Towards Secure Multi-Agent Deep Reinforcement Learning: Adversarial Attacks and Countermeasures

Changgang Zheng[†], Chen Zhen[◇], Haiyong Xie[◇], Shufan Yang[§]

[†]Department of Engineering Science, University of Oxford, Oxford, UK
[◇]School of Computer Science and Technology, University of Science and Technology of China, Hefei, China
[§]School of Computing, Edinburgh Napier University, Edinburgh, UK

changgang.zheng@eng.ox.ac.uk, cz2016@mail.ustc.edu.cn, haiyong.xie@ieee.org, S.Yang@napier.ac.uk

***Abstract*—Reinforcement Learning (RL) is one of the most popular methods for solving complex sequential decision-making problems. Deep RL needs careful sensing of the environment, selecting algorithms as well as hyper-parameters via soft agents, and simultaneously predicting which best actions should be. The RL computing paradigm is progressively becoming a popular solution in numerous fields. However, many deployment decisions, such as security of distributed computing, the defence system of network communication and algorithms details such as frequency of batch updating and the number of time steps, are typically not treated as an integrated system. This makes it difficult to have appropriate vulnerability management when applying deep RL in real life problems. For these reasons, we propose a framework that allows users to focus on the algorithm of reasoning, trust, and explainability in accordance with human perception, followed by exploring potential threats, especially adversarial attacks and countermeasures.**

***Index Terms*—Secure computing, Multi-agents, Deep reinforcement learning**

I. Introduction

Reinforcement learning (RL), as a research paradigm, not only learns from trial-and-error without prior knowledge of the environment modelling but also learns a more powerful strategy that can deal with dynamic environments [1]. In the simplest case, this means that RL can estimate the state-transition probabilities of the environment and how the immediate rewards under the new environment [2]. It also means that RL has the ability to predict the future response of the environment under agents' behaviours (a.k.a action). Therefore, observing software agents' behaviours in an RL system could provide insights into the future environment through software agents' self replay. It also helps researchers to investigate which policy will be the best to fit long-term goals.

Most popular RL methods that are based on temporal difference predictions can be categorised as either deterministic policy methods [3] or policy gradient based methods [4]: a deterministic policy method offers optimal policies for model based scenarios, while a policy gradient method offers a set of optimal solutions via stochastic gradient based optimisation. Deep RL combines the deep learning and the reinforcement learning methods [5], where deep learning tackles the problem of training artificial neural networks in an effective way such that it can be leveraged in the deep RL computing paradigm as a function approximation. Deep RL is one of key advances in recent advanced algorithm development; however, only a fraction of real life applications have successful RL case studies [6]. A fundamental problem remains: how to enable a secure multi-agent deep RL paradigm? This secure system should provide a framework to support computer security foundations such as risk based log analysis, ease of use, increased resilience, reduced vulnerabilities, and design with communication networks in mind to provide scalability.

We believe that there exist three key barriers for the widespread adoption of deep RL: **reasoning, trust, and explainability**. These fundamental questions are not separate. Together, they form a holistic strategy for applying the deep RL paradigm to a secure system. So far, solving these questions simultaneously has not been attempted, owing to the scale of the problem and depth of understanding of deep RL on multi-agents scenarios. In this paper we review threats that potentially exist in canonical distributed learning environments and multi-agent deep learning systems, followed by an overview of adversarial attack and countermeasures are provided. Finally based on our previous works [7]–[9], a secure system with the three elements of reasoning, trust, and explainability is proposed to allow the community to make further progress in building more secure models.

The contribution of this paper is to propose a new framework to provide a secure deep RL computing paradigm for real life applications. We hypothesize that a best solution of building secure deep RL framework is to provide a framework with scalable machine reasoning techniques and explainability techniques to explain behaviours of those algorithms in accordance with human perception. Our proposed new framework can effectively cope with the inherent uncertainty and variation in a dynamic environment with the ability to respond to adversarial attacks and other threats.

The authors thank the BCS Edinburgh Branch Committee for their support.

978-1-6654-2141-6/22/$31.00 ©2022 IEEE

II. Threat Analysis

In this section, we go through threats that may potentially cause harm to the multi-agent reinforcement learning programming paradigm. We start with the threats that are generated from distributed learning systems, followed by discussions on deep RL methods in multi-agent scenarios.

A. Distributed Learning Systems

Different from centralised systems, distributed systems have a different set of potential problems that threaten the normal system functions [10]. Use the computer network as an example, when nodes communicate with each other, packets are transmitted and queued throughout the network, which results in communication delays. Such delays are totally acceptable in small-size networks; however, when it scales to hundreds or thousands of nodes, it is inevitably possible that a node will fail [11]. Another typical severe case is the massive failure of nodes due to unstable network environments (e.g., the lost connection of nodes located in a server room due to the failure of the network) [12], which can easily result in the collapse of a distributed system.

There have been many theories and algorithms proposed: for instance, the CAP theorem [13] and the BASE theorem (i.e., basically available, soft-state, eventually-consistent) [14] that are related to distributed learning system design and implementation. In a distributed system, only two of the three attributes, namely, consistency, availability and partition tolerance, can be satisfied simultaneously [15]. Among them, partition tolerance must be satisfied by default. Thus, a balance must be made between consistency and availability. Satisfying consistency will sacrifice availability, where the client requests access to data but the server cannot respond in a timely manner. The requirement for availability comes at the expense of consistency, which can lead to inconsistent results when clients access data from different nodes.

