Related Work

A table of related works

<table>
<thead>
<tr>
<th>Method</th>
<th>Online Adaptive Attack-model agnostic Mitigation</th>
<th>Related Work</th>
</tr>
</thead>
<tbody>
<tr>
<td>VFAS (Lin et al 2017)</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>ARLS (Mady et al 2017)</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>MLAH [This paper]</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

Online: no offline training retraining required. Adaptive: can adapt to a change in attack strategy. Attack model agnostic: assumes no specific attack model. Mitigation: is the impact of the attack actually mitigated?

- Previous attempts in mitigating adversarial attacks have been successful against only assumed specific attack models.
- Robust training strategies are typically off-line (e.g., using augmented datasets) and may fail to adapt to different attacker strategies in an online fashion.

Motivation

- The use of Deep reinforcement learning (DRL) for cyber-physical systems has raised concerns around safety and robustness of autonomous agents.
- It is computationally feasible for a bad actor to fool a DRL policy into behaving sub-optimally. Even very small perturbations can result in significant performance loss.

Formulation

- We want to maximize this discounted reward sum by optimizing a policy
  \[ V: S \rightarrow \mathbb{R}, R(t) = \gamma R_{t+n}, \sum_{t=0}^{T-1} \pi_t(s_t) \]
- A finite set of MDPs \( M = \{ m_1, m_2, \ldots, m_n \} \), where an MDP \( m_i \) for all \( i \in \{ 0, 1, \ldots, n \} \) is sampled for learning at time \( t \) and corresponding sub-policies \( \{ \pi_0, \pi_1, \ldots, \pi_n \} \) which may individually be used at any instant.
- The joint hierarchical model is composed of sub-policies, which can represent adversarial and nominal conditions.
  \[ \Phi(t) = \mathbb{E}_{\pi_{true}, \pi_{nom}} \sum_{t=0}^{T-1} \pi_t(s_t) \]

Algorithm

Algorithm 1: MLAH

Input: \( \pi_{true}, \pi_{nom} \) sub-policies parameterized by \( \theta_{true}, \theta_{nom} \). Master policy \( \pi_{true} \) with parameter vector \( \phi \).

Initialize \( \pi_{true}, \pi_{nom}, \phi \)

for pre-training iterations (optional) do

Train \( \pi_{true} \) and \( \pi_{nom} \) on only nominal experiences.

end

for learning life-time do

for Time steps \( t \) to \( T \) do

Compute \( A_{true} \) over sub-policies (see eq. 1)

\( \pi_{true} \) selects to switch or stay with sub-policy based on \( A_{true} \) observations to take action

Estimate all \( A_{true} \) for \( \pi_{true} \) and \( \pi_{nom} \) over \( T \) steps

Estimate all \( A_{true} \) for \( \pi_{true} \) over \( T \) with respect to \( A_{true} \), observations

Optimize \( \beta_{true} \) based on experiences collected from \( \pi_{true} \)

Optimize \( \beta_{nom} \) based on experiences collected from \( \pi_{nom} \)

Optimize \( \phi \) based on all experiences with respect to \( \pi_{true} \) observations

end

Analysis of return Lower-bound:

If \( \Delta C < \Delta V \), where \( C \geq \sum_{t=0}^{T-1} \pi_t(s_t) \) and \( \Delta V = V_{true} - V_{nom} \), then the conditioned policy has a higher lower bound of expected discounted reward compared to that of the unconditioned policy.

References: