CONTROLLED SENSING AND ANOMALY DETECTION VIA SOFT ACTOR-CRITIC REINFORCEMENT LEARNING

Chen Zhong, M. Cenk Gursoy, and Senem Velipasalar

Department of Electrical Engineering and Computer Science, Syracuse University, Syracuse, NY 13244
Email: czhong03@syr.edu, mcgursoy@syr.edu, svelipas@syr.edu

ABSTRACT

To address the anomaly detection problem in the presence of noisy observations and to tackle the tuning and efficient exploration challenges that arise in deep reinforcement learning algorithms, we in this paper propose a soft actor-critic deep reinforcement learning framework. To evaluate the proposed framework, we measure its performance in terms of detection accuracy, stopping time, and the total number of samples needed for detection. Via simulation results, we demonstrate the performance when soft actor-critic algorithms are employed, and identify the impact of key parameters, such as the sensing cost, on the performance. In all results, we further provide comparisons between the performances of the proposed soft actor-critic and conventional actor-critic algorithms.

***Index Terms*—** Anomaly detection, controlled sensing, reinforcement learning, soft actor-critic algorithm.

1. INTRODUCTION

In a wide-range of applications involving, for instance, remote health monitoring [1], smart grid [2], assembly lines, structural health monitoring, autonomous systems [3], adaptive radar [4], cognitive radio networks [5], Internet of Things (IoT), one crucial task is to monitor a set of functionalities of stochastic systems *via sensors*, and make reliable and time-sensitive decisions and detect anomalies (e.g., in order to maintain safe operation, identify faulty or compromised components, detect targets or obstacles, avoid collisions, protect incumbent users).

Motivated by these, we consider active hypothesis testing for the anomaly detection problem in which there are k anomalous processes out of N processes, where $0 \leq k \leq N$. It is not known a priori which processes and how many of them are anomalous. The decision-making agent aims at detecting all the anomalous processes by sensing/probing the processes sequentially and receiving noisy observations until it reaches a desired confidence level. During the detection process, the decision maker can select any n out of N processes (where $0 < n \leq N$) to probe in each time slot. Probing all the processes in every time slot is the quickest approach to confirm the state of every process. However, such probing comes at a cost in practice due to the cost of sensing and communication performed during this activity. In this context, a fundamental problem is to detect the anomalous processes as quickly as possible with high confidence while incurring low cost. In particular, efficient sensor scheduling/probing and decision-making policies, which strike the optimal balance between such competing requirements, need to identified.

Such a problem can be formulated within the framework of active hypothesis testing with cost control. The active hypothesis testing problem was first studied in [6]. Recently, more complicated and practical anomaly detection problems have been addressed. For instance, it is assumed that the prior information on the hypotheses is not perfectly known to the decision maker in [7], and the distribution of the observations is not distinguishable under some of the experiments in [8] and [9]. Extensions have also been explored to seek for a stopping rule that can hold in general cases [10]. And similar to our purpose, authors in [11] jointly considered the detection errors and the switching costs.

More recently, machine learning-based methods have also been applied to address detection and hypothesis testing problems. For instance, in [12] and [13], the deep Q-network (DQN) has been employed, and in [14] an adversarial statistical learning method has been proposed. In [15], we investigated the application of deep actor-critic reinforcement learning to anomaly detection. These deep reinforcement learning algorithms can achieve very competitive performance levels when compared to the traditional methods. However, fine tuning and efficient exploration in these deep reinforcement learning algorithms and having them operate robustly in noisy environments are key challenges. To tackle these challenges, we in this work seek for a solution utilizing the recently proposed *soft actor-critic reinforcement learning algorithm* [16], which is based on the maximum entropy reinforcement learning framework. Additionally, as further differences from our previous work in [15], we consider Gaussian noise distorted observations (instead of observations distorted by a binary symmetric channel), both log-likelihood ratio and entropy based reward formulations, and potentially multiple probing of processes at a time.

