# Out-of-Distribution Detection for Neurosymbolic Autonomous Cyber Agents

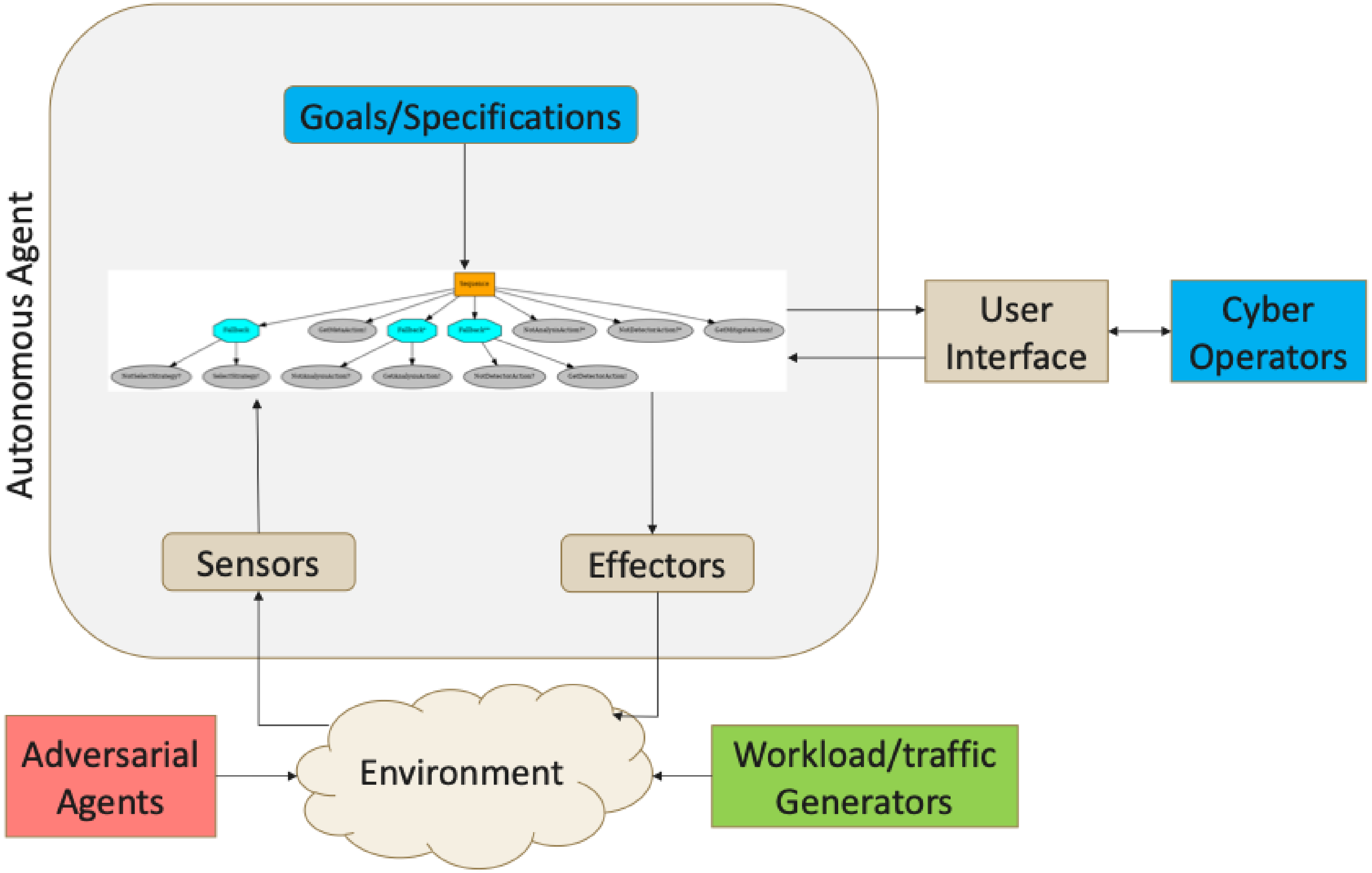
**Ankita Samaddar, Nicholas Potteiger, Xenofon Koutsoukos**

**Department of Computer Science**

VANDERBILT
UNIVERSITY

# Neurosymbolic Autonomous Cyber Agents

- Neurosymbolic autonomous agents are intelligent AI-based agents that can learn, reason about and solve problems.
- They use a mix of standard and learning enabled components (LECs) that are function approximators with a reinforcement learning (RL) policy, so that they can take optimal actions to effectively mitigate dynamic complex attacks



# Challenges in Autonomous Cyber Agents

***Uncertainties*** due to ***limited knowledge*** about the ***runtime behavior*** of the operational system and environment during designing and training of the autonomous agents

Significant ***challenge*** in characterizing the ***trustworthiness*** of the autonomous agents

