Deep reinforcement learning for active hypothesis testing with heterogeneous agents and cost constraints Supplementary material

George Stamatelis, Nicholas Kalouptsidis

In this supplement, we consider continuous observations as well observations from finite sets but originating from an almost real world environment. The number of processes and the communication graphs remain the same with those of the main body. We carry out performance comparisons in the unconstrained case with 25000 training episodes. We evaluate the decentralised PPO with a joint reward, the decentralised PPO with global critic and the centralised PPO using the same environment as in the main text. The single PPO agent with access to all sensors is taken as benchmark.

I. Gaussian data

A sensor model often considered in the literature (see for instance [1]) is the Gaussian model

$$y_t \sim \begin{cases} N(0,\sigma^2), & \text{if the process is normal} \\ N(1,\sigma^2), & \text{otherwise} \end{cases} \tag{1}$$

Similarly to [1] we set $\sigma = 0.5$. The results are summarized in tables I, II, III.

TABLE I: First decentralised scenario, accuracy and average stopping time, 10000 episodes, Gaussian observations

(a) accuracy

algorithm	$\epsilon = 0.05$	$\epsilon = 0.1$
single Agent PPO	0.979	0.973
centralised PPO	0.962	0.955
decentralised PPO with joint reward	0.951	0.937
PPO with global critic	0.969	0.963

(b) average stopping time

algorithm	$\epsilon = 0.05$	$\epsilon = 0.1$
single Agent PPO	17.371	14.349
centralised PPO	8.141	7.789
decentralised PPO with joint reward	7.673	6.961
PPO with global critic	7.24	6.227

TABLE II: Second decentralised scenario, accuracy and average stopping time, 10000 episodes, Gaussian observations

(a) accuracy

algorithm	$\epsilon = 0.05$	$\epsilon = 0.1$
single Agent PPO	0.979	0.973
centralised PPO	0.962	0.955
decentralised PPO with joint reward	0.978-1.0-1.0	0.978-0.999-0.999
PPO with global critic	0.985-0.999-0.999	0.974-0.982-0.982

(b) average stopping time

algorithm	$\epsilon = 0.05$	$\epsilon = 0.1$
single Agent PPO	17.371	14.349
centralised PPO	8.141	7.789
decentralised PPO with joint reward	12.211	10.8
PPO with global critic	9.329	8.43

TABLE III: Third decentralised scenario, accuracy and average stopping time, 10000 episodes, Gaussian observations

(a) accuracy

algorithm	$\epsilon = 0.05$	$\epsilon = 0.1$
single Agent PPO	0.979	0.973
centralised PPO	0.962	0.955
decentralised PPO with joint reward	0.977-0.999-0.999	0.967-0.981-0.981
PPO with global critic	0.991-0.994-0.994	0.974-0.98-0.98

(b) average stopping time

algorithm	$\epsilon = 0.05$	$\epsilon = 0.1$
single Agent PPO	17.371	14.349
centralised PPO	8.141	7.789
decentralised PPO with joint reward	12.392	11.283
PPO with global critic	10.526	9.392

The main takeaways of these experiments are the following

1) The decentralised PPO with joint reward and the PPO with global critic perform better than the centralised algorithm in the fully connected setting.
2) The decentralised PPO with global critic performs slightly better than the fully decentralised algorithm in all communication graphs.
3) All multiagent algorithms achieve significantly shorter stopping times compared to the single agent.

II. ToN IOT windows10 data

Next we evaluate the algorithms in an example closer to reality, using an cybersecurity dataset.

We use the windows 10 dataset from the TON_IOT data [2] [3] [4] [5] [6] [7] [8] [9]. Each row of the dataset contains information about one process. It includes information about the processor activity, the network activity, the memory activity, the disk activity e.t.c. Each process has a binary label about whether it is abnormal or not and a second label describing the type of attack, which we ignore. We selected the following five features

1) 'Memory Demand Zero Faults sec'
2) 'Network_I(Intel R _82574L_GNC)Bytes Received sec'
3) 'Network_I(Intel R _82574L_GNC) Bytes Sent sec'
4) 'Process_Page Faults_sec'
5) 'Memory Page Writes sec'

Using these features we train a decision tree that performs intrusion classification. We used the sklearn [10] implementation of decision trees setting class_weight='balanced' and a standard scaler.

We split the windows 10 dataset in three sets a training set (70% of the data), a validation set (10%of the data), and a test set. The model is trained on the training set, and the conditional observation probabilities are estimated on the validation set. They are then used to train the DRL algorithms [1].

The testing environment works as follows. First, the true hypothesis is sampled from the uniform prior. Then, at each time, depending on whether the queried process is normal or not we sample an entry from the test set and pass it from the trained decision tree classifier. The output of the classifier (0 or 1) is used to update the belief. The other features of the environment remain unchanged.

A. Results

TABLE IV: First decentralised scenario, accuracy and average stopping time, 10000 episodes, TON_IOT observations

(a) accuracy

algorithm	$\epsilon = 0.05$	$\epsilon = 0.1$
single Agent PPO	0.998	0.989
centralised PPO	0.995	0.998
decentralised PPO with joint reward	0.999	0.992
PPO with global critic	0.999	0.998

(b) average stopping time

algorithm	$\epsilon = 0.05$	$\epsilon = 0.1$
single Agent PPO	14.951	11.878
centralised PPO	6.399	7.611
decentralised PPO with joint reward	6.418	7.172
PPO with global critic	5.88	6.001

[1]It turns out that $P[Y = 1|\text{process is abnormal}] = 0.842$ and $P[Y = 1|\text{process is normal}] = 0.058$.