Although less emphasis on availability, there are many distributed locks targeting to improve the consistency of distributed systems. A typical example is a key-value store system named Redis [16]. Even with high consistency, Redis can still be vulnerable to several threats, such as unauthorized access, writing to web shell, and key secure login. Among these, one typical scenario is that client A forgets to release the lock, which expires and therefore is automatically released; however, the task is not terminated and client B acquires a lock, at which point the shared resource is not secure. Another scenario is that locking and setting timeouts are separate operations and are not atomic. As a result, if the lock is successful but the timeout fails, the lock key will not expire, in which case the number of non-expiring lock keys will increase and potentially cause Redis to run out of memory space.

B. Multi-agent Deep Reinforcement Learning

Reinforcement learning is a learning method that allows agents to interact with the environment, which performs well for complex tasks in various applications, such as playing video games like StarCraft [17], HoK [18], [19], the traditional board game of Go [20], or cooling data centres [21]. However, the recent International Conference on Learning Representation (ICLR) revealed that most methods are limited to single, stationary agents instead of a learning environment with multi-agents and multi-objective [22], [23]. Under this assumption, when each agent is motivated by its own rewards and does actions to advance its local goals, local goals can be opposed to the interests of other agents, resulting in complex group decision-making behaviour [24]. For instance, resources allocations in health and social care systems, such as hospital beds, care home capacity, and associated resources of doctors, nurses, social care workers and equipment [25], [26]. These medical resources should be simultaneously allocated to those patients who are in need of care, and thus, each entity in this system has multiple objectives. For example, in an acute care setting, patients may care about delay related objectives, such as reducing mortality. In primary care, patients may care about minimising the expected waiting time for seeing consultants. In community care, patients may care about the satisfaction of care receivers. This will be a complex group decision-making behaviour and will be hard to find a policy for global optimal.

Solving distributed optimisation for multi-agent reinforcement learning is a closely related task of multi-agent networks. Every agent has a private input, where the interest between each agent's private input value may have conflicts. The goal of the overall agents is to balance these inputs and reach an agreement that optimises the overall system performance. Specifically, the goal is to collaboratively arrange the cost function of every agent and formulate a proper global objective. The threats from multi-agent optimisation are particularly related to the perturbation of agent actions and false environment sensing and failed gradient-based optimisation [27], [28]. In multi-agent deep RL, those treats are particularly hard to detect since the nature of distribution and stochastic elements of reward, states or reward-states pair.

Reaching consensus and solving distributed optimisation is not an easy task and may threats the overall multi-agent networks. When the agent number is large and agents are different from each other, especially the environment is complex and dynamic, it will be difficult to reach an agreement on a value function and thus result in low system performance. The consensus between agents in multi-agent systems is hard to reach [29], [30]. However, when using simple consensus methods, like an average consensus, it is much easier to be attacked and result in the leaking of privacy.

III. Adversarial Attack

Since software agents are vulnerable to small adversarial perturbations on the agent's inputs under the deep reinforcement learning paradigms [31], it is important to overview adversarial attacks as one of the main threats.

A. Adversarial Attack in Deep Learning

Adversarial attack is a major threat in deep learning methods that rely on stochastic gradient descent. This attack is able to

fool the machine with some minor and well-designed changes, which are hard for even humans to detect [32]. This section introduces these attacks through different attacking paradigms.

1) Classified by Attacking Environment: The most common adversarial attacks on deep learning algorithms are white-box attacks, grey-box attacks, and black-box attacks.

A white-box attack refers to when the attacker can access all information about the network, including model, parameters, weights, and test input and output. Typical white-box attacks include FGSM [33], JSMA [34], DeepFool [35], C&W [36], PGD [37].

Grey-box attacks represent the middle groud of black and white box attacks. Attackers can access the input and output of the network and have limited information about the model, such as the model structure, but have no access to the model parameters and weights.

Black-box attacks only allows attackers to access the input and output of the target network. Some typical black-box attacks include NES [38], ZOO [39], SPSA [40], N-Attack [41], Boundary [42], Evolutionary [43], which are usually limited by scalability.

2) Classified by Attacking Targets: Adversarial attacks can be classified as untargeted attacks and targeted attacks. The untargeted attack aims to fool neural networks, misleading the network to make false decisions. The targeted attack, more than simply fooling the network, has some extra requirements, which aim to generate a particular false pattern.

3) Typical Attacking Method: Attacking deep learning mainly focuses on three targets: the gradient, the optimization process, and the decision.

- Gradient-based Attacks: The gradient-based attack imposes certain hardly perceptible perturbations to mislead the network by maximising the network's prediction error. However, the end to end mappings of deep neural networks is fairly discontinuous to a significant extent [44].
- Optimisation-based Attacks: Unlike attacks that utilise the transferability of the substitutes model, the optimised attack uses a zero-order (derivative-free) optimisation [39] to directly estimate the gradient of the target DNN for generating adversarial examples.
- Decision-based Adversarial Attack: This type of attack works on both a robust decision boundary of the saddle point problem and a decision boundary that simply separates the benign data points. The decision-based attack is a one method of attacking based on mainly hyperparameter tuning [42].