The ***consequences*** can propagate deep into the system

***Impact*** system behaviors at all levels

***Thus, anomaly or out-of-distribution (OOD) detection methods need to be incorporated to identify information that is nonconformal with the environment used in training***

# Related Work

| Related Work | Description | Drawback |
|---|---|---|
| [1] | OOD behavior detection in vehicle controller using variational autoencoders and deep support vector data description | Do not focus on OOD detection scenarios for RL agent based autonomous systems |
| [2] | OOD detection using β-variational autoencoder with partially disentangled latent space | |
| [3] | OOD detection using Probably Approximately Correct Bayes framework in a robotic environment with guaranteed bounds | |
| [4], [5] | OOD detection using frameworks to detect semantic and covariate shifts | |
| [6], [7], [8] | OOD detection for RL-based agents | Do not consider discrete state space |

**Motivation:** Develop an ***OOD Monitoring algorithm*** that can ***detect OOD situations*** in autonomous system with ***discrete states and discrete actions*** to assure safety at runtime

# Autonomous Agents for Cyber Defense

Designed an autonomous agent for cyber-defense from a partially observable pursuit evasion game using genetic programming [9]

**CybORG CAGE Challenge Scenario 2**

Interface to evaluate the ***attacker (red agent)*** and the ***defender (blue agent)***

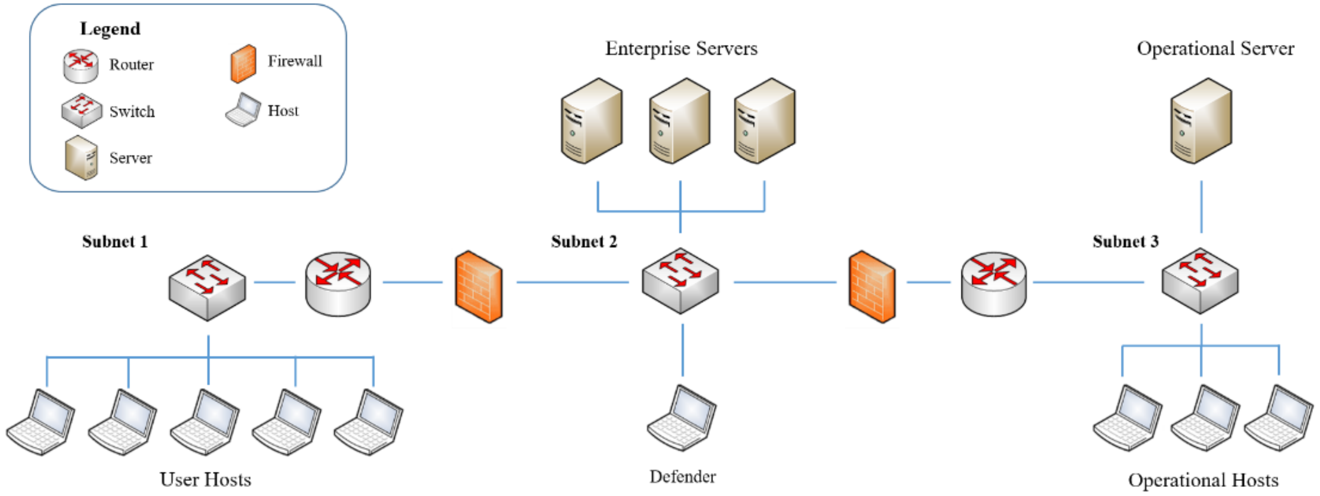
***Red agent :***

- Initial access to one of the user hosts in Subnet 1
- Scan hosts and subnets, exploit hosts, perform privilege escalation

**Objective:** Exploit the operational server through "Impact" action

***Blue agent:***

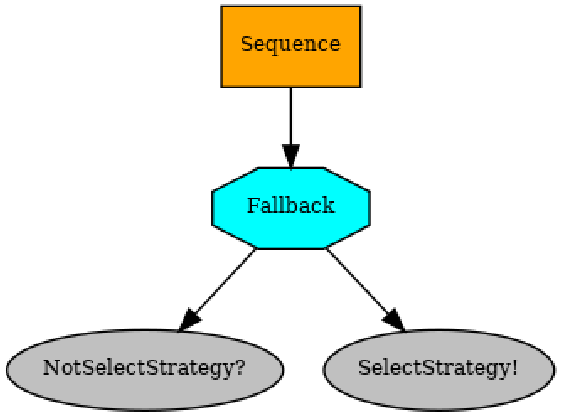
Mitigate red actions through Monitor, Analyze, Deploy Decoys, Remove and Restore actions