2. SYSTEM MODEL

We consider N potentially statistically dependent processes. Each of these N processes is in either normal (denoted by 0) or anomalous (denoted by 1) state. Therefore, the states of the N processes can be denoted as a binary-valued random vector $\mathcal{S} \in \{0,1\}^N$. In this work, the goal is to detect all the anomalous processes, which is equivalent to estimating the random vector $\mathcal{S}$. It is also assumed that the states of all processes will remain the same until all anomalous processes are detected and fixed.

We assume that the states of the processes can be probed using a series of sensors. Ideally, the sensors report signal "0" if the selected processes are normal, and report signal "1" if the selected processes are anomalous. However, in practice, the sensor observations and/or communication links are noisy. Due to this, we consider a practical setting in which the decision-making agent receives observations distorted by additive Gaussian noise $\mathcal{N}(0,\sigma^2)$, with mean zero and variance σ^2. Hence, if we denote the observation of process i as Y_i, we have

$$Y_i = \begin{cases} y \sim \mathcal{N}(0,\sigma^2) & \text{if process } i \text{ is normal} \\ y \sim \mathcal{N}(1,\sigma^2) & \text{if process } i \text{ is anomalous} \end{cases}. \quad (1)$$

To estimate the random vector $\mathcal{S}$, the agent, which has no prior knowledge of number of anomalous processes and the statistical dependence between the processes, dynamically selects n out of the N processes to probe at each time, where $n = 1, 2, \ldots, N$. It is important to note that the number of processes to be probed, n, is potentially different at each time, depending on the sensor scheduling policy. Hence, sequential decision making is required for such controlled sensing. We impose a probing cost depending on the number of samples obtained by the agent in every time slot. And based on the noisy observations, the agent aims to minimize the time slots needed to reach a a decision with certain confidence level while also controlling the probing cost.

Considering N processes in total and an unknown number k anomalous processes, we have $M = 1 + \sum_{k=1}^{N} \binom{N}{k}$ hypotheses. We denote these hypotheses by H_m, $m = 0, 1, 2, \ldots, (M-1)$. Each hypothesis stands for a valid combination of processes that are anomalous. Moreover, we denote the prior probabilities of each hypothesis being true by the probability vector $\pi = [\pi_0, \ldots, \pi_{M-1}]$, which are joint probabilities of the N processes being in the corresponding states. With this, we further denote by π_m^t the posterior belief of the hypothesis H_m being true at time t, and update the posterior belief as

$$\pi_m^t = \frac{\pi_m \prod_{t'=1}^{t} f_m^{i'_t}(Y'_t)}{\sum_{l=0}^{M-1} \pi_l \prod_{t'=1}^{t} f_l^{i'_t}(Y'_t)} \quad (2)$$

where we denote the sensor selected at time t by i_t, and

$$f_m^{i_t}(Y_t) = \begin{cases} \mathcal{N}(1,\sigma^2;Y_t) & \text{if } i_t \in H_m \\ \mathcal{N}(0,\sigma^2;Y_t) & \text{if } i_t \notin H_m \end{cases}. \quad (3)$$

When the agent selects n processes in a time slot, the posterior probabilities will be updated n times.

3. PROBLEM FORMULATION

The hypothesis H_m is claimed to be accepted when the posterior belief π_m is greater than the upper bound π_{upper}, or to be rejected when the posterior belief is less than the lower bound π_{lower}. And once any of the M hypotheses is accepted, the observer stops receiving samples immediately. In this work, we consider two different reward functions.

Log-likehood Ratio Based Reward: Similarly as in [12] and [17], we consider the confidence level as the maximization objective. The confidence level of the posterior probability at time t is given by the average Bayesian log-likelihood ratio (LLR)

$$\mathcal{C}(\pi(t)) = \sum_{m=0}^{M-1} \pi_m \log \frac{\pi_m}{1-\pi_m}. \quad (4)$$

And the LLR-based reward is defined as $r_{\mathcal{C}}(t) = \mathcal{C}(\pi(t)) - \mathcal{C}(\pi(t-1))$.

