves between the environment and the agent, allowing them to carry out white-box attacks with ease. The motivation behind considering a white-box attack scenario is to develop a defense mechanism that can effectively counter the most challenging and sophisticated attacks.By assuming maximum knowledge for the attacker, our goal is to create a defense approach that is resilient even in the worst-case scenario, where the attacker strategically manipulates reward signals to deceive and redirect the agent's policy (Bouhaddi et al., 2018).

At each time step t, the attacker observes the agent's policy network $\pi_\theta(s,a)$, the current state s_t, the agent's action a_t, the resulting new state s_{t+1}, and the received reward r_t. The attacker's profile is defined as $\xi = (\pi^\dagger, \upsilon, \Delta^t)$, with $\pi^\dagger$ representing the target policy, υ limiting reward manipulations within an episode of length L, and Δ^t constraining perturbations added to rewards. These limitations ensure attack effectiveness, avoid detection, and prevent excessive disruptions to the agent's learning process. By carefully controlling perturbations, the attacker can target specific rewards, achieving their objectives while maintaining a low profile.

The target policy is defined as a function from the state space to the action space, $\pi^\dagger : S \to 2^{|\mathcal{A}|}, \pi^\dagger(s) \subseteq \mathcal{A}$, which specifies the set of actions desired by the attacker at the state s. The attacker can focus on certain states more than others, as these states trigger particular actions desired by the attacker. Thus, $\mathcal{S}^\dagger$, the set of target states, can be defined as $\mathcal{S}^\dagger = \{s \in \mathcal{S} : \pi^\dagger(s) \neq \pi^*(s)\}$, with $\pi^*(s)$ the optimal policy of the agent, which represents the actions it would have chosen in the absence of reward perturbations. So, given a target state set $\mathcal{S}^\dagger \subseteq \mathcal{S}$, the target policy is denoted as:

$$\pi^\dagger_\theta(s) = \begin{cases} a^\dagger & \text{if } s \in \mathcal{S}^\dagger \\ \pi_{\theta_i}(s) & \text{otherwise.} \end{cases}$$

where $a^\dagger$ is the target action desired by the attacker, $\pi_{\theta_i}(s)$ is the victim's actual policy and $\pi^\dagger_\theta(s)$ the partial target policy which is more suitable for large-scale state spaces, either discrete or continuous, as compared to the complete target policy that defines desired actions in all states.

The attacker can introduce a perturbation $\delta_t \in \mathbb{R}$ to the reward associated with the current state-action pair $r(s_t, a_t)$. For simplicity, we use the notation r_t instead of $r_{(s_t,a_t)}$. As a result, the reward perceived by the agent at time step t is given by $r_t + \delta_t$. We assume that the attack is limited by the infinity norm, which is referred to as limited per-step perturbation. This means that $|\delta_t| \leq \Delta^t$ for any time step t. In other words, we consider two constraints regarding the added perturbation: it should not be too substantial nor too frequent.

The attacker's objective is to discover the best sequence of perturbations to incite the agent to adopt the target policy while reducing the number of rounds during which the agent becomes aware of the attack. This involves minimizing the agent's disagreement with the target policy, so as not to raise suspicions and maintain the success of the attack.

Therefore, the attacker's problem is modeled as the following optimization problem:

$$\min_{\delta_t} \quad d(r_t, r_t + \delta_t) \tag{1}$$

$$\text{s.t} \quad \rho^{\pi_\dagger} \geq \rho^{\pi}, \ \ \forall \pi \in \Pi^{det} \backslash \{\pi^\dagger\} \tag{2}$$

$$\delta_t \leq \Delta^t, \forall t \tag{3}$$

$$\sum_t 1[Q_t \notin Q^\dagger] \leq \upsilon \tag{4}$$

where, d computes the Euclidean distance between the true reward and the altered reward. Therefore, the attacker's problem boils down to solving this optimization problem to find the minimum perturbation that allows for the adoption of its target policy by the agent, while avoiding detection by limiting the number of interventions, the amount of perturbation added per time step, and bounding the number of times the agent deviates from $Q^\dagger$.

5 PROPOSED DEFENSE MECHANISM USING MULTI-ENVIRONMENT TRAINING

We propose a defense approach to mitigate poisoning attacks in deep reinforcement learning (DRL) settings. Our approach utilizes multi-environment training, where an agent interacts randomly with multiple environments, each with different transition probabilities. This enables the agent to learn a more robust policy that is less affected by perturbations in the reward signal introduced by the attacker.

Our approach offers significant advantages over existing methods. By exposing the agent to diverse environments, it enhances the agent's experience and leads to a more robust policy that can handle new

and unseen situations. Furthermore, our approach is computationally efficient, requiring minimal modifications to the existing RL training pipeline.