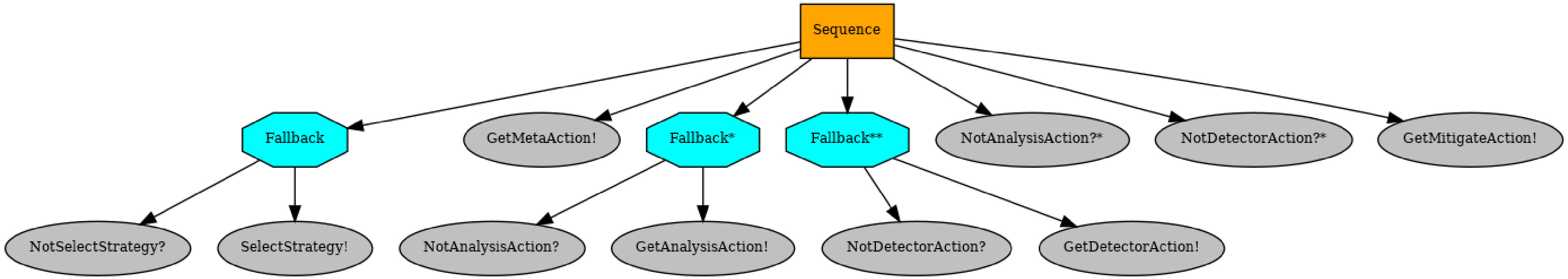
# Evolving Behavior Trees (EBT) based Autonomous Cyber-Defense Agent

## Behavior Trees

- Symbolic structure in our autonomous cyber-defense agent
- Provides high level control and reactive switching to adapt to new environments
- Modular in nature allowing seamless integration of new behaviors



## EBT based autonomous cyber-defense agent



## Cyber BT behaviors

1. SelectStrategy!
2. GetMetaAction!
3. GetDetectorAction!
4. GetMitigateAction!
5. GetAnalysisAction!

# System Model

Our system can be represented as a **discrete-time Partially Observable Markov Decision Process**

**M = (S, A, T, R, μ0)**

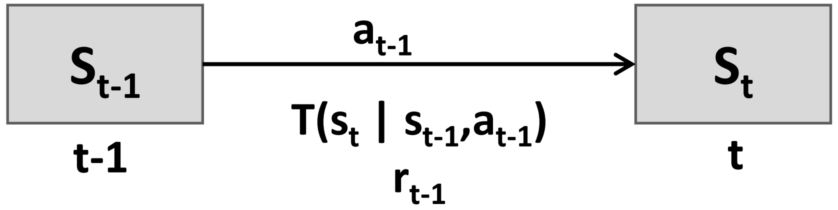
**S** : set of discrete and partially observable states

**A** : set of defender (blue agent) discrete actions

**T** : conditional transition probabilities

**R ( S x A x S → R ) :** the reward function

**μ0 ( s0, a0 )** : initial state and action



**Objective :** Select blue agent actions at each timestep so that the cumulative rewards **maximize** over time, i.e., $\sum_{t=1}^{t=\infty} \boldsymbol{r}_{t\,-\,1}$

# Problem Statement

*Given a network consisting of hosts, enterprise servers and operational servers and a neurosymbolic cyber-agent trained with a policy* ***π****, our objective is to develop a* ***safety assurance algorithm*** *to detect shifts from the distribution used for training.*

We address two key questions.

1. *Can we assure safety if the system transitions to any state s' such that* $\mathbf{Pr((s,a) \rightarrow s') < \rho}$ ***(Transition Probability Threshold)*** *in our training distribution?*
2. *Can we assure safety if the* ***red agent switches*** *to a* ***different strategy*** *than the one used for training?*

# Out-of-Distribution (OOD) Monitoring Algorithm

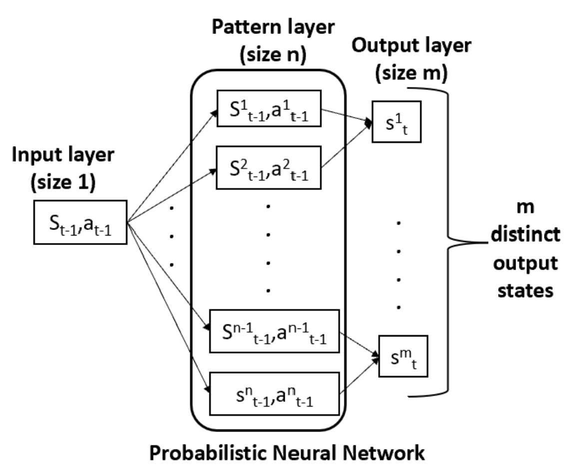
**OOD Monitoring Algorithm (Blue agent policy π, Transition Probability Threshold ρ)**

**Data Generation Phase**

- Collect transitions $(s_{t-1}, a_{t-1}) \rightarrow s_t$ for **τ** timesteps, (τ is very large), over multiple episodes (say **N** ) to generate the training data $\mathbf{D}_{\mathbf{train}}$