Entropy Based Reward: Since the entropy of the posterior probability distribution is minimized by having one of the posterior probabilities to be 1 and all the other probabilities to be 0, we can also consider an entropy-based reward given as $r_{\mathcal{H}}(t) = \mathcal{H}(\pi(t-1)) - \mathcal{H}(\pi(t))$ where entropy is formulated as $\mathcal{H}(\pi(t)) = -\sum_{m=0}^{M-1} \pi_m \log \pi_m$.

Cost: As mentioned in the previous section, we consider a probing/sensing cost and incorporate it into the reward function (as described in Section 4 below). This instantaneous cost $c(t)$ depends on the number of processes that are selected to be probed in time slot t. We assume a sensing cost of λ per process, and define the cost at time t as $c(t) = \lambda |a_t|$ where a_t denotes the action selected in time t, and $|a_t|$ is the size of the corresponding action, quantifying the number processes/sensors selected to be probed at time t.

4. SOFT ACTOR-CRITIC FRAMEWORK

In this section, we describe the proposed soft actor-critic learning framework for the considered anomaly detection problem. We first introduce the relevant definitions within the framework.

Agent's Observation and State: Since the agent can only have observations from the selected sensors/processes, the problem can be modeled as a partially observable Markov decision process (POMDP). With this sample, the agent can

update the posterior belief π^t according to (2). We take the posterior belief vector as the state (or input) of the agent, and we denote the state at time t as $\mathcal{O}_t$, and define it as

$$\mathcal{O}_t = \begin{cases} \pi & t = 1 \\ \pi^{t-1} & \text{otherwise} \end{cases}. \tag{5}$$

Action: We denote the action space as $\mathcal{A}$, in which all valid actions are included. Here, the size of the action space is $M = \sum_{k=1}^{N} \binom{N}{k}$, and a valid action a stands for selecting the corresponding sensors and receiving the samples to update the posterior belief.

Reward: Since the decision-making agent aims to reach the confidence level as soon as possible, it should maximize the accumulated reward from the first time slot to the stopping time T_{stop} in an episode. So we define the immediate reward r_t as

$$r_t = \begin{cases} r_{\mathcal{C}}(t) - c(t) & \text{if LLR-based reward is employed} \\ r_{\mathcal{H}}(t) - c(t) & \text{if entropy-based reward is employed.} \end{cases} \tag{6}$$

Here, we define the state $\mathcal{O}_T$ as the terminal state if any of the M hypothesis is claimed to be accepted, i.e., $\max(\pi^{T-1}) \geq \pi_{\text{upper}}$. When we update the agent, we consider a weighted reward R_t at time $t \leq T$, as a discounted sum of the rewards $R_t = \sum_{\tau=t}^{T} \eta^{\tau-t} r_\tau$, so that each previous selection that can lead to better future steps will achieve a greater reward.

The soft actor-critic architecture consists of three neural networks: policy network, Q network, and value network. These three networks will not share any neurons but exchange information to update each other.

Policy network: The policy network is employed to explore a policy μ that maps the agent's observation $\mathcal{O}$ to the action space $\mathcal{A}$: $\mu_\phi(\mathcal{O}) : \mathcal{O} \to \mathcal{A}$. Since the action space is discrete, we use the softmax function at the output layer of the policy network so that we can obtain the scores of each action. The scores sum up to 1 and can be regarded as the probabilities of obtaining a good reward when the corresponding actions are chosen.

Q network: The Q network Q_θ, parameterized by θ, is an approximator to the soft Q function. It is fed the $(\mathcal{O}, a)$ pairs, and it estimates the corresponding Q value. The Q network encourages the policy to converge to the real Q value distribution instead of converging to a promising action. In this way, the agent tends to explore the environment more and engage in effective exploration strategies.