B. Adversarial Attacks in Deep Reinforcement Learning

Most researches focus on adversarial attacks in deep learning for classification or prediction purposes. Deep RL not only includes adversarial perturbations on the training stage, but also the state space, reward, and action spaces. Generally deep RL can be attacked from five perspectives: observation, environment, reward, action, and policy.

1) Observation-based Attack: The observation based attack includes a series of attacks applied to all procedures of observation. On top of deep learning-based attacks, observation-based attacks create a proxy environment [45], [46], using the proxy environment to create attacker modified observations. Thus, this leads to the malfunctioning of the deep RL algorithm. Attackers can even input carefully trained fake observations which lead to a faulty pattern [47]–[49]. Moreover, timing is also important in attacking deep RL. Researchers have focused on using thresholds to determine the critical time, action, or environment at which the attack can be released [47], [48], [50].

2) Reward-based Attack: The reward-based attack tampers with the action values that the environment gives to the agent. For example, when some key instance occurs (e.g., identified by time or space information), changing or flipping the action values (rewards) of each agent can slow done or even break the learning process [51]–[53]. The most obvious reward-based attack is to change the reward to optimise the loss function of the attack. Besides, these flips and changes can mislead deep reinforcement learning agents so that they are unable to learn or act under a mislead paradigm.

3) Environment-based Attack: Similar but different from the observation-based attack, environment-based attacks focus only on the environment instead of when and how to observe. These attacks focus on changing the environment at key steps to slow down or mislead the learning process. Attackers can place obstacles at some key points based on reward or gradient information [54], [55]. There is a more diverse and complex index to find the key points and paradigms of the changing environment [56].

4) Action-based Attack: Action-based attackers impose attacks across the action space dimensions. Attackers usually place noise in the action space and minimise the reward value during the policy evaluation process of agents [57]. Similar to the environment-based attack, the action space is separate from the policy, which can be fatal to deep RL, especially those combined with time-coherent attacks.

5) Policy-based Attack: The policy-based attack aims to confuse the policy. Deep RL agents usually learn well in terms of policy and are able to achieve a high score in the zero-sum game. However, under the usual training environment, the learned policy does not apply to all circumstances in the rule space. When there are other agents who show a valid move allowed by the rules but do not learn from the agent in advance, the learned policy may be fooled, resulting in a low reward [58].

IV. Counter measurements

Here we review methods for developing effective countermeasures for adversarial attacks.

A. Common Defence Methods

1) Adversarial training: Adversarial training is also a commonly used defence technique in deep RL. It trains the deep neural network together with attacking samples, which

increases the robustness of the trained network to attacks. One simple example is adding distribution and noise to the training sample (agents) for higher robustness [49], [59]. Other than inputs, adversarial training is also applied based on gradient [55], where the trained model has a high generalization ability to defend against similar attacks. Not all the time needs adversarial training inputs, some works use the greedy method or significant level of actions to determine if the adversarial training needs to be applied, and have a similar result with less adversarial training inputs [60], [61].

2) Robustness training: Robustness training is another method of defence against adversarial attacks. The robustness can be represented in adding noise to both environment, action and reward [62]–[64]. These noises can also be simulated by adding an additional dummy agent and competing against it [62], [65]. The additional agent brings a higher level of competition, which increases the capacity of agents' robustness in acting on the original environment.

3) Adversarial detection: Adversarial detection techniques can be used to provide an early warning to adversarial attacks and trigger defence actions so as to realise the defence. The detection can be environment-based, policy-based, and action-based, where the system will evaluate if the current applied policy [66], action [67] and environment are being attacked. When that happens, the system will take action, such as interrupting the learning process, predicting the input [68], and fixing or changing the policy.

4) Federated learning: Threats from distributed learning systems that use the deep learning techniques often rely on leveraging the reduction of raw data process implied by the federated learning techniques [69]. Federated Learning is a new framework to address those threats. Basic federated learning consists in distributing multiple computing units that form an artificial neural network for completing tasks such as object detection and recognition in a mobile device, standalone device and computer server, without exchanging users' data samples. Due to this learning paradigm, federated learning can take advantage of the distributed computing power and data (with different privacy level) from all end users to obtain a more powerful model [70].

B. Four Key Procedures

In summary, the mitigation of adversarial attacks has four key procedures. Preventing failures, detecting attacks, locating attacks, and mitigating attacks. The previous four counter measurement methods are used in these procedures.

1) Preventing Failures: Failure prevention can be done through robustness and adversarial training. More generalized models can survive in many attacks. Besides, early warning of weaknesses and early detection of attacks both help to prevent system failure.

2) Detecting Attacks: The key to the survival of the deep RL system is the detection of attacks. Accurately reporting the happening of the attack can help the system to reduce the effect of the attack. The runtime analysis of attacks, e.g., predicting types, volume, and procedure, helps to mitigate the attack.

3) Locating Attacks: Knowing the time of attach, which inputs or decisions belong to the attack, and how these changes resulted in failure, can help to defend from the current attack and further build RL systems with higher robustness. The reasoning behind the failure has an influential impact on how to mitigate and prevent the attack.

