REINFORCEMENT LEARNING WITH SAFE EXPLORATION FOR NETWORK SECURITY

Canhuang Dai, Liang Xiao, Xiaoyue Wan, Ye Chen

Dept. Communication Engineering, Xiamen University, Xiamen, China. Email: lxiao@xmu.edu.cn

ABSTRACT

Safe reinforcement learning is important for the safety critical applications especially network security, as the exploration of some dangerous actions can result in huge short-term losses such as network failure or large scale privacy leakage. In this paper, we propose a reinforcement learning algorithm with safe exploration and uses transfer learning to reduce the initial random exploration. A blacklist is maintained to record the most dangerous state-action pairs as a safety constraint. A safe deep reinforcement learning version uses a convolutional neural network to estimate the risk levels and thus further improves the safety of the exploration and accelerates the learning speed for the learning agent. As a case study, the proposed reinforcement learning with safe exploration is applied in the anti-jamming robot communications. Experimental results show that the proposed algorithms can improve the jamming resistance of the robot and reduce the outage rate to enter the most dangerous states compared with the benchmark algorithms.

Index Terms— reinforcement learning, safe exploration, deep reinforcement learning, network security

1. INTRODUCTION

Reinforcement learning (RL) techniques such as Q-learning and deep Q-network (DQN) enable a learning agent to make decision via trial-and-error in network security applications, such as the anti-jamming communication [1] and malware detections [2]. However, most classical reinforcement learning techniques explore all the state-action pairs including those that will cause immediate network failure or huge user privacy leakage to estimate the expected reward of the policy especially at the initial learning. The learning speed of such reinforcement learning algorithms is often slower than the network dynamics in network security applications due to the difficulty accurately estimating the reward and the state especially under unknown attack policies.

Safe exploration has become an critical issue for reinforcement learning especially in the safety critical applications [3]. For instance, the seminal work in [3] uses a Gaussian process to model the safety constraints and only explores the safe state-action pairs following the safety constraints according to the knowledge of a safety function with Lipschitz continuity. Berkenkamp et al. further propose a safe model based reinforcement learning algorithm with Lyapunov stability and applies it in a simulated inverted pendulum system with known statical models of the dynamics [4].

Transfer learning (TL) can initialize the learning model by exploiting the knowledge gained from solving the previous similar tasks to accelerate learning. For example, Laroche et al. propose a transfer reinforcement learning with shared dynamics, in which the transition samples are cast in the target environment with a reshaped reward based on upper confidence bound [5]. In particular, the case-based reinforcement learning (CARL) as proposed in [6] uses the case-based reasoning as an instance-based state approximator for RL.

In this paper, we propose a Reinforcement Learning algorithm with Safe Exploration (RLSE) to enable a learning agent such as a smartphone or robot to choose the security policies. This algorithm evaluates the risk level of each state-action pairs, which can be the security performance metrics such as the probability of privacy leakage or network failure during the learning process. The action is chosen based on a modified Boltzmann distribution according to the Q-values and the risk levels to ensure safe exploration. Transfer learning is applied to initialize the learning parameters to reduce the initial explorations. A Deep Reinforcement Learning algorithm with Safe Exploration (DRLSE) is also designed to further accelerate the learning speed and reduce the outage probability, i. e., the rate for the learning agent to stay in the most risky states during the exploration. This algorithm uses a convolutional neural network (CNN) to estimate the Q-values and the risk levels of the actions under the current state.

As a case study, the proposed algorithms are applied in the anti-jamming communication, a typical wireless security scenario. Specifically, a robot chooses the communication policy including the transmit power and the moving direction to send the sensing data to a remote controller to resist the jammers that send faked signals to degrade/interrupt the ongoing communications. Experimental results verify the performance gain of the safe exploration regarding the jamming resistance of the robot and the outage probability.

This paper was in part supported by National Natural Science Foundation of China under Grant 61671396 and 91638204 and in part by the open research fund of National Mobile Communications Research Laboratory, Southeast University (No. 2018D08).

978-1-5386-4658-8/18/$31.00 ©2019 IEEE

2. RELATED WORK

Safe exploration of the Markov decision process (MDP) in [7] formulates the safety of a MDP based on its ergodicity and ensures the reachability of the initial states. The safe exploration as presented in [8] uses relative reachability between the states to measure and avoid side effects such as the disruption to the environments. [3] defines the exploration safety as an a priori unknown safety constraint of the state-action pair with a Gaussian process. [9] combines offline formal verification and runtime monitoring to improve the safety of reinforcement learning in adaptive cruise control. The learning-based control system proposed in [10] uses the Gaussian processes to approximate the error between the commanded acceleration and the actual acceleration of the system to ensure stability of the closed loop system and high-accuracy tracking of smooth trajectories.

Anti-jamming transmission is one of the first network security applications that use reinforcement learning techniques. For instance, the wireless communication system in [1] uses Q-learning to choose the frequency channel to resist jamming. The anti-jamming vehicular transmission system in [11] uses the policy hill climbing algorithm to optimize the relay policy of the unmanned aerial vehicles against jamming and interference. The anti-jamming communication system in [12] applies DQN to determine the communication policy without knowing the jamming and the channel model. However, the communication performance of these schemes is not good enough to be implemented in practical networks.

3. REINFORCEMENT LEARNING WITH SAFE EXPLORATION

We propose a safe reinforcement learning algorithm with safe exploration for safety critical applications. This algorithm evaluates the risk level of each state-action pair according to the security performance metrics and adjust the action selection policy based on the risk levels. We focus on the safe exploration of the discrete-time, finite-state and finite-action MDP, in which the next state only depends on the current state and action. Let k denote the time index that can be omitted if no confusion happen in the following content. The MDP is denoted by a tuple $\mathcal{M} = (\mathbf{S}, \mathbf{A}, \mathcal{P}, \mathcal{R})$, where

- $\mathbf{S}$ is a set of the possible states.
- $\mathbf{A}$ is a set of the feasible actions.
- $\mathcal{P} : \mathbf{S} \times \mathbf{A} \times \mathbf{S} \to [0, 1]$ represents the transition probabilities to another state by taking action a_k at s_k.
- $\mathcal{R} : \mathbf{S} \times \mathbf{A} \to \mathbb{R}$ is the reward function as performance metric of the agent that takes action a_k in state s_k.

A policy denoted by $\pi : \mathbf{S} \times \mathbf{A} \to [0, 1]$ represents the probability of taking action a_k in state s_k. The goal of the learning agent is to optimize its policy that maximises its long term expected rewards that can be estimated with the Q-value. In the classical Q-learning algorithm, the Q-values are updated via the Bellman iterative equation according to the reward r_k and its state transition given by:

$$\begin{aligned} Q\left(s_k, a_k\right) \leftarrow & (1-\alpha) Q\left(s_k, a_k\right) \\ & + \alpha \left(r_k + \gamma \max_{a \in \mathbf{A}} Q\left(s_{k+1}, a\right) \right), \end{aligned} \tag{1}$$

in which $\alpha \in [0, 1]$ is the learning rate weighs the current experience in the learning process and $\gamma \in (0, 1]$ denote the discount factor that represents the weight of the future reward.

Let $l(s_k, a_k)$ denote the risk level of taking action a_k in the state s_k. For simplicity, this algorithm assumes L risk levels, where risk level L is the most dangerous and the state-action pair is safe with zero risk level. The learning agent updates a blacklist denoted by $\mathcal{T} = \{(s_k, a_k) | l(s_k, a_k) = L\}$ to store the most dangerous state-action pairs.

Let ξ_i denote the criterion to measure the risk level i, the learning agent uses a criterion vector $\boldsymbol{\xi} = [\xi_i]_{1 \le i \le L}$ according to prior knowledge to identify whether a new explored state-action pair is safe or not. Note that there exists different methods to evaluate the risk levels for the agent. For simplicity, we use a predetermined criterion vector but our algorithm can be extended to other methods. Let $\mathrm{I}(\cdot)$ denote an indicator that equals to 0 if the argument is true and 1 otherwise. The risk level $l(s_k, a_k)$ depends on the reward r_k and the criterion vector $\boldsymbol{\xi}$ given by $l(s_k, a_k) = \sum_{i=0}^{L} \mathrm{I}\left(r_k > \xi_i\right)$.

In the MDP, taking an inappropriate action a_k in state s_k is likely to bring the agent into a sequence of risky state-action pairs although $l(s_k, a_k)$ is low. Therefore, the learning agent traces back the previous experienced state-action pairs to estimate the long-term risk levels of their previous decision. We use the discount factor γ to represent the degradation of influence of current decision to future rewards. The algorithm considers the long-term risk level denoted by $E(s, a)$ for λ steps given by

$$E\left(s_k, a_k\right) = \sum_{j=0}^{\lambda} \gamma^j l(s_{k+j}, a_{k+j}). \tag{2}$$

The action selection policy $\pi(s_k, a)$, i.e., the probability to select action $a \in \mathbf{A}$ in state s_k, is determined according to the risk level and Q-value of each state-action pair via a modified Boltzmann distribution [13] given by

$$\pi\left(s_k, a\right) = \frac{\exp\left(\frac{Q(s_k, a)}{E(s_k, a)+1}\right) \mathrm{I}\left(l(s_k, a) = L\right)}{\sum\limits_{a' \in \mathbf{A}} \exp\left(\frac{Q(s_k, a')}{E(s_k, a')+1}\right) \mathrm{I}\left(l(s_k, a') = L\right)}. \tag{3}$$

The denominator $E(s, a)$ in the exponent works as a temperature parameter that adjusts the randomness of the decisions. The action with higher Q-value and lower risk level will be selected with a higher probability and the action in the blacklist will be forbidden.

This algorithm uses transfer learning and exploits the prior network defense knowledge to initialize the Q-values and the risk levels. Let $(\hat{s}_i, \hat{a}_i, \hat{s}_{i+1}, \hat{r}_i)$ denote a transition in a similar network environment that shares a similar dynamics and reward function to current network security application. The learning agent samples η transition randomly from a transition set $\Omega = \{(\hat{s}_i, \hat{a}_i, \hat{s}_{i+1}, \hat{r}_i)\}_{1 \le i \le N}$ to update the Q-values and the risk levels via Eq. (1) and Eq. (2), respectively.

4. DEEP REINFORCEMENT LEARNING WITH SAFE EXPLORATION

We further propose a deep reinforcement learning algorithm with safe exploration named DRLSE. This algorithm uses a CNN like the DQN algorithm to estimate the Q-value of each action [14] and introduces another CNN named E-network to estimate the long-term risk levels of each state-action pair. Current state s_k is input with the previous W state-action pairs to the Q-network to estimate $Q(s_k, a)$ and the E-network to estimate $E(s_k, a)$, $\forall a \in \mathbf{A}$. The action a_k is then chosen according to the Eq. (3).

Similar to the RLSE algorithm, the agent evaluates the reward r_k, observes the next state s_{k+1} and evaluates the risk level of (s_k, a_k) based on the criterion vectior. This transition $e_k = (s_k, a_k, r_k, s_{k+1}, l(s_k, a_k))$ is stored in a memory pool $\mathcal{D}$. This algorithm uses the experience replay technique to update the CNN with a minibatch $\mathcal{B}$ sampled from $\mathcal{D}$ based on the stochastic gradient descent (SGD) algorithm. The Q-network weights $\boldsymbol{\theta}_Q$ are updated by minimizing the loss function given by

$$\mathcal{L}(\boldsymbol{\theta}_Q) = \mathbb{E}_{e_i \in \mathcal{B}} \left[\left(r_i + \gamma \max_{a \in \mathbf{A}} Q(s_{i+1}, a) - Q(s_i, a_i) \right)^2 \right]. \quad (4)$$

Meanwhile, the E-network weights $\boldsymbol{\theta}_E$ are updated by minimising the loss function between the estimate risk level and the target risk level given by

$$\mathcal{L}(\boldsymbol{\theta}_E) = \mathbb{E}_{e_i \in \mathcal{B}} \left[\left(\sum_{j=0}^{\lambda} \gamma^j l(s_{i+j}, a_{i+j}) - E(s_i, a_i) \right)^2 \right]. \quad (5)$$

Moreover, the Q-network and E-network can share the convolutional (Conv) layers to reduce computation as both the Conv layers of the two CNNs work as a feature extractor. The architecture of the designed network is illustrated in Fig. 1 as an example.