Value network: The value network $V_\psi(\mathcal{O})$ is parameterized by ψ, and it estimates the soft values of the given states. Since the estimated state value indicates the potential future reward, the value network encourages the policy to exploit the promising actions that are learned from the experience.

Update: To update the neural networks, we adopt a memory $\mathcal{D}$ to store the historical transitions, and sample a mini-batch at every iteration. And all three neural networks are updated using stochastic gradient descent.

The value network is updated by minimizing the squared residual error

$$J_{V_\psi} = \mathbb{E}_{\mathcal{O}_t \sim \mathcal{D}}[\frac{1}{2}(V_\psi(\mathcal{O}_t) - \mathbb{E}_{a_t \sim \mu_\phi}[Q_\theta(\mathcal{O}_t, a_t) - \log \mu_\phi(a_t|\mathcal{O}_t)])^2]. \tag{7}$$

The Q network is updated by minimizing the soft Bellman residual

$$J_{Q_\theta} = \mathbb{E}_{(\mathcal{O}_t, a_t) \sim \mathcal{D}}[\frac{1}{2}(Q_\theta(\mathcal{O}_t, a_t) - \hat{Q}_\theta(\mathcal{O}_t, a_t))^2] \tag{8}$$

where $\hat{Q}_\theta(\mathcal{O}_t, a_t) = r(\mathcal{O}_t, a_t) - c_t + \gamma \mathbb{E}[V_\psi(\mathcal{O}_{t+1})]$.

The policy network is trained by minimizing the expected KL-divergence

$$J_{\mu_\phi} = \mathbb{E}_{\mathcal{O}_t \sim \mathcal{D}}[D_{KL}(\mu_\phi(\cdot|\mathcal{O}_t)||\text{softmax}(Q_\theta(\mathcal{O}_t, \cdot)))]. \tag{9}$$

5. SIMULATION RESULTS

In the experiments, we set the total number of monitored processes as $N = 3$. Therefore, there will be 8 hypothesis and 7 valid actions for the decision-making agent.

Even though the proposed framework is applicable to any model and does not require any model information, we consider a setting with correlations between the three processes and set the probability of a process being normal to be $q = 0.8$ [18]. Specifically, we assume that the process indexed by 1 and 2 are dependent, and the process which is indexed by 3 is independent of the other two processes. We denote the correlation between processes 1 and 2 as $\rho \in [0, 1]$.

As noted before, the *soft actor-critic (SAC)* structure consists of three neural networks: policy network, Q network, and value network. Both the policy network and the value network consist of three layers, and the ReLU activation function is applied between each of the consecutive layers. In the Q network, there are four layers. Since both the observation and the action are taken as inputs of the Q network, the two components are loaded to the neural network through separate entries. Then, the extracted features of the observation and action will be merged in the second layer, and an estimated soft Q value will be given at the output layer.

For comparison purposes, we also implement the conventional actor-critic (AC) algorithm. The implementation of the AC framework is also explored in [19]. In the experiments, both the LLR-based reward and entropy-based reward will be considered for the actor-critic framework.

In the experiments, we consider three performance metrics: accuracy, stopping time, and total cost. The accuracy is defined as the ratio of the number of times that the agent declares the true hypothesis to the total number of detection