**Training Phase**

- Develop a **Probabilistic Neural Network (PNN)** following $(s_{t-1}, a_{t-1}) \rightarrow s_t$ for policy **π** over $\mathbf{D}_{\mathbf{train}}$



**OOD Monitoring Phase**

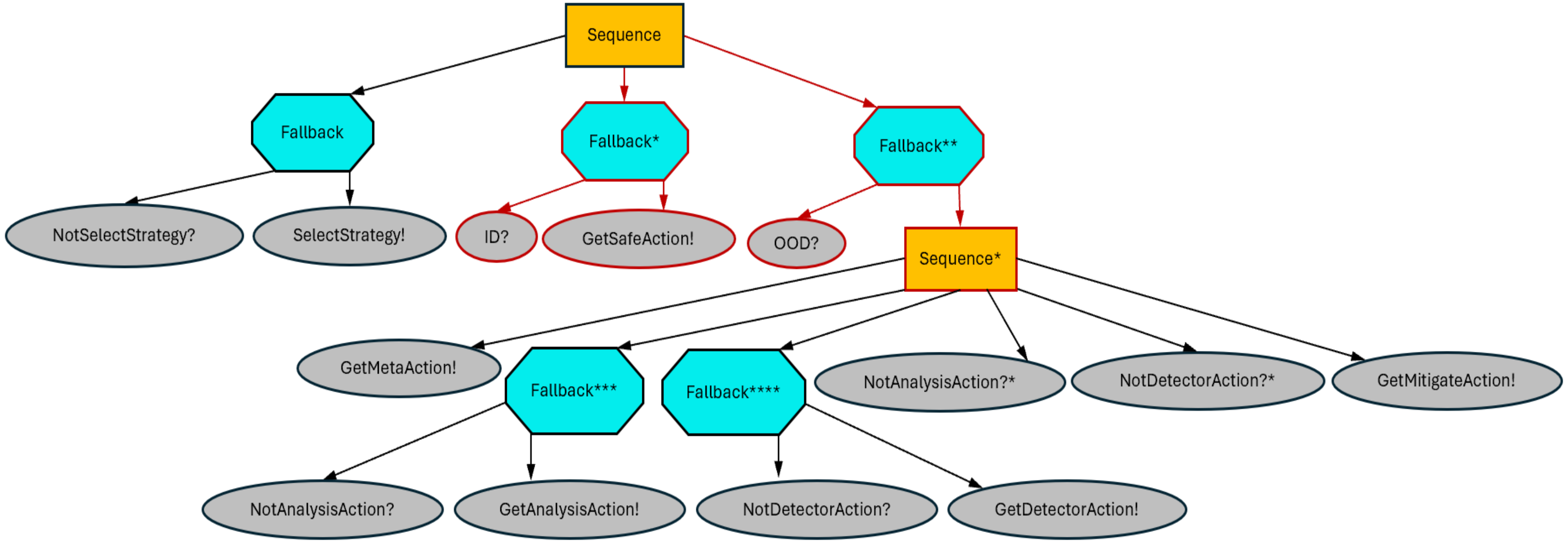
- $\mathbf{s_t}$ = Current state at timestep t on executing $\mathbf{a_{t-1}}$ on system state $\mathbf{s_{t-1}}$
- $\{s^1_t, s^2_t, \ldots s^k_t\}$ = set of **k** predicted current states from PNN
- If $\mathbf{s_t} \in \{s^1_t, s^2_t, \ldots s^k_t\}$ and $\mathbf{Pr((s_{t-1}, a_{t-1}) \rightarrow s_t) > \rho}$, then $\mathbf{s_t}$ is In-Distribution
- Else $\mathbf{s_t}$ is Out-of-Distribution