Transfer learning is also applied to initialize the CNN weights $\boldsymbol{\theta}_Q$ and $\boldsymbol{\theta}_E$. More specifically, the loss functions $\mathcal{L}(\boldsymbol{\theta}_Q)$ and $\mathcal{L}(\boldsymbol{\theta}_E)$ are calculated via Eq. (4) and Eq. (4), respectively, based on a minibatch $\hat{\mathcal{B}}$ sampled from Ω. The learning agent then updates the weights of the network based on the gradients of the loss functions.

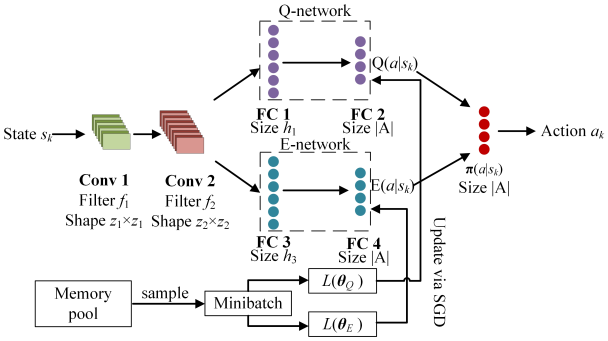


Fig. 1. Illustration of DRLSE.

5. CASE STUDY: SAFE EXPLORATION IN THE ANTI-JAMMING ROBOT COMMUNICATION

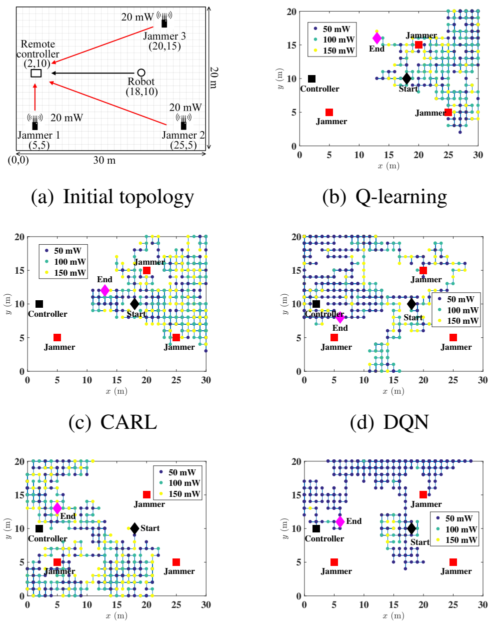
As a concrete example, we implement the reinforcement learning with safe exploration in the robot communication system to resist jamming attacks. The robot chooses the transmit power and the mobility policy to send messages such as the sensed images to the remote controller. By using smart radio devices, a jammer can sense the radio channel state and chooses its jamming power to degrade the ongoing communication performance.

In the reinforcement learning algorithm with safe exploration, the robot observes its current location vector denoted by LOC_k, estimates the bit error rate of the last message based on the feedback information from the remote controller BER_{k-1}, and formulates the current state as $\mathbf{s}_k = [LOC_k, BER_{k-1}]$. The robot determines its moving direction $\nu_k \in \{0; 1; 2; 3; 4\}$, which corresponding to staying in the previous location, and moving north, south, west and east, respectively. The robot moves according to ν_k and sends a message with transmit power $p_k \in [0, P_{max}]$ mW. The action vector $\mathbf{a}_k = [\nu_k, p_k]$ is chosen following the policy $\pi(\mathbf{a}|\mathbf{s}_k)$ that is given by Eq. (3) based on the Q-values $Q(\mathbf{s}_k, \mathbf{a})$ and the risk levels $E(\mathbf{s}_k, \mathbf{a})$.

