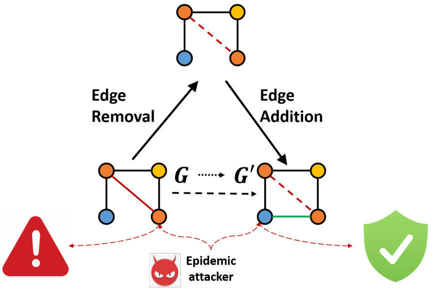
Deep Reinforcement Learning-based Vulnerability-Aware Network Adaptations for Resilient Software-Defined Networks

- **Motivation**
  - Ensuring security and service availability simultaneously
  - Developing autonomous network adaptation schemes
  - Performing efficient and effective moving target defense in large-scale networks
- **Research Goal**: Achieve network security and network resilience by network topology adaptation under a software polyculture environment



Problem Statement

- **Main idea**: Optimize network security ($\mathcal{F}_C$) + connectivity ($\mathcal{S}_G$) + service availability ($\mathcal{P}_{MD}$)
- **Objective function** :

$$\arg\max_{b_A, b_R} f(G') - f(G), \quad s.t. \quad 0 \leq b_A + b_R \leq B,$$

$$\begin{aligned} G &: \text{original network} \\ G' &: \text{adapted network} \\ b_A &: \text{addition budget} \\ b_R &: \text{removal budget} \\ \textbf{O-SG}: f &: G \mapsto \mathcal{S}_G(G) - \mathcal{F}_C(G) \\ \textbf{O-MD}: f &: G \mapsto \mathcal{P}_{MD}(G) - \mathcal{F}_C(G) \\ \textbf{O-SG-MD}: f &: G \mapsto \mathcal{S}_G(G) + \mathcal{P}_{MD}(G) - \mathcal{F}_C(G) \end{aligned}$$

Network Model

- A centralized system with one centralized controller

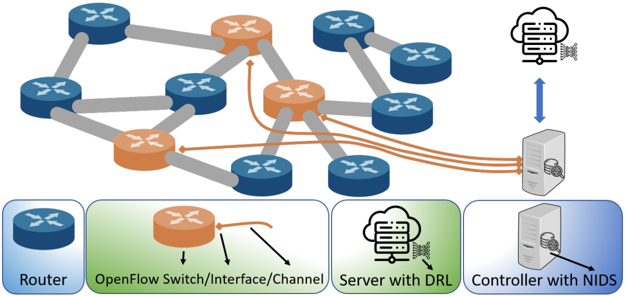


Figure 1: An overview of the network model.

Node and Attack Models

- **Node Model**
  - Activity indicator(IDS): $na_i = 1(\text{alive})/0(\text{failed})$
  - Compromise indicator: $nc_i = 1(\text{compromised})/0(\text{not compromised})$
  - Software version: $s_i \in [1, N_s]$, N_s: # of available software packages
  - Software vulnerability: $sv_i \in [0, 1]$ [1]
- **Attack Model**
  - Epidemic attacks: P_a
    - Perform two attack trials to infect its direct neighbors
    - Learn software versions along attacks
  - State manipulation attacks: P_s
    - Inject fake rewards
  - Packet drop attack
  - Packet modification attack

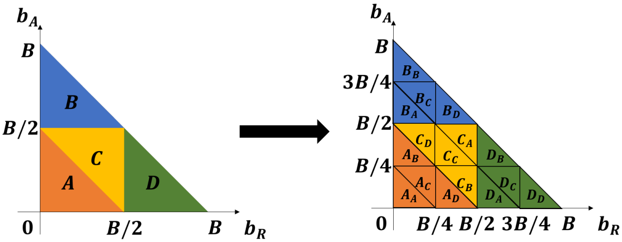
[1] The extent of a Common Vulnerabilities and Exposures (CVE) based on a Common Vulnerability Scoring System (CVSS)

Vulnerability Ranking of Edges and Nodes (VREN)

- Precision control by # of attack simulations
- Edge vulnerability level V_E: # of times it is used by attackers to compromise other nodes
- Node vulnerability level V_V: # of times it becomes an attacker (being compromised)
- Ranking system
  - R_E: edge ranking based on V_E in descending order
  - R_V: node ranking based on V_V in ascending order
- Adaptation based on budget constraints $[b_R, b_A]$
  - b_R: edge removal budget
  - b_A: edge addition budget