# Integration of OOD Monitoring Behavior in EBT

1. **ID? :** Determines if current state $s_t$ is In-Distribution

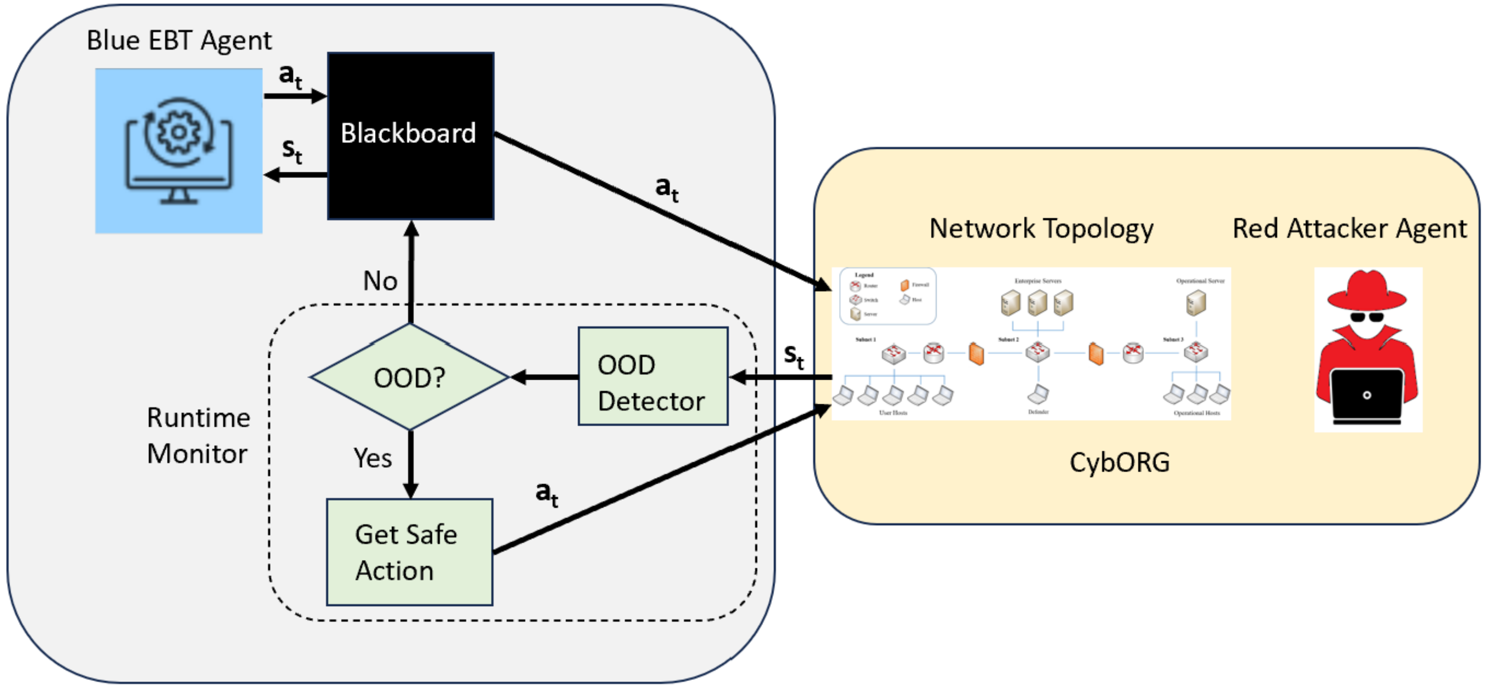
2. **GetSafeAction! :** Executes *Restore* action to restore the affected host/server to a previously known *"safe"* state, to assure safety

3. **OOD? :** Returns Failure if current state $s_t$ is In-Distribution to ensure normal execution of the system



# Experiments

- Cyber-Architecture consists of EBT-based autonomous cyber-defense agent, OOD Monitoring algorithm and CybORG CAGE Challenge Scenario 2
- Initialize a blackboard as the communication interface between the EBT and the simulator
- Perform experiments with two red agent strategies, *Meander* and *B_line*
- Generate $D_{train}$ for each of these agents over 10,000 episodes each with 100 steps to train the PNN



Blue EBT Agent

$a_t$

$s_t$

Blackboard

$a_t$

No

Runtime Monitor

OOD?