Upon receiving the feedback of this message transmission from the remote controller, the robot evaluates the immediate reward bases on the BER of the signal and the transmission cost with $r_k = -BER_k - C_0 p_k - C_1 \mathrm{I}(\nu_k = 0)$, where C_0 and C_1 are the weights of the robot transmission and movement cost, respectively. We mainly concern about BER as a security issue in the anti-jamming communication, thus the robot compares the BER with a criterion vector $\boldsymbol{\xi} = \{0.1, 0.05, 0.03, 0.01\}$ to evaluate the risk level $l(\mathbf{s}_k, \mathbf{a}_k)$.

The BER larger than 0.1 will cause communication outage and the corresponding state-action pair is regarded as most dangerous and recorded in the blacklist $\mathcal{T}$. The robot updates the risk levels of the previous 5 state-action pairs via Eq. (2), with $\gamma = 0.5$.

The robot that supports deep learning can apply the DRLSE algorithm to further improve the jamming resis-



(a) Initial topology

(b) Q-learning

(c) CARL

(d) DQN

(e) RLSE

(f) DRLSE

Fig. 2. Illustration of the moving path and the transmit power.

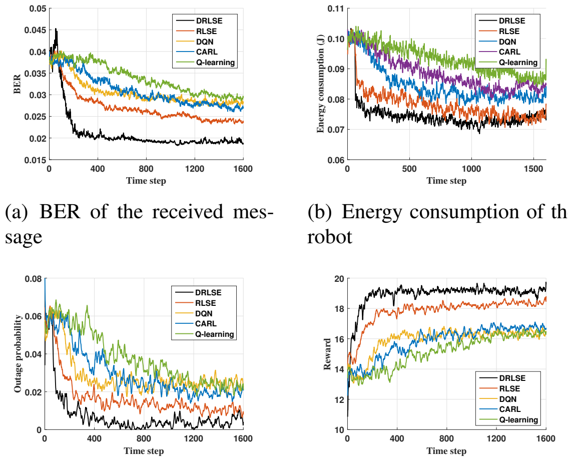
tance. Current state $\mathbf{s}_k$ is the input to the CNN to estimate the Q-values $Q(\mathbf{s}_k, \mathbf{a})$ over the Q-network and the risk levels $E(\mathbf{s}_k, \mathbf{a})$ over the E-network, $\forall \mathbf{a} \in \mathbf{A}$. We use a network with two one-dimension convolutional layers and two full-connected layers for each branch. The activation function is the Rectified Linear Units (ReLUs). The robot's anti-jamming communication experience at time step k, $e_k = (\mathbf{s}_k, \mathbf{a}_k, r_k, \mathbf{s}_{k+1}, l(\mathbf{s}_k, \mathbf{a}_k))$ is stored in the memory pool $\mathcal{D}$. At each time step, the robot randomly samples 32 experiences from the memory pool to formulate the minibatch $\mathcal{B}$. The loss functions of the Q-values and the risk levels are calculated according to $\mathcal{B}$ via Eq. (4) and (5), respectively. The weights of the Q-network and E-network, i. e., $\boldsymbol{\theta}_Q$ and $\boldsymbol{\theta}_E$, are updated via SGD based on the loss functions.

We record a transition set Ω from a similar communication scenario with different topology, in which another robot chooses its action randomly. At the beginning of the learning process, the robot applies the transfer learning to initialize both the two algorithms based on Ω with $\eta = 200$.

6. PERFORMANCE EVALUATION

We evaluate the performance of the proposed RL based robot communication scheme via the toy experiments in a 30 ×20 m^2 room against three jammers. Each jammer stayed in the same location and sent signals at 20 mW power as shown in Fig. 2(a). The robot initially stayed at $(18, 10)$ m and chose a direction to move 1 m at each time step to send the sensing report with power 50, 100 or 150 mW.

The moving path and the transmit power of the robot that applies the proposed safe exploration algorithms over the first 400 time steps are illustrated in Fig. 2. The result is compared with the classical Q-learning, DQN and CARL that uses the case-based reasoning to reduce random exploration [6]. The proposed RLSE enables the robot to find the "good" transmit location that saves the transmit power and is less impacted by the jammers faster than CARL that is in turn better than Q-learning. Compared with RLSE and DQN, the safe exploration with deep RL, DRLSE further saves the exploration and provides stronger jamming resistance with higher transmit power over the long-distance communication.