4) Mitigating Attacks: There are several attack mitigation policies. After locating the attack and understanding the reasoning behind the attack, the system can apply the relevant defence policy. For example, with the observation environment and reward attack, the system can stop the learning process or filter out the attacking inputs or learn from selected inputs, for the policy-based attack, the system can switch to another pre-trained policy. All these counter measurements can protect the system from further attacks and push the system back to normal functioning.

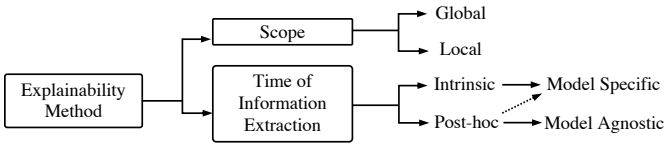

Fig. 1. XAI methods taxonomy [71].

V. Explainability

Explainability is often referred to interpretability, which is the ability to explain how the optimal solutions are found through training or searching in problem spaces. In RL, the explainability can be divided into two dimensions [71], [72] (Figure 1):

1) When to explain: When information is extracted is essential to the explainability of the overall multi-agent RL system [8]. There are two trends, one is to allow the system to be intrinsically explainable from the training phase [73] [71]. Specifically, this ensures every step of training (training data, cost function, rule formulation) is explainable. Although not directly belonging to the RL system, Decision Tree and Explainable Artificial Intelligence [74] are two good examples of training explainability, where the trained model have a clear structure that can originally provide easy reasoning and can be understandable to domain experts [75]. The other trend is to explain the model after the model is trained and converged, which is also named as post-hoc explainability [8]. This type of explainability does not need to have the knowledge of the model [76]. It just learns from the system input-output and extracts the explainable rules to model the system behaviour [77].

2) Scope of the explanations: The explanation scope can be global or local [71]. Local explanation focus on fine-grained system activities, for example, why an agent does some specific activities at a certain time [71]. The fine-grained explanation allows experts to diagnose the system in a easier way, which helps to do early warning and provide a more secure system. The global explanations are more

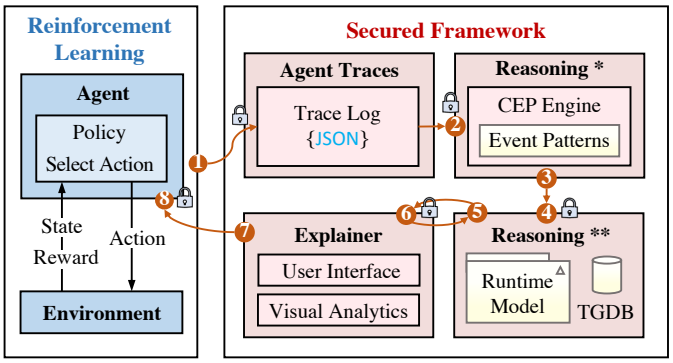

Fig. 2. Framework of a secured multi-agent reinforcement learning programming paradigm. The secured system is based on the extension work of the first explanation for reinforcement learning, ETeMoX framework [8].

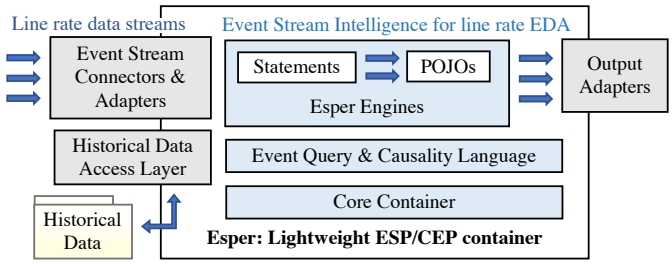

Fig. 3. Esper systems [78], one of the typical CEP engine.

coarse-grained and usually focused on multi-agent and their interactive activities and trends. The coarse-grained system usually lasts for a longer time to better understand the strategy and how it influences the reward in a broader time period, which helps to provide a more general understanding of the system [73].

The first explanation for reinforcement learning by using temporal database run time mode was published in 2021 [8]. Parrar et al. described the feasibility of using hark query on a temporal database to provide explainable deep RL (e.g. [9]) for run-time explanations and post-hoc analyses. This work also lays the foundation of our proposed secure deep RL computing paradigm.

VI. A Secure Multi-Agent Reinforcement Learning Programming Paradigm

We hypothesis that a secure multi-agent deep RL computing paradigm needs support machine reasoning, trust, and explainability. The framework we proposed in below, as shown in Figure 2, can provide support to defend against malicious attack and provide capability for a general multi-agent deep RL systems. There are four components in the framework: *Agent Traces*, *Machine Reasoning* *, *Machine Reasoning* ** and *Explainer* and *Trust Authentication Interfaces*. These will be described in detail below.

1) Agents' Traces: Our implementation decouples observation in the system from agents' traces. The agent traces will be extracted as log files and forwarded to the *Agent Traces* component. These logs contain all agent-related information from its environment input, model parameters, hyperparameters, actions, rewards, and other sensing data, or even information from human observers. These collected time series logs files will be send to a translator component through a authentication interface (❶ in Figure 2). Based on the different translator designs, the log file can be self-defined and structured like JSON, XML, or plain text with key-value pairs. These raw data will then be processed into a format that can be understood by the complex event engine for further process in the *Machine Reasoning** component(❷ in Figure 2).