OOD Detector

$s_t$

Yes

Get Safe Action

$a_t$

Network Topology

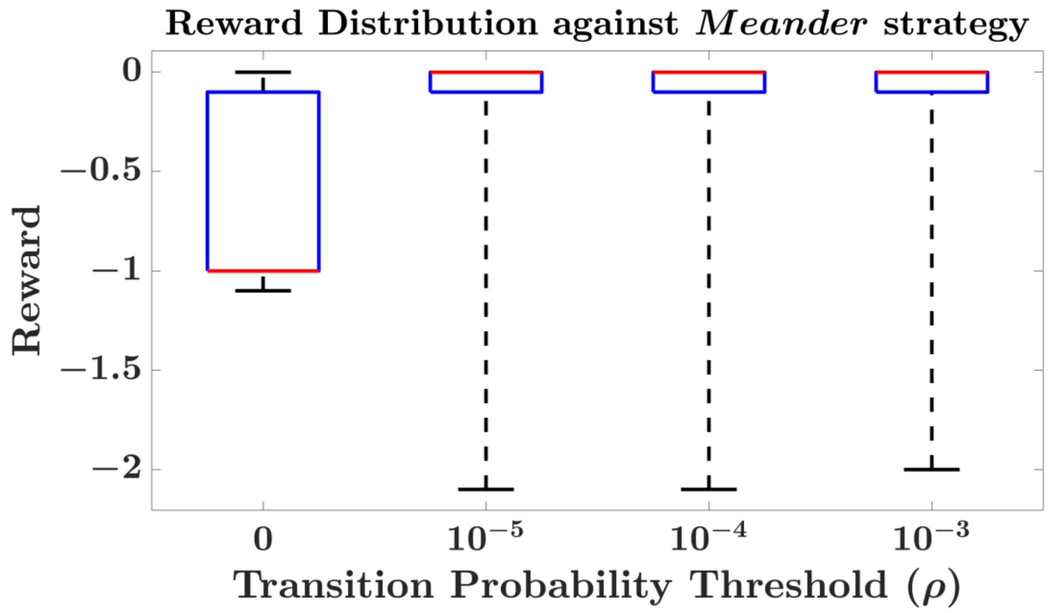
Red Attacker Agent

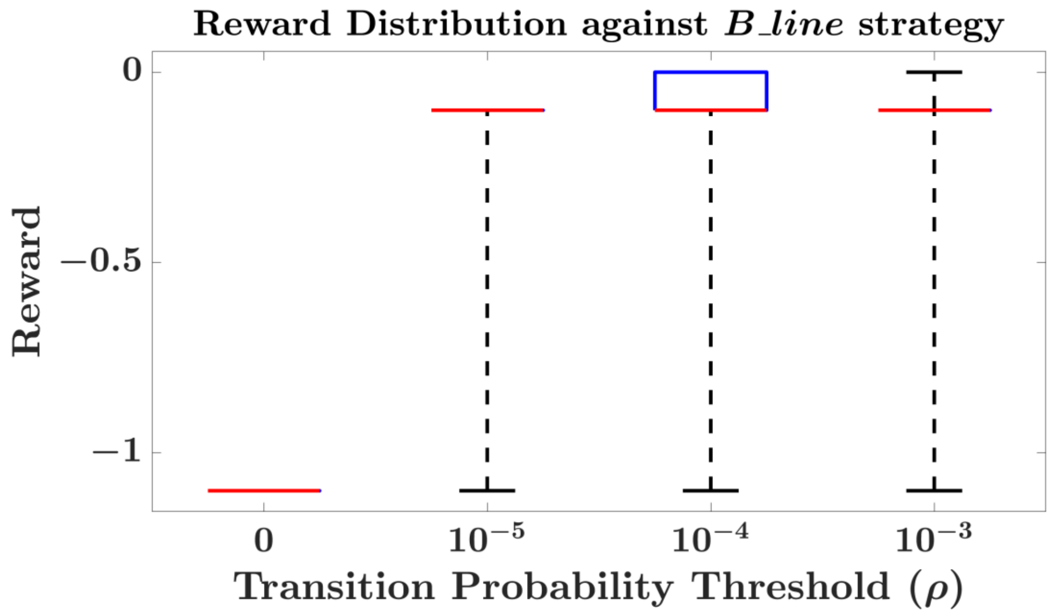
CybORG

# Results

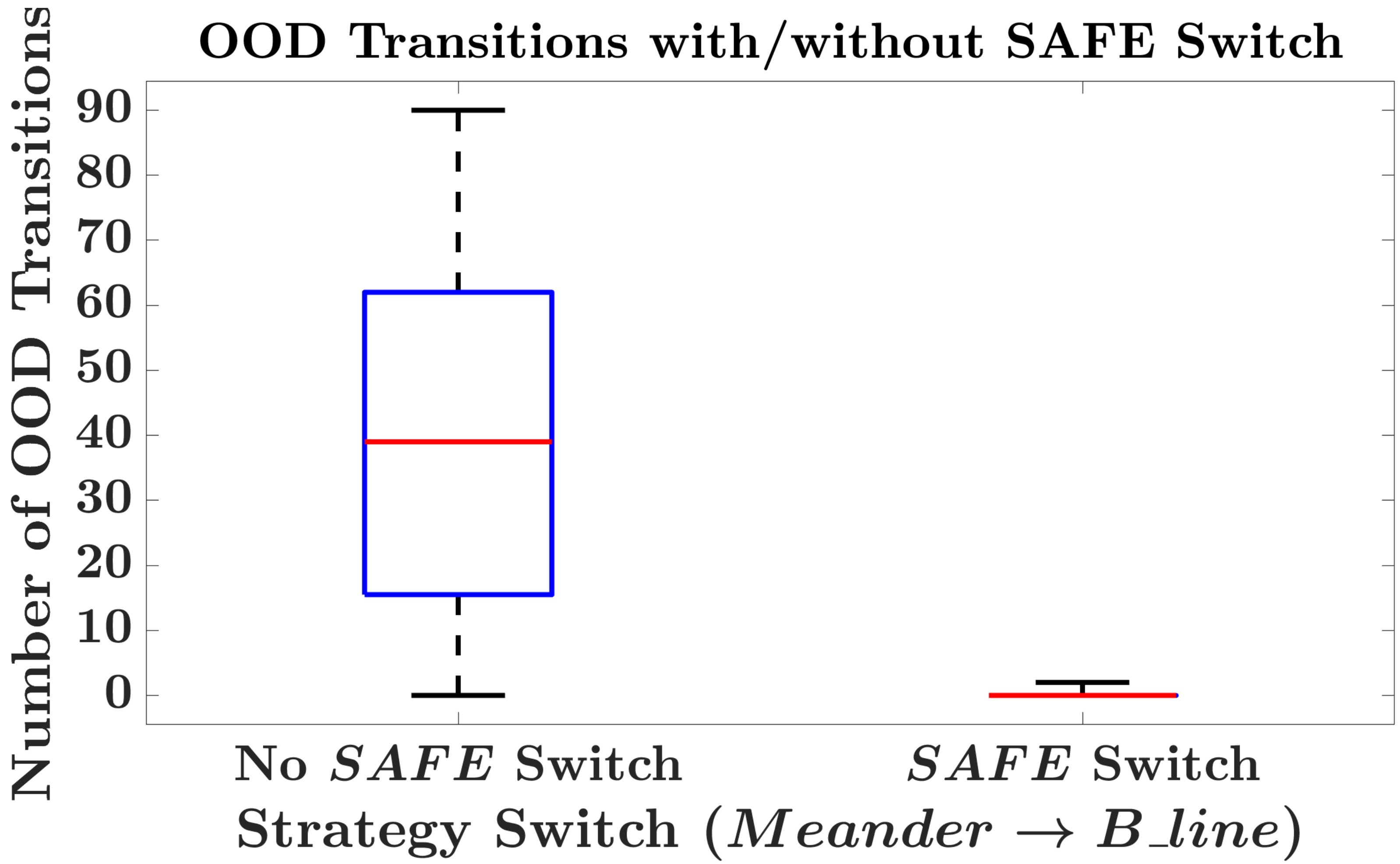
| Red Agent Strategy | Transition Probability Threshold (ρ) | Number of OOD Episodes (out of 1000) |
|---|---|---|
| ***Meander*** | **0** | **15** |
| | **$10^{-5}$** | **1000** |
| | **$10^{-4}$** | **1000** |
| | **$10^{-3}$** | **1000** |
| ***B_line*** | **0** | **1** |
| | **$10^{-5}$** | **782** |
| | **$10^{-4}$** | **1000** |
| | **$10^{-3}$** | **1000** |

With increase in ρ, we observe more probable transitions that are known to the system, causing less reward penalties