Fractal-based Solution Search (FSS)

- Reduce solution search space in edge addition and removal budgets
- Self-similar fractals
  - Centroid representation for each division
  - Logarithm complexity: $\lceil \log B \rceil$
    (B: the upper bound of the total adaptation budget)
- Discrete evaluation
  - Nearest integer points: (b_R, b_A)
    (b_R: edge removal budget, b_A: edge addition budget)



DRL-based Budget Adaptation

- **States**
  - $s_t = (b_A^t, b_R^t, G_t')$
  - b_R^t: removal budget at time t
  - b_A^t: addition budget at time t
  - G_t': the network at time t
- **Actions**
  - FSS: $\mathbf{a}_t = \{A, B, C, D\}$, where $1 \leq t \leq \lceil \log_2 B \rceil$
  - LS (Linear Search): $\mathbf{a}_t = \{stop, add, remove\}$, where $1 \leq t \leq B$
- **Rewards**
  - $\mathcal{R}(s_t, a_t, s_{t+1}) = f(G_{t+1}') - f(G_t')$, where $f =$O-SG/O-MD/O-SG-MD.

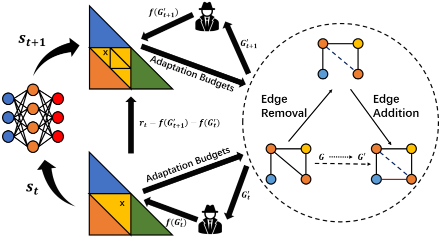


Figure 2: The overall architecture of the proposed DREVAN and DeepNETAR: The color of each node refers to a different software package installed in it.

Previous Work Summary

- Developed a DRL-based framework, DREVAN, to minimize system vulnerability while maintaining comparable or better network connectivity.
- Demonstrated the outperformance of three different types of Deep Q-learning algorithms against the counterpart and baseline schemes.
- Devised DQN-DeepNETAR-SG-MD can better ensure security, connectivity, and service availability simultaneously with an appropriate evaluation function.
- Found that the size of the giant component, as a network connectivity metric, is more related to security than actual service availability under epidemic attacks.

DRL-based Budget Adaptation

- **Reward tree**
  - Store the history states
  - Online update
- **Simple rewards**
  - Reduce evaluation times
  - $\mathcal{R}(s_t, a_t, s_{t+1}) = f(G'_{t+1})$ if $t = \lceil \log_2 B \rceil$; 0 otherwise.
- **Parallel environments**
  - Store action sequences
  - Evaluate together

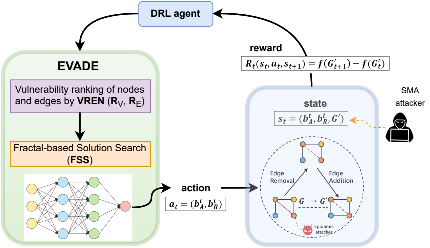
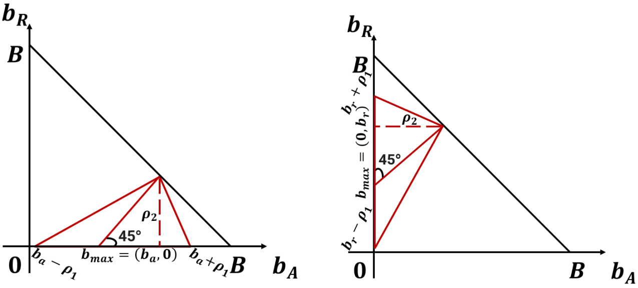


Figure 3: DRL-based optimal budget identification in EVADE.

Greedy MTD Using Density Optimization

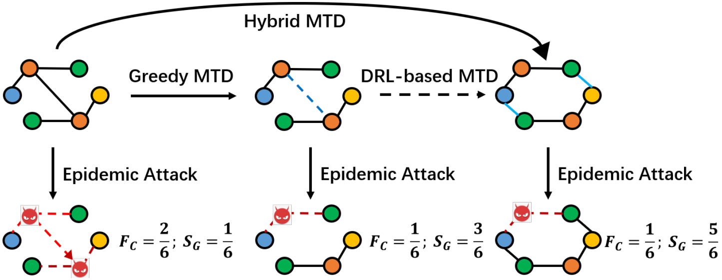


(a) $b_{max} = (b_a, 0)$ (b) $b_{max} = (0, b_r)$

Figure 4: The procedure of generating an expanded triangle based on the proposed greedy MTD algorithm: This algorithm can reduce a solution search space to identify an optimal (b_A^*, b_R^*).

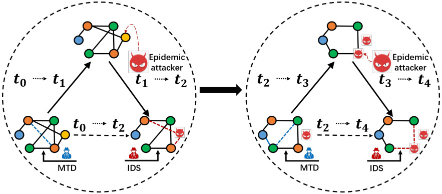
- Single variable optimization
  - Density candidates
  - Minimal budget
- Approximate sampling
  - Sample with precision
  - Choose the best

Hybrid Moving Target Defense



- Density optimization
  - Find the best budget
  - Expand the triangle search area
- DRL-based MTD
  - VREN
  - FSS
  - DRL-based budget adaptation

Experimental Setup



Datasets

- **Synthetic network**
  - Random network (ER): $N = 200,\ p = 0.05,\ M = 1021$
- **Real network**
  - Dense network: $N = 963,\ M = 11310$
  - Medium network: $N = 1000,\ M = 6123$
  - Sparse network: $N = 1476,\ M = 2907$

Attack order

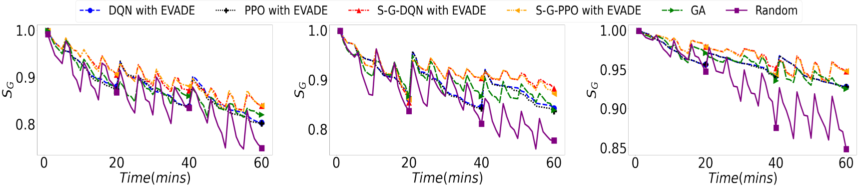
- State Manipulation Attacks: P_s
- Epidemic attacks: P_a/λ

Asymptotic Complexity Analysis

Scheme	Complexity
S-G-DQN/S-G-PPO with `EVADE`	$O(n_e \times \lceil \log_2 B \rceil \times t_{train} \times n_a)$
DQN/PPO with `EVADE`	$O(n_e \times \lceil \log_2 B \rceil \times t_{train} \times n_a)$
Genetic Algorithm (GA)	$O(n_s \times n_a)$
Random	$O(n_a)$

- S-G-DQN/S-G-PPO with EVADE incurs a similar low cost as DQN/PPO with EVADE
- GA and Random are the most efficient algorithms among all while showing poor performance

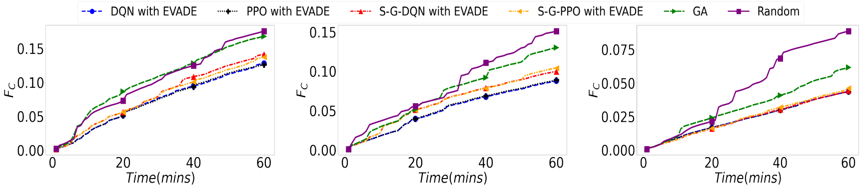
Comparative Performance Analysis with respect to Time in terms of Size of the Giant Component ($\mathcal{S}_G$)



(a) Dense network (b) Medium network (c) Sparse network

- The hybrid MTD schemes outperform more significantly in the dense and sparse networks due to high skewness in degree distribution and better convergence in VREN.

Comparative Performance Analysis with respect to Time in terms of Fraction of Compromised Nodes ($\mathcal{F}_C$).



(a) Dense network (b) Medium network (c) Sparse network

- The overall performance order with respect to the two metrics is: S-G-PPO with `EVADE` $\approx$ S-G-DQN with `EVADE` $\geq$ PPO with `EVADE` $\approx$ DQN with `EVADE` $\geq$ GA $\geq$ Random.

Key Contributions and Findings

- The proposed density optimization (`DO`)-based greedy algorithm further reduces the search space for DRL algorithms, along with `VREN`.
- The proposed hybrid `EVADE` can converge even faster with a smaller solution space than other counterparts by using `DO`.
- The proposed hybrid `EVADE` shows acceptable asymptotic complexity compared to their effectiveness.