2) Machine Reasoning:* This component mainly aims to locate conditions (patterns or events) from the traces of the translated log files from the *Agent Traces* component. The located patterns, events and their associate information will be sent to the Runtime Model and Temporal Graph Database (TGDB) in the second stage of *Machine Reasoning*** component. In detail, based on the complex event engine (CEP), as the first stage of machine reasoning, the event and pattern will be selected and the reasoning of agent traces selection will be located. One typical CEP example is the Esper [78]. As shown in Figure 3, Esper is able to realise runtime correlations location between simple events and the predefined paradigm from log files of each individual agent trace. Esper provides streaming analysis, available for Java as well as for .NET framework [79]. Users will define the outlook and characteristics of these paradigms based on their tasks and interests (e.g. what is the characteristic of an agent when it is stuck somewhere). These can be defined in the Esper EPL and deployed to the Esper engine. When the Esper engine detects the event, the system will do the predefined event-triggered actions (e.g. where, when, and how the agent is stuck) and will collect the data based on needs and send them to the second *Machine Reasoning*** component for further process (❸ to ❹ in Figure 2). Again the interfaces are clients authenticated to guarantee secure communication.

*3) Machine Reasoning**:* In this component, the observations and reasoning (received indexed event data) from the previous component *reasoning** are linked to the system goals and decisions. This linking task is realised by using the Eclipse Modeling Framework[1] (EMF) [7], which focus on two directions. The first direction is generalised observation, which mainly analyses the general system performance over a long period. The second direction is goal-oriented observation, which mainly focuses on specific task-related events.

The runtime model based on Java will update the TGDB and help to create strategies based on a predefined response paradigm. The updated temporal graph will be auto-generated and adjusted by the model indexer according to the coming logs. The updated graph will replace the old one at some specific time based on needs. The TGBD can help to trace the log (e.g. changing of model parameters) of each agent and show the intrinsic relationship between different log parame-

[1]Eclipse Modeling Framework: https://www.eclipse.org/modeling/emf/

ters. These will be sent to the final component *Explainer* via clients authenticated interfaces (❺ to ❻ in Figure 2) for further process.

4) Explainer: Obligations of this kind are especially relevant for systems that rely on reinforcement learning since the reasoning processes of such systems are typically opaque. It is essential to provide a graphical explanation system that helps carers and people who need care to understand the operation and limitations of the high dimension data that is used in software, particularly with optimisation problems. The explainer component is based on temporal difference algorithms read MQTT message through ❼ to ❽ in Figure 2, in which RL algorithms to be equipped with the ability to provide human understandable reasoning including user oriented interface and high dimension visual analytic techniques.

5) Trust Authentication Interfaces: The most easy way of adding authentication interfaces is based on assigning an individual secret key to each user, so that machine reasoning components can sign off their own downstream data, which is based on first conceptualized and implemented in 1984 by Shamir [80]. However, those may bring limitations of privacy with deniable signatures. Enabling regular change of signature does make, and this method is arguably still the best lightweight authentication technique. Take the advantage of the fast computing techniques, those lightweight authentication methods are still one of the best practical solutions. Those lightweight authentication brings a level of secured communication in a general communication network.

VII. Discussion and Future Work

The classical adversarial attack is only focused on the threats from the artificial neural networks, with most of the work focused on supervised learning and unsupervised learning methods. Deep RL learning is used to generate optimal actions, which depends on consecutively neural network predictions unlike single predictions at a time in classification or regression problems. It is essential to look into potential threats and adversarial attacks to allow research communities to have a secure system to apply deep RL algorithms into real life scenarios with safeguarding procedures.

We discussed the adversarial attacks by categorising them into observation-based attack, reward-based attach, environment-based attack, action-based attack and policy-based attack. Most of counter measurements are focusing on making the RL or deep RL algorithms to learn a robust policy or alternatively depending on the observation base on image inputs instead of environment sensing from agents.

Due to the evolution of new attack techniques, the chance of developing of a inherently robust deep RL algorithms is challenging. We hypnosis that a best solution of building defensible deep RL framework is to provide a framework with scalable machine reasoning techniques and explainability techniques to explain behaviour of those algorithms in accordance with human perception. In this paper, we propose a framework of providing secure defense system and discuss the feasibility of use temporal database to provide safeguarding for deep RL systems. Despite the vulnerabilities of deep RL methods, it is possible to build a secure multi-agent deep RL framework to benefit in robotics, smart city and industry4.0 initiatives. A decoupled approach with temporal database and advanced machine reasoning methods could provide techniques to implement necessary surveillance. Our future work will focus on thoroughly testing our framework using widely popular adversarial attackers, such as Foolbox [81] to evaluate our systems. Also like traditional standard secure computing systems, we are planning to develop benchmarks to quantify the robustness and resilience of those RL and deep RL algorithms.

Acknowledgment

We would like to thank Juan Marcelo Parra-Ullauri, Antonio García-Domínguez, Nelly Bencomo, Juan Boubeta-Puig and Guadalupe Ortiz for initiating the conversion of this framework and for providing the first public repository.

References

[1] Y. Wang, H. He, and C. Sun, "Learning to navigate through complex dynamic environment with modular deep reinforcement learning," *IEEE Transactions on Games*, vol. 10, no. 4, pp. 400–412, 2018.

[2] S. Ishii, W. Yoshida, and J. Yoshimoto, "Control of exploitation–exploration meta-parameter in reinforcement learning," *Neural networks*, vol. 15, no. 4-6, pp. 665–687, 2002.

[3] P. Wang, H. Li, and C.-Y. Chan, "Continuous control for automated lane change behavior based on deep deterministic policy gradient algorithm," in *2019 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2019, pp. 1454–1460.

[4] Z. Zheng, J. Oh, and S. Singh, "On learning intrinsic rewards for policy gradient methods," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[5] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[6] V. Pong, S. Gu, M. Dalal, and S. Levine, "Temporal difference models: Model-free deep rl for model-based control," *arXiv preprint arXiv:1802.09081*, 2018.

[7] J. M. Parra-Ullauri, A. García-Domínguez, L. H. García-Paucar, and N. Bencomo, "Temporal models for history-aware explainability," in *Proceedings of the 12th System Analysis and Modelling Conference*, 2020, pp. 155–164.

[8] J. M. Parra-Ullauri, A. García-Domínguez, N. Bencomo, C. Zheng, C. Zhen, J. Boubeta-Puig, G. Ortiz, and S. Yang, "Event-driven temporal models for explanations - ETeMoX: explaining reinforcement learning," *Software and Systems Modeling*, Dec. 2021.

[9] C. Zheng, S. Yang, J. M. Parra-Ullauri, A. Garcia-Dominguez, and N. Bencomo, "Reward-reinforced generative adversarial networks for multi-agent systems," *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2021.

[10] R. L. Sherman, "Distributed systems security," *Computers & Security*, vol. 11, no. 1, pp. 24–28, 1992.

[11] C. Dobre, C. Stratan, and V. Cristea, "Realistic simulation of large scale distributed systems using monitoring," in *2008 International Symposium on Parallel and Distributed Computing*, 2008, pp. 434–438.

[12] Z. Huang, C. Wang, M. Stojmenovic, and A. Nayak, "Characterization of cascading failures in interdependent cyber-physical systems," *IEEE Transactions on Computers*, vol. 64, no. 8, pp. 2158–2168, 2015.

[13] E. Brewer, "A certain freedom: thoughts on the cap theorem," in *Proceedings of the 29th ACM SIGACT-SIGOPS symposium on Principles of distributed computing*, 2010, pp. 335–335.

[14] D. Pritchett, "Base: An acid alternative: In partitioned databases, trading some consistency for availability can lead to dramatic improvements in scalability." *Queue*, vol. 6, no. 3, p. 48–55, may 2008. [Online]. Available: https://doi.org/10.1145/1394127.1394128

[15] S. De Angelis, L. Aniello, R. Baldoni, F. Lombardi, A. Margheri, and V. Sassone, "Pbft vs proof-of-authority: Applying the cap theorem to permissioned blockchain," 2018.

[16] "Redis," https://redis.io/, accessed: 2022-05-17.

[17] O. Vinyals, I. Babuschkin, W. M. Czarnecki, M. Mathieu, A. Dudzik, J. Chung, D. H. Choi, R. Powell, T. Ewalds, P. Georgiev *et al.*, "Grandmaster level in starcraft ii using multi-agent reinforcement learning," *Nature*, vol. 575, no. 7782, pp. 350–354, 2019.

[18] D. Ye, G. Chen, W. Zhang, S. Chen, B. Yuan, B. Liu, J. Chen, Z. Liu, F. Qiu, H. Yu *et al.*, "Towards playing full moba games with deep reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 621–632, 2020.

[19] D. Ye, Z. Liu, M. Sun, B. Shi, P. Zhao, H. Wu, H. Yu, S. Yang, X. Wu, Q. Guo *et al.*, "Mastering complex control in moba games with deep reinforcement learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 04, 2020, pp. 6672–6679.

[20] E. Gibney *et al.*, "Google ai algorithm masters ancient game of go," *Nature*, vol. 529, no. 7587, pp. 445–446, 2016.

[21] T. Wei, S. Ren, and Q. Zhu, "Deep reinforcement learning for joint datacenter and hvac load control in distributed mixed-use buildings," *IEEE Transactions on Sustainable Computing*, vol. 6, no. 3, pp. 370–384, 2019.

[22] Y. Wang, Y. Xia, T. He, F. Tian, T. Qin, C. Zhai, and T.-Y. Liu, "Multi-agent dual learning," in *Proceedings of the International Conference on Learning Representations (ICLR) 2019*, 2019.

[23] J. Kang, M. Liu, A. Gupta, C. Pal, X. Liu, and J. Fu, "Learning multi-objective curricula for deep reinforcement learning," *arXiv preprint arXiv:2110.03032*, 2021.

[24] C. Zheng, S. Yang, J. Parra-Ullauri, A. Garcia-Dominguez, and N. Bencomo, "Reward-reinforced reinforcement learning for multi-agent systems," 2021. [Online]. Available: https://arxiv.org/abs/2103.12192

[25] C. Zhang, C. Gupta, A. Farahat, K. Ristovski, and D. Ghosh, "Equipment health indicator learning using deep reinforcement learning," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2018, pp. 488–504.

[26] J. Weltz, A. Volfovsky, and E. B. Laber, "Reinforcement learning methods in public health," *Clinical therapeutics*, 2022.