In our proposed defense approach, we use a generalized environment $\mathcal{G} = (\mathcal{E}, \mu)$, consisting of a set of multiple environments $\mathcal{E}$ and a distribution μ over these environments. Each environment $e_i \in \mathcal{E}$ is modeled as a Markov Decision Process (MDP). This allows us to evaluate the agent's performance across different environments sampled from $\mathcal{E}$ according to μ. Training the agent in multiple environments with different reward structures increases its robustness to reward poisoning attacks.

We adopt the multi-environment training technique, where the agent interacts with a randomly sampled environment from $\mathcal{E}$ according to the probability distribution μ. The attacker, aware of the training environment, may attempt to poison the rewards. However, they must balance their actions to avoid detection. This limits the attacker's ability to perturb all rewards in all environments, reducing the overall impact of the poisoning.

To compute the agent's policy, we introduce the weighted policy π_w, which combines the policies learned in each environment e_i based on their corresponding probabilities μ_i:

$$\pi_w = \sum_i \mu_i \cdot \pi_i \tag{5}$$

Similarly, we compute the weighted Q-values $Q_w(s,a)$ by combining the Q-values $Q_i(s,a)$ obtained in each environment:

$$Q_w(s,a) = \sum_i \mu_i \cdot Q_i(s,a) \tag{6}$$

Finally, the average policy $\pi_{avg}(s)$ is obtained by taking the argmax over the weighted Q-values:

$$\pi_{avg}(s) = \arg\max_a Q_w(s,a) \tag{7}$$

This computation allows the agent to select actions that are robust to reward poisoning attacks by considering the variability of rewards across environments.

Our defense approach using multi-environment training provides a more robust training environment that mitigates the impact of poisoning attacks. It encourages the agent to learn a generalized policy effective across multiple environments. Additionally, it can be combined with other defense mechanisms to further enhance the agent's resilience against poisoning attacks.

To detect reward poisoning, we employ a variance-based technique. We compare the observed rewards to an expected value under an unpoisoned reward signal. By computing the variance of observed rewards across multiple environments, we can determine if the agent has been subject to reward poisoning.

The variance-based technique assumes that the attacker cannot poison all rewards in all environments, resulting in higher variance under a poisoned reward signal compared to an unpoisoned one. We calculate the reward variance $Var(R)$ using the observed rewards R_i and the mean reward $\bar{R}$ across all environments:

$$Var(R) = \frac{1}{n-1} \sum_{i=1}^{n} (R_i - \bar{R})^2 \tag{8}$$

where n is the number of environments.

To compare observed rewards to an expected unpoisoned value, we calculate the variance of the unpoisoned rewards in each environment. The variance is computed using the formula:

$$Var(U) = \frac{1}{n-1} \sum_{i=1}^{n} (U_i - \bar{U})^2 \tag{9}$$

where U_i represents the expected unpoisoned reward in the i^{th} environment, $\bar{U}$ is the mean expected unpoisoned reward across all environments, and n is the number of training environments.

If the ratio of the observed reward variance to the expected unpoisoned reward variance exceeds a threshold value h, we determine that the agent has been subjected to reward poisoning, as indicated by the equation:

$$\frac{Var(R)}{Var(U)} > h \tag{10}$$

Our defense mechanism involves the environment $e_i \in \mathcal{E}$, the agent, and the attacker. We propose Algorithm 1 to implement our approach, which utilizes multi-environment training to defend against reward poisoning attacks. By following this algorithm, the agent learns a generalized policy across multiple environments, enhancing its resistance to reward poisoning attacks. The algorithm also imposes constraints on the attacker, making it more difficult for them to poison all rewards of all actions in all environments without detection.

6 CONCLUSION

This work addresses reward poisoning attacks in deep reinforcement learning by introducing a novel defense mechanism. We propose training agents in a

multi-environment setting, randomly selecting environments for agent interaction. By averaging rewards across multiple environments, our approach effectively mitigates the impact of poisoning and enhances agent robustness. Our method ensures the preservation of true reward performance while providing provable guarantees for defense policy effectiveness, ensuring safety and reliability in critical applications. This contribution represents a significant advancement in the development of robust and secure deep reinforcement learning systems for real-world scenarios. Future goals include conducting experiments to compare our approach with existing defenses, validating its effectiveness and practicality, and leveraging the insights gained to further enhance our defense mechanism.

REFERENCES

Akhtar, N. and Mian, A. (2018). Threat of adversarial attacks on deep learning in computer vision: A survey. *Ieee Access*, 6:14410–14430.