TABLE V: Second decentralised scenario, accuracy and average stopping time, 10000 episodes, TON_IOT observations

(a) accuracy

algorithm	$\epsilon = 0.05$	$\epsilon = 0.1$
single Agent PPO	0.999	0.992
centralised PPO	0.999	0.995
decentralised PPO with joint reward	0.999-1.0-1.0	0.995-0.996-0.996
PPO with global critic	0.999 -1.0- 1.0	0.98-0.999-0.999

(b) average stopping time

algorithm	$\epsilon = 0.05$	$\epsilon = 0.1$
single Agent PPO	14.951	11.878
centralised PPO	6.399	7.611
decentralised PPO with joint reward	10.073	8.323
PPO with global critic	9.008	7.611

TABLE VI: Third decentralised scenario, accuracy and average stopping time, 10000 episodes, TON_IOT observations

(a) accuracy

algorithm	$\epsilon = 0.05$	$\epsilon = 0.1$
single Agent PPO	0.999	0.992
centralised PPO	0.999	0.995
decentralised PPO with joint reward	0.999-1.0-1.0	0.994-0.996-0.996
PPO with global critic	1.0-1.0-1.0	0.992-0.995-0.995

(b) average stopping time

algorithm	$\epsilon = 0.05$	$\epsilon = 0.1$
single Agent PPO	14.951	11.878
centralised PPO	6.399	7.611
decentralised PPO with joint reward	10.897	9.458
PPO with global critic	9.704	8.878

The main takeaways are

1) Decentralised PPO with global critic achieves a shorter stopping time than its fully decentralised version.
2) For the fully connected setting both decentralised algorithms achieve a shorter stopping time than the global controller.
3) All multiagent algorithms are faster than the single agent algorithm.

We reach more or less the same conclusions for all three observation models considered.

III. Global critic Architecture

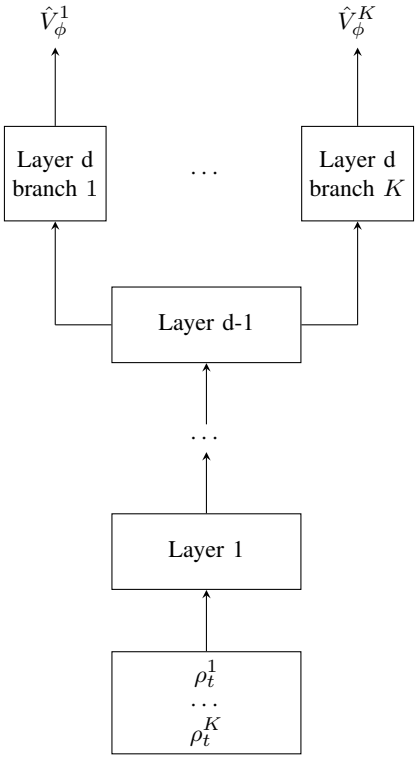

Fig. 1: A global critic network with d layers for the decentralised PPO with global critic.

References

[1] C. Zhong, M. C. Gursoy, and S. Velipasalar, "Controlled sensing and anomaly detection via soft actor-critic reinforcement learning," in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4198–4202, 2022.

[2] N. Moustafa, "A new distributed architecture for evaluating ai-based security systems at the edge: Network ton_iot datasets," vol. 72, 05 2021.

[3] A. Alsaedi, N. Moustafa, Z. Tari, A. Mahmood, and A. Anwar, "Ton_iot telemetry dataset: A new generation dataset of iot and iiot for data-driven intrusion detection systems," *IEEE Access*, vol. 8, pp. 165130–165150, 2020.

[4] N. Moustafa, M. Ahmed, and S. Ahmed, "Data analytics-enabled intrusion detection: Evaluations of ton_iot linux datasets," in *2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, pp. 727–735, 2020.

[5] N. Moustafa, "A systemic iot-fog-cloud architecture for big-data analytics and cyber security systems: A review of fog computing," 2019.

[6] N. Moustafa, M. Keshk, E. Debie, and H. Janicke, "Federated ton_iot windows datasets for evaluating ai-based security applications," 2020.

[7] T. Booij, I. Chiscop, E. Meeuwissen, N. Moustafa, and F. den Hartog, "Ton_iot: The role of heterogeneity and the need for standardization of features and attack types in iot network intrusion data sets," *IEEE Internet of Things Journal*, vol. PP, pp. 1–1, 05 2021.

[8] J. Ashraf, M. Keshk, N. Moustafa, M. Abdel-Basset, H. Khurshid, A. D. Bakhshi, and R. R. Mostafa, "Iotbot-ids: A novel statistical learning-enabled botnet detection framework for protecting networks of smart cities," *Sustainable Cities and Society*, vol. 72, p. 103041, 2021.

[9] N. Moustafa, "New generations of internet of things datasets for cybersecurity applications based machine learning: Ton_iot datasets," *Proceedings of the eResearch Australasia Conference, Brisbane, Australia.*, 2019.

[10] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.