[27] J. Tu, T. Wang, J. Wang, S. Manivasagam, M. Ren, and R. Urtasun, "Adversarial attacks on multi-agent communication," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 7768–7777.

[28] N. N. Bakhtadze, I. B. Yadykin, V. A. Lototsky, E. M. Maximov, and E. A. Sakrutina, "Multi-agent approach to design of multimodal intelligent immune system for smart grid," *IFAC Proceedings Volumes*, vol. 46, no. 9, pp. 1164–1169, 2013.

[29] C. N. Hadjicostis and T. Charalambous, "Average consensus in the presence of delays in directed graph topologies," *IEEE Transactions on Automatic Control*, vol. 59, no. 3, pp. 763–768, 2013.

[30] C. N. Hadjicostis, N. H. Vaidya, and A. D. Domínguez-García, "Robust distributed average consensus via exchange of running sums," *IEEE Transactions on Automatic Control*, vol. 61, no. 6, pp. 1492–1507, 2015.

[31] B. G. Tekgul, S. Wang, S. Marchal, and N. Asokan, "Real-time attacks against deep reinforcement learning policies," *arXiv preprint arXiv:2106.08746*, 2021.

[32] J. Zhang and C. Li, "Adversarial examples: Opportunities and challenges," *IEEE transactions on neural networks and learning systems*, vol. 31, no. 7, pp. 2578–2593, 2019.

[33] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[34] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in *2016 IEEE European symposium on security and privacy (EuroS&P)*. IEEE, 2016, pp. 372–387.

[35] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "Deepfool: a simple and accurate method to fool deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2574–2582.

[36] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *2017 ieee symposium on security and privacy (sp)*. IEEE, 2017, pp. 39–57.

[37] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *arXiv preprint arXiv:1706.06083*, 2017.

[38] A. Ilyas, L. Engstrom, A. Athalye, and J. Lin, "Black-box adversarial attacks with limited queries and information," in *International Conference on Machine Learning*. PMLR, 2018, pp. 2137–2146.

[39] P.-Y. Chen, H. Zhang, Y. Sharma, J. Yi, and C.-J. Hsieh, "Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models," in *Proceedings of the 10th ACM workshop on artificial intelligence and security*, 2017, pp. 15–26.

[40] T. Foroud, A. Baradaran, and A. Seifi, "A comparative evaluation of global search algorithms in black box optimization of oil production: A case study on brugge field," *Journal of Petroleum Science and Engineering*, vol. 167, pp. 131–151, 2018.

[41] Y. Li, L. Li, L. Wang, T. Zhang, and B. Gong, "Nattack: Learning the distributions of adversarial examples for an improved black-box attack on deep neural networks," in *International Conference on Machine Learning*. PMLR, 2019, pp. 3866–3876.

[42] W. Brendel, J. Rauber, and M. Bethge, "Decision-based adversarial attacks: Reliable attacks against black-box machine learning models," *arXiv preprint arXiv:1712.04248*, 2017.

[43] Y. Dong, H. Su, B. Wu, Z. Li, W. Liu, T. Zhang, and J. Zhu, "Efficient decision-based black-box adversarial attacks on face recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7714–7722.

[44] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

[45] V. Behzadan and W. Hsu, "Adversarial exploitation of policy imitation," *arXiv preprint arXiv:1906.01121*, 2019.

[46] L. Hussenot, M. Geist, and O. Pietquin, "Copycat: Taking control of neural policies with constant attacks," *arXiv preprint arXiv:1905.12282*, 2019.

[47] V. Behzadan and A. Munir, "Vulnerability of deep reinforcement learning to policy induction attacks," in *International Conference on Machine Learning and Data Mining in Pattern Recognition*. Springer, 2017, pp. 262–275.

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

[49] J. Kos and D. Song, "Delving into adversarial attacks on deep policies," *arXiv preprint arXiv:1705.06452*, 2017.

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

[51] E. Tretschk, S. J. Oh, and M. Fritz, "Sequential attacks on agents for long-term adversarial goals," *arXiv preprint arXiv:1805.12487*, 2018.

[52] P. Kiourti, K. Wardega, S. Jha, and W. Li, "Trojdrl: Trojan attacks on deep reinforcement learning agents," *arXiv preprint arXiv:1903.06638*, 2019.

[53] Y. Han, B. I. Rubinstein, T. Abraham, T. Alpcan, O. D. Vel, S. Erfani, D. Hubczenko, C. Leckie, and P. Montague, "Reinforcement learning for autonomous defence in software-defined networking," in *International Conference on Decision and Game Theory for Security*. Springer, 2018, pp. 145–165.

[54] T. Chen, W. Niu, Y. Xiang, X. Bai, J. Liu, Z. Han, and G. Li, "Gradient band-based adversarial training for generalized attack immunity of a3c path finding," *arXiv preprint arXiv:1807.06752*, 2018.

[55] X. Bai, W. Niu, J. Liu, X. Gao, Y. Xiang, and J. Liu, "Adversarial examples construction towards white-box q table variation in dqn pathfinding training," in *2018 IEEE Third International Conference on Data Science in Cyberspace (DSC)*. IEEE, 2018, pp. 781–787.