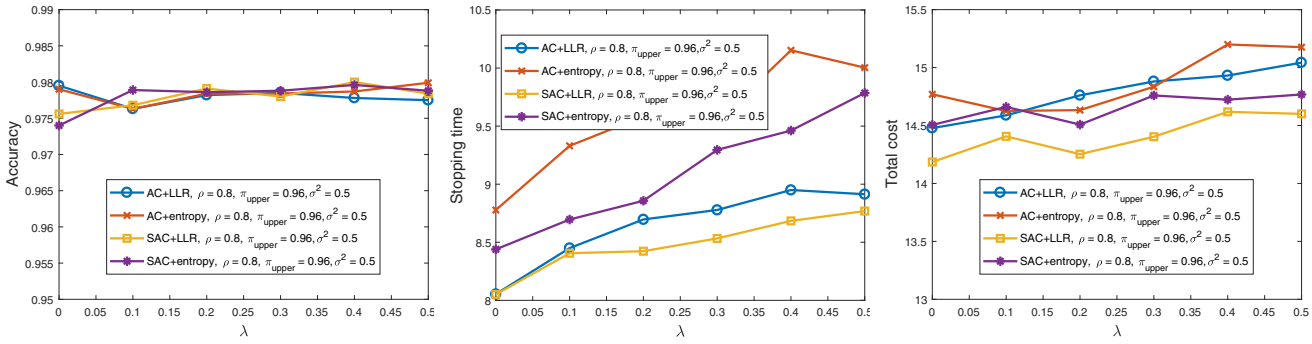

Fig. 1: Performance of the SAC algorithm and AC algorithm when $\lambda = 0, 0.1, 0.2, 0.3, 0.4, 0.5$.

results. The accuracy indicates the reliability of the detection. The stopping time is the number of time slots needed before the confidence reaches a desired threshold so that the agent can declare a hypothesis which is estimated to be true. In addition, the total cost is the summation of the number of selected sensors in each time slot in an episode, quantifying the total cost of sensing/probing.

In Fig. 1, we vary the value of cost per sensing as $\lambda = \{0, 0.1, 0.2, 0.3, 0.4, 0.5\}$ and observe the variations in the three performance metrics. In this experiment, we set $\rho = 0.8$, $\pi_{\text{upper}} = 0.96$, and $\sigma^2 = 0.5$. In Figure 1, it can be seen that the accuracy and total cost curves of the four frameworks (i.e., SAC with LLR reward, SAC with entropy reward, AC with LLR reward and AC with entropy reward) vary only over a small range, while the curves of stopping time have an obvious uptrend. This is because the variables ρ and σ influence the distribution of the random state vector $\mathcal{S}$ and the distribution of the observations, respectively. And the value of π_{upper} is the lowest confidence level to accept a hypothesis, since the agent will stop sampling when the threshold is reached. When we fix these three variables, the complexity of the detection process is mostly decided. Hence, when the value of λ changes, only the frameworks that are sensitive to the cost term in the reward function will be markedly impacted. Here, the algorithms with the entropy-based reward show relatively high sensitivity to the cost as the stopping times achieved by these two algorithms vary over a wider range. When λ is 0, all four frameworks tend to choose more processes in every time slot. As the value of λ increases, the cost term gradually takes a more dominant role in the reward function. To achieve lower cost, the numbers of processes selected by the four frameworks in every time slot tend to decrease. And for the cost-sensitive agents, the numbers of selected processes decrease more considerably than the other agents, which consequently leads to a more significant increase in the stopping time.

Overall, comparing the four frameworks, we observe that the algorithms with the entropy-based reward are more sensitive to the cost and can adjust the number of sensors to probe depending on the value of λ. As shown in the figures, the SAC algorithms perform better by achieving competitive or higher accuracy levels and lower total cost with a smaller stopping time. We further note that even though different algorithms have advantages and drawbacks depending on the setting and parameter values, the fine-tuning of the SAC algorithm is less complex than that of the AC algorithm.

6. CONCLUSION

In this work, we have proposed a soft actor-critic based reinforcement learning framework to solve the anomaly detection problem. First, we have formulated the anomaly detection problem as an active sequential hypothesis testing with noisy observations. To address this problem, we have designed a soft actor-critic algorithm with two different reward functions. Specifically, the reward function contains either the Bayesian log-likelihood ratio or the entropy as the reward and a cost term depending on the cost per sensing and the number of processes being sensed. To evaluate the performance, we have considered three performance metrics: accuracy, stopping time, and total cost. In the experiments, we have evaluated the performance as a function of the sensing cost. Additionally, we have compared the proposed framework with the conventional actor-critic algorithm. Via simulation results, we have demonstrated that the proposed soft actor-critic agent can attain smaller stopping times and operate competitively in terms of other performance metrics, and this agent explores effectively and more easily fine-tuned.