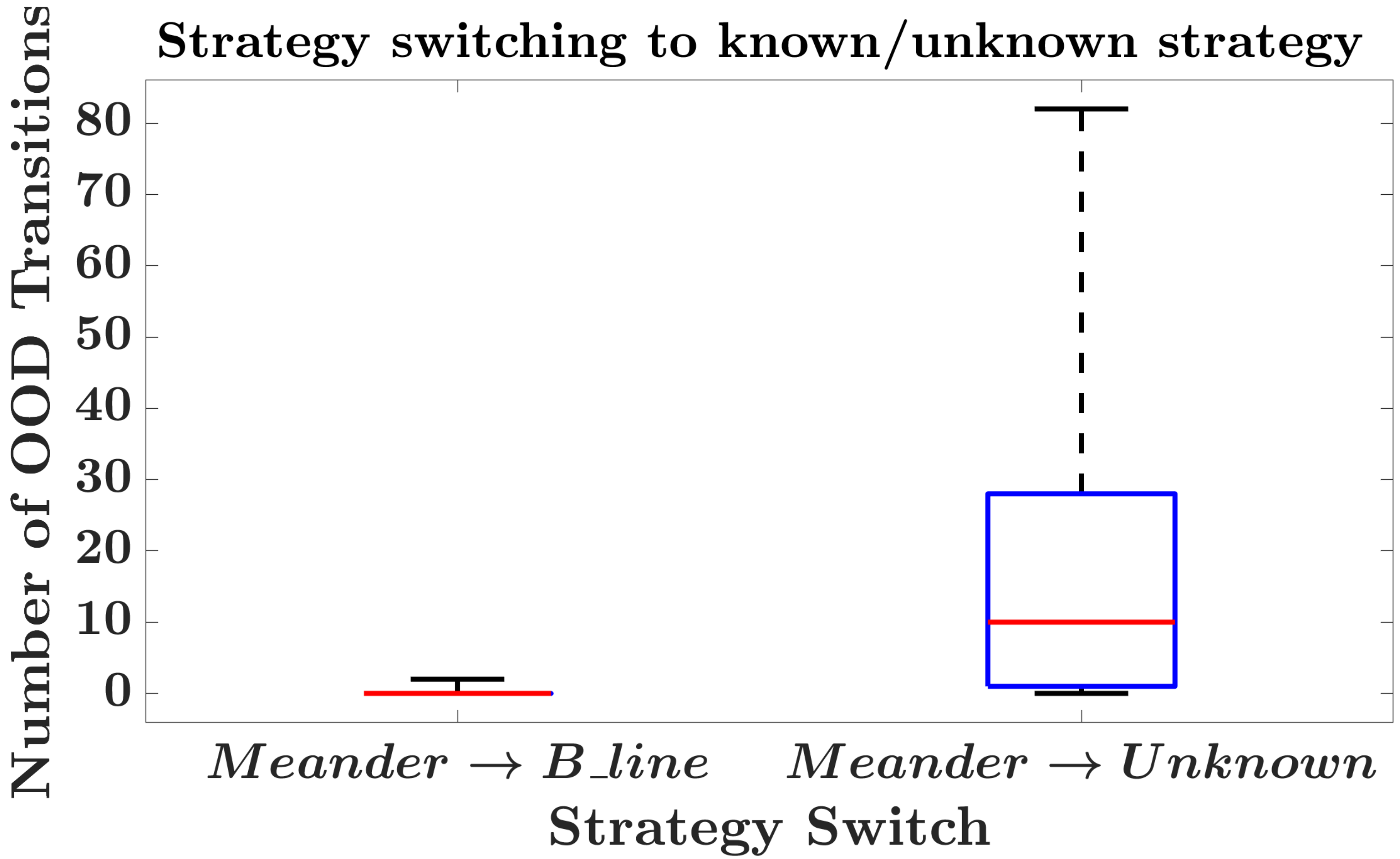


# Results



GetSafeAction! behavior in the EBT significantly reduces the number of OOD transitions by restoring the system to a “*safe*” state

# Results



*Total number of OOD transitions is significantly small when the blue agent knows the strategy*

# Conclusion

- Uncertainties in the runtime behavior of neurosymbolic cyber agents pose significant challenges in designing trustworthy agents
- Propose an OOD Monitoring algorithm to detect OOD situations for any RL-based agent with discrete states and actions
- Evaluate our approach on a neurosymbolic autonomous cyber-defense agent
- Perform experiments on a complex network simulator, the CybORG CAGE Challenge Scenario 2

# Future Works

- Evaluate our adversarial strategy on a real testbed to determine system dynamics at runtime

- Online learning techniques that can adapt and learn new adversarial movements to mitigate adversarial attacks on autonomous networks at runtime

# References

[1] F. Cai and X. Koutsoukos, “Real-time out-of-distribution detection in learning-enabled cyber-physical systems,” in 2020 ACM/IEEE 11thInternational Conference on Cyber-Physical Systems (ICCPS), pp. 174–183, 2020.

[2] S. Ramakrishna, Z. Rahiminasab, G. Karsai, A. Easwaran, and A. Dubey, “Efficient out-of-distribution detection using latent space of β-vae for cyber-physical systems,” ACM Transactions on Cyber-Physical Systems, vol. 6, Apr. 2022.

[3] A. Farid, S. Veer, and A. Majumdar, “Task-driven out-of-distribution detection with statistical guarantees for robot learning,” in 5th Annual Conference on Robot Learning, 2021.

[4] A. Reza and C. Wei-Lun, “Unified out-of-distribution detection: A model-specific perspective,” 2023 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 1453–1463, 2023.

[5] J. Yang, K. Zhou, and Z. Liu, “Full-spectrum out-of-distribution detection,” International Journal of Computer Vision, vol. 131, p. 2607–2622,June 2023.

[6] T. Haider, K. Roscher, F. Schmoeller da Roza, and S. Gunnemann, “Out-of-distribution detection for reinforcement learning agents with probabilistic dynamics models,” in Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems, AAMAS’23, p. 851–859, International Foundation for Autonomous Agents and Multiagent Systems, 2023.

[7] L. Navitus, K. Sandbrink, J. Foerster, T. Franzmeyer, and C. S. de Witt, “Rethinking out-of-distribution detection for reinforcement learning : Advancing methods for evaluation and detection,” 2024.

[8] A. J. Singh and A. Easwaran, “Pas: Probably approximate safety verification of reinforcement learning policy using scenario optimization,” in Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems, AAMAS ’24, p. 1745–1753, International Foundation for Autonomous Agents and Multiagent Systems, 2024.

[9] N. Potteiger, A. Samaddar, H. Bergstrom, and X. Koutsoukos, “Designing robust cyber-defense agents with evolving behavior trees,” in 2024International Conference on Assured Autonomy (ICAA), pp. 1–10, 2024.

**Link to the paper:** Out-of-Distribution Detection for Neurosymbolic Autonomous Cyber Agents | IEEE Conference Publication | IEEE Xplore

**Link to the code:** anki2911/Out-of-Distribution-Detection_Neurosymbolic_RL_Agents: Implementation of "Out-of-Distribution Detection for Neurosymbolic Autonomous Cyber Agents"

# THANK YOU