[56] C. Xiao, X. Pan, W. He, J. Peng, M. Sun, J. Yi, M. Liu, B. Li, and D. Song, "Characterizing attacks on deep reinforcement learning," *arXiv preprint arXiv:1907.09470*, 2019.

[57] X. Y. Lee, S. Ghadai, K. L. Tan, C. Hegde, and S. Sarkar, "Spatiotemporally constrained action space attacks on deep reinforcement learning agents," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 04, 2020, pp. 4577–4584.

[58] A. Gleave, M. Dennis, C. Wild, N. Kant, S. Levine, and S. Russell, "Adversarial policies: Attacking deep reinforcement learning," *arXiv preprint arXiv:1905.10615*, 2019.

[59] A. Pattanaik, Z. Tang, S. Liu, G. Bommannan, and G. Chowdhary, "Robust deep reinforcement learning with adversarial attacks," *arXiv preprint arXiv:1712.03632*, 2017.

[60] V. Behzadan and A. Munir, "Whatever does not kill deep reinforcement learning, makes it stronger," *arXiv preprint arXiv:1712.09344*, 2017.

[61] V. Behzadan and W. Hsu, "Analysis and improvement of adversarial training in dqn agents with adversarially-guided exploration (age)," *arXiv preprint arXiv:1906.01119*, 2019.

[62] L. Pinto, J. Davidson, R. Sukthankar, and A. Gupta, "Robust adversarial reinforcement learning," in *International Conference on Machine Learning*. PMLR, 2017, pp. 2817–2826.

[63] V. Behzadan and A. Munir, "Mitigation of policy manipulation attacks on deep q-networks with parameter-space noise," in *International Conference on Computer Safety, Reliability, and Security*. Springer, 2018, pp. 406–417.

[64] K. Neklyudov, D. Molchanov, A. Ashukha, and D. Vetrov, "Variance networks: When expectation does not meet your expectations," *arXiv preprint arXiv:1803.03764*, 2018.

[65] Z. Gu, Z. Jia, and H. Choset, "Adversary a3c for robust reinforcement learning," *arXiv preprint arXiv:1912.00330*, 2019.

[66] A. Havens, Z. Jiang, and S. Sarkar, "Online robust policy learning in the presence of unknown adversaries," *Advances in neural information processing systems*, vol. 31, 2018.

[67] B. Lütjens, M. Everett, and J. P. How, "Certified adversarial robustness for deep reinforcement learning," in *Conference on Robot Learning*. PMLR, 2020, pp. 1328–1337.

[68] Y.-C. Lin, M.-Y. Liu, M. Sun, and J.-B. Huang, "Detecting adversarial attacks on neural network policies with visual foresight," *arXiv preprint arXiv:1710.00814*, 2017.

[69] Q. Yang, Y. Liu, Y. Cheng, Y. Kang, T. Chen, and H. Yu, "Federated learning," *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 13, no. 3, pp. 1–207, 2019.

[70] A. Fallah, A. Mokhtari, and A. Ozdaglar, "Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach," *Advances in Neural Information Processing Systems*, vol. 33, pp. 3557–3568, 2020.

[71] E. Puiutta and E. Veith, "Explainable reinforcement learning: A survey," in *International cross-domain conference for machine learning and knowledge extraction*. Springer, 2020, pp. 77–95.

[72] A. Heuillet, F. Couthouis, and N. Díaz-Rodríguez, "Explainability in deep reinforcement learning," *Knowledge-Based Systems*, vol. 214, p. 106685, 2021.

[73] A. Adadi and M. Berrada, "Peeking inside the black-box: a survey on explainable artificial intelligence (xai)," *IEEE access*, vol. 6, pp. 52 138–52 160, 2018.

[74] D. Gunning, M. Stefik, J. Choi, T. Miller, S. Stumpf, and G.-Z. Yang, "Xai—explainable artificial intelligence," *Science robotics*, vol. 4, no. 37, p. eaay7120, 2019.

[75] B. Letham, C. Rudin, T. H. McCormick, and D. Madigan, "Interpretable classifiers using rules and bayesian analysis: Building a better stroke prediction model," *The Annals of Applied Statistics*, vol. 9, no. 3, pp. 1350–1371, 2015.

[76] Z. C. Lipton, "The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery." *Queue*, vol. 16, no. 3, pp. 31–57, 2018.

[77] G. Ras, M. van Gerven, and P. Haselager, "Explanation methods in deep learning: Users, values, concerns and challenges," in *Explainable and interpretable models in computer vision and machine learning*. Springer, 2018, pp. 19–36.

[78] A. Mathew, "Benchmarking of complex event processing engine-esper," *Dept. Comput. Sci. Eng., Indian Inst. Technol. Bombay, Maharashtra, India, Tech. Rep. IITB/CSE/2014/April/61*, 2014.

[79] J. Richter, *Applied Microsoft. NET framework programming*. Microsoft Press Redmond, 2002, vol. 1.

[80] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Workshop on the theory and application of cryptographic techniques*. Springer, 1984, pp. 47–53.

[81] M.-I. Nicolae, M. Sinn, M. N. Tran, B. Buesser, A. Rawat, M. Wistuba, V. Zantedeschi, N. Baracaldo, B. Chen, H. Ludwig *et al.*, "Adversarial robustness toolbox v1. 0.0," *arXiv preprint arXiv:1807.01069*, 2018.