(a) BER of the received message

(b) Energy consumption of the robot

(c) Outage probability

(d) Reward of the robot

Fig. 3. Performance of the robot communication system.

The BER of the received message, the energy consumption of the robot, the outage probability that is the rate for the robot enters the most risky states and the reward of the robot over time averaged over 200 experiments are provided in Fig. 3. The proposed RLSE algorithm reduces the error rate of the message transmission, energy consumption and the outage probability, and increases the reward compared with Q-learning and CARL. The DRLSE algorithm provides the best anti-jamming communication performance for the robot that supports CNN due to the safe exploration.

7. CONCLUSION

In this paper, we have proposed a safe reinforcement learning algorithm with safe exploration and the deep reinforcement learning version for safety critical applications. This approach enables a learning agent to learn the risk levels of the state-action pairs via trial-and-error and reduce random explorations. A case study was performed for the robot communication against three jammers. Experiment results verify its performance gain including the communication quality, the outage probability and the reward compared with the benchmark algorithms.

8. REFERENCES

[1] Youngjune Gwon, Siamak Dastangoo, Carl Fossa, et al., "Competing mobile network game: Embracing antijamming and jamming strategies with reinforcement learning," in *Proceedings IEEE Conference on Communications and Network Security*, 2013, pp. 28–36. National Harbor, MD.

[2] Yuzhe Li, Daniel E. Quevedo, Subhrakanti Dey, et al., "SINR-based DoS attack on remote state estimation: A game-theoretic approach," *IEEE Transactions on Control of Network Systems*, vol. 4, no. 3, pp. 632 – 642, 2016.

[3] Matteo Turchetta, Felix Berkenkamp, and Andreas Krause, "Safe exploration in finite markov decision processes with gaussian processes," in *Advances in Neural Information Processing Systems*, 2016, pp. 4312–4320. Barcelona, Spain.

[4] Felix Berkenkamp, Matteo Turchetta, Angela Schoellig, et al., "Safe model-based reinforcement learning with stability guarantees," in *Advances in Neural Information Processing Systems*, 2017, pp. 908–918. Long Beach, CA.

[5] Romain Laroche and Merwan Barlier, "Transfer reinforcement learning with shared dynamics," in *AAAI Conference on Artificial Intelligence*, 2017, pp. 2147–2153. San Francisco, California.

[6] Manu Sharma, Michael P Holmes, Juan Carlos Santamaría, et al., "Transfer learning in real-time strategy games using hybrid CBR/RL," in *International Joint Conference on Artificial Intelligence*, 2007, pp. 1041–1046, Hyderabad, India.

[7] Teodor Mihai Moldovan and Pieter Abbeel, "Safe exploration in markov decision processes," in *International Conference on Machine Learning*, 2012, pp. 1711–1718. Edinburgh, UK.

[8] Victoria Krakovna, Laurent Orseau, Miljan Martic, and Shane Legg, "Measuring and avoiding side effects using relative reachability," *arXiv preprint arXiv:1806.01186*, 2018.

[9] Nathan Fulton and André Platzer, "Safe reinforcement learning via formal methods," in *AAAI Conference on Artificial Intelligence*, 2018, pp. in press. New Orleans, LA.

[10] Mohamed K Helwa, Adam Heins, and Angela P Schoellig, "Provably robust learning-based approach for high-accuracy tracking control of lagrangian systems," *arXiv preprint arXiv:1804.01031*, 2018.

[11] Liang Xiao, Xiaozhen Lu, Dongjin Xu, et al., "UAV relay in VANETs against smart jamming with reinforcement learning," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 5, pp. 4087–4097, 2018.

[12] Guoan Han, Liang Xiao, and H Vincent Poor, "Two-dimensional anti-jamming communication based on deep reinforcement learning," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2017, pp. 2087–2091. New Orleans, LA.

[13] Emilio Parisotto, Jimmy Lei Ba, and Ruslan Salakhutdinov, "Actor-mimic: Deep multitask and transfer reinforcement learning," in *International Conference on Learning Representations*, 2016, pp. 1–16. San Juan, Puerto Rico.

[14] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–541, 2015.