7. REFERENCES

[1] A. Bujnowski, J. Ruminski, A. Palinski, and J. Wtrorek, "Enhanced remote control providing medical functionalities," in *Proc. Inter. Conf. Pervasive Comput. Tech Healthc. Workshops*, pp. 290–293, May 2013.

[2] F. Passerini and A. M. Tonello, "Smart grid monitoring

using power line modems: Effect of anomalies on signal propagation," *IEEE Access*, vol. 7, pp. 27302–27312, 2019.

[3] F. Alotibi and M. Abdelhakim, "Anomaly detection for cooperative adaptive cruise control in autonomous vehicles using statistical learning and kinematic model," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–11, 2020.

[4] Z. Li, J. Xie, H. Zhang, H. Xiang, and Z. Zhang, "Adaptive sensor scheduling and resource allocation in netted collocated MIMO radar system for multi-target tracking," *IEEE Access*, vol. 8, pp. 109976–109988, 2020.

[5] Q. Zhao, L. Tong, A. Swami, and Y. Chen, "Decentralized cognitive MAC for opportunistic spectrum access in ad hoc networks: A POMDP framework," *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 3, pp. 589–600, 2007.

[6] H. Chernoff, "Sequential design of experiments," *The Annals of Mathematical Statistics*, vol. 30, no. 3, pp. 755–770, 1959.

[7] F. Cecchi and N. Hegde, "Adaptive active hypothesis testing under limited information," in *Advances in Neural Information Processing Systems*, pp. 4035–4043, 2017.

[8] K. Cohen and Q. Zhao, "Active hypothesis testing for anomaly detection," *IEEE Transactions on Information Theory*, vol. 61, no. 3, pp. 1432–1450, 2015.

[9] B. Huang, K. Cohen, and Q. Zhao, "Active anomaly detection in heterogeneous processes," *IEEE Transactions on Information Theory*, vol. 65, no. 4, pp. 2284–2301, 2019.

[10] M. R. Leonard and A. M. Zoubir, "Robust sequential detection in distributed sensor networks," *IEEE Transactions on Signal Processing*, vol. 66, no. 21, pp. 5648–5662, 2018.

[11] D. Chen, Q. Huang, H. Feng, Q. Zhao, and B. Hu, "Active anomaly detection with switching cost," in *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5346–5350, IEEE, 2019.

[12] D. Kartik, E. Sabir, U. Mitra, and P. Natarajan, "Policy design for active sequential hypothesis testing using deep learning," in *2018 56th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 741–748, IEEE, 2018.

[13] A. Puzanov and K. Cohen, "Deep reinforcement one-shot learning for change point detection," in *2018 56th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 1047–1051, IEEE, 2018.

[14] N. Moustafa, K. R. Choo, I. Radwan, and S. Camtepe, "Outlier dirichlet mixture mechanism: Adversarial statistical learning for anomaly detection in the fog," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 8, pp. 1975–1987, 2019.

[15] C. Zhong, M. C. Gursoy, and S. Velipasalar, "Deep actor-critic reinforcement learning for anomaly detection," in *2019 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, 2019.

[16] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," *arXiv preprint arXiv:1801.01290*, 2018.

[17] M. Naghshvar, *Active learning and hypothesis testing*. PhD thesis, UC San Diego, 2013.

[18] G. Joseph, C. Zhong, M. C. Gursoy, P. K. Varshney, and S. Velipasalar, "Anomaly detection under controlled sensing via active inference," *to appear in IEEE Global Communications Conference (GLOBECOM)*, 2020.

[19] G. Joseph, M. C. Gursoy, and P. K. Varshney, "Anomaly detection under controlled sensing using actor-critic reinforcement learning," in *2020 IEEE 21st International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pp. 1–5, 2020.