Banihashem, K., Singla, A., and Radanovic, G. (2021). Defense against reward poisoning attacks in reinforcement learning. *arXiv preprint arXiv:2102.05776*.

Behzadan, V. and Munir, A. (2017a). Vulnerability of deep reinforcement learning to policy induction attacks. In *Machine Learning and Data Mining in Pattern Recognition: 13th International Conference, MLDM 2017, New York, NY, USA, July 15-20, 2017, Proceedings 13*, pages 262–275. Springer.

Behzadan, V. and Munir, A. (2017b). Whatever does not kill deep reinforcement learning, makes it stronger. *arXiv preprint arXiv:1712.09344*.

Behzadan, V. and Munir, A. (2018). Mitigation of policy manipulation attacks on deep q-networks with parameter-space noise. In *Computer Safety, Reliability, and Security: SAFECOMP 2018, Västerås, Sweden, September 18, 2018, Proceedings 37*, pages 406–417. Springer.

Bouhaddi, M., Radjef, M. S., and Adi, K. (2018). An efficient intrusion detection in resource-constrained mobile ad-hoc networks. *Computers & Security*, 76:156–177.

Chen, Y., Du, S., and Jamieson, K. (2021). Improved corruption robust algorithms for episodic reinforcement learning. In *International Conference on Machine Learning*, pages 1561–1570. PMLR.

Greydanus, S., Koul, A., Dodge, J., and Fern, A. (2018). Visualizing and understanding atari agents. In *International conference on machine learning*, pages 1792–1801. PMLR.

Huang, S., Papernot, N., Goodfellow, I., Duan, Y., and Abbeel, P. (2017). Adversarial attacks on neural network policies. *arXiv preprint arXiv:1702.02284*.

Inkawhich, M., Chen, Y., and Li, H. (2019). Snooping attacks on deep reinforcement learning. *arXiv preprint arXiv:1905.11832*.

Kiran, B. R., Sobh, I., Talpaert, V., Mannion, P., Al Sallab, A. A., Yogamani, S., and Pérez, P. (2021). Deep reinforcement learning for autonomous driving: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 23(6):4909–4926.

Kos, J. and Song, D. (2017). Delving into adversarial attacks on deep policies. *arXiv preprint arXiv:1705.06452*.

Lin, Y.-C., Hong, Z.-W., Liao, Y.-H., Shih, M.-L., Liu, M.-Y., and Sun, M. (2017). Tactics of adversarial attack on deep reinforcement learning agents. *arXiv preprint arXiv:1703.06748*.

Lykouris, T., Simchowitz, M., Slivkins, A., and Sun, W. (2021). Corruption-robust exploration in episodic reinforcement learning. In *Conference on Learning Theory*, pages 3242–3245. PMLR.

Ma, Y., Zhang, X., Sun, W., and Zhu, J. (2019). Policy poisoning in batch reinforcement learning and control. *Advances in Neural Information Processing Systems*, 32.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. (2015). Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533.

Rakhsha, A., Radanovic, G., Devidze, R., Zhu, X., and Singla, A. (2020). Policy teaching via environment poisoning: Training-time adversarial attacks against reinforcement learning. In *International Conference on Machine Learning*, pages 7974–7984. PMLR.

Schulman, J., Levine, S., Abbeel, P., Jordan, M., and Moritz, P. (2015). Trust region policy optimization. In *International conference on machine learning*, pages 1889–1897. PMLR.

Sutton, R. S. and Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.

Wang, J., Liu, Y., and Li, B. (2020). Reinforcement learning with perturbed rewards. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 6202–6209.

Wei, C.-Y., Dann, C., and Zimmert, J. (2022). A model selection approach for corruption robust reinforcement learning. In *International Conference on Algorithmic Learning Theory*, pages 1043–1096. PMLR.

Wu, F., Li, L., Xu, C., Zhang, H., Kailkhura, B., Kenthapadi, K., Zhao, D., and Li, B. (2022). Copa: Certifying robust policies for offline reinforcement learning against poisoning attacks. *arXiv preprint arXiv:2203.08398*.

Zhang, H., Chen, H., Boning, D., and Hsieh, C.-J. (2021a). Robust reinforcement learning on state observations with learned optimal adversary. *arXiv preprint arXiv:2101.08452*.

Zhang, X., Ma, Y., Singla, A., and Zhu, X. (2020). Adaptive reward-poisoning attacks against reinforcement learning. In *International Conference on Machine Learning*, pages 11225–11234. PMLR.

Zhang, Z., Lim, B., and Zohren, S. (2021b). Deep learning for market by order data. *Applied Mathematical Finance*, 28(1):79–95.