DREVAN: Deep Reinforcement Learning-based Vulnerability-Aware Network Adaptations for Resilient Networks

Qisheng Zhang (presenter) [1] Jin-Hee Cho [2]
Terrence J. Moore [3] Frederica Free Nelson [4]

[1,2]Department of Computer Science, Virginia Tech

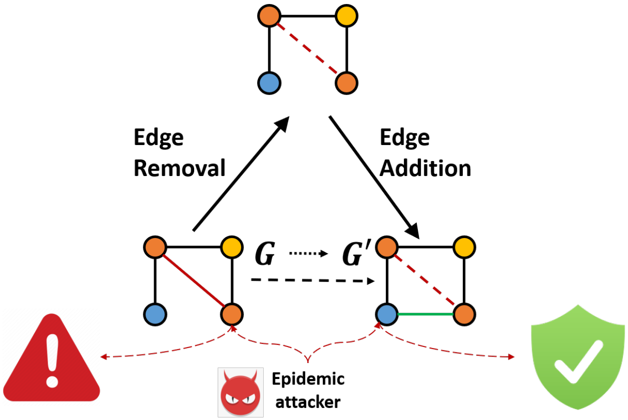
[3,4]US Army Research Laboratory

IEEE CNS 2021, October 2021

Outline

Motivation

- Achieving network security and network resilience by network topology adaptation under software polyculture environment.



Key Contributions

- Proposed a network topology adaptation technique to achieve network resilience in terms of maximizing system security and network connectivity.
- Presented two algorithms to support the DRL agent to efficiently identify an optimal adaptation budget strategy to meet the two system goals.
  - VREN: Vulnerability Ranking algorithm of Edges and Nodes
  - FSS: Fractal-based Solution Search algorithm
- Conducted extensive comparative performance analysis based on six network topology adaptation schemes.
- Found that DRL-based network topology adaptations particularly outperform with regard to minimizing system security vulnerability.

Related Work

- **Deployment of diversity-based network adaptations**
  - Metric-based: graph coloring based software allocation/assignment [1]
  - Metric-free: software assignment [2]; network topology shuffling [3]
- **DRL-based network topology shuffling**
  - Addition: adding edges to networks [4]
  - Removal: removing edges from networks [5]
  - Shuffling: redirecting edges in networks [6] [7]
- **Limitations**
  - Lack of study to determine an optimal number of edge adaptations for resilient networks
  - Limited topology operations
  - Slow convergence for DRL agents to identify optimal solutions

[1] Borbor et al., 2019
[2] Yang et al., 2016
[3] Hong et al., 2016
[4] Darvariu et al., 2020
[5] Dai et al., 2018
[6] Chai et al., 2020
[7] Zhang et al., 2020

Problem Statement

- **Main idea**: optimize network security($\mathcal{F}_C$) + resilience($\mathcal{S}_G$)
- **Objective function** :

$$\arg\max_{b_A, b_R} f(G') - f(G), \quad s.t. \quad 0 \leq b_A + b_R \leq B, \tag{1}$$

$$G : \text{original network}$$
$$G' : \text{adapted network}$$
$$b_A : \text{addition budget}$$
$$b_R : \text{removal budget}$$
$$f : G \mapsto \mathcal{S}_G(G) - \mathcal{F}_C(G)$$

System Model

- **Network Model**: A centralized system with one centralized controller
- **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: $sv_i \in [0, 1]$ [8]
- **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

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

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)

- 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)

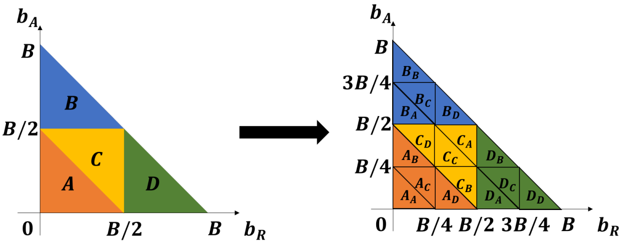


Figure 1: Generation of self-similar fractals to reduce solution search space in edge addition and removal budgets, (b_A, b_R).

Proposed DREVAN Framework

- **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: $a_t = \{A, B, C, D\}$, where $1 \leq t \leq \lceil \log_2 B \rceil$
    - LS (Linear Search): $a_t = \{stop, add, remove\}$, where $1 \leq t \leq B$
  - Rewards
    - $\mathcal{R}(s_t, a_t, s_{t+1}) = f(G'_{t+1}) - f(G'_t)$, where $f : G \mapsto \mathcal{S}_G(G) - \mathcal{F}_C(G)$ (size of the giant component - fraction of compromised nodes).

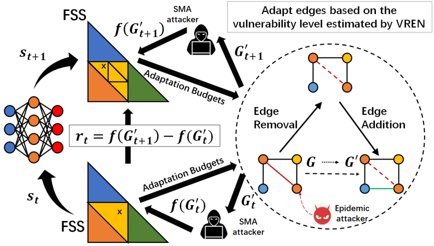


Figure 2: The overall architecture of the proposed DREVAN: The color of each node refers to a different software package installed in it.

Experimental Setup

- Random Graph
  - ER: Erdős–Rényi random graph model
  - Number of nodes $N = 200$
  - Connection probability $p = 0.05$
- Attack Types Considered
  - State Manipulation Attacks
    - Probability for a system state to be manipulated by the attacker $P_s = 0.3$
  - Epidemic Attacks
    - Fraction of initial attackers in a network $P_a = 0.3$

Experimental Setup

Table 1: Key Design Parameters, Meanings, and Default Values

Param.	Meaning	Value
n_a	Attack simulation times	500
n_r	Number of simulation runs	200
n_e	Training episodes of DRL-based schemes	1000
N	Total number of nodes in a network	200
k	Upper hop bound for edge addition	3
γ	Intrusion detection probability	0.9
P_{fn}, P_{fp}	False negative or positive probability	0.1, 0.05
x	Degree of software vulnerability	0.5
p	Connection probability between pairs of nodes in an ER network	0.05
l	Number of software packages available	5
P_a	Fraction of initial attackers in a network	0.3
B	Upper bound of the total adaptation budget	500
P_s	Probability of state manipulation attacks	0.3
D_r	Detection rate of state manipulation attacks	0.99

Asymptotic Analysis of the Compared Schemes

Scheme	Complexity
DQN with DREVAN	$O(n_e \times \lceil \log_2 B \rceil \times T_{\text{train}} \times n_a)$
DQN with FSS	$O(n_e \times \lceil \log_2 B \rceil \times T_{\text{train}} \times n_a)$
DQN with VREN	$O(n_e \times B \times T_{\text{train}} \times n_a)$
DQN	$O(n_e \times B \times T_{\text{train}} \times n_a)$
Greedy	$O(\lceil \log_2 B \rceil \times n_a)$
Random	$O(n_a)$
Optimal	$O(B^2 \times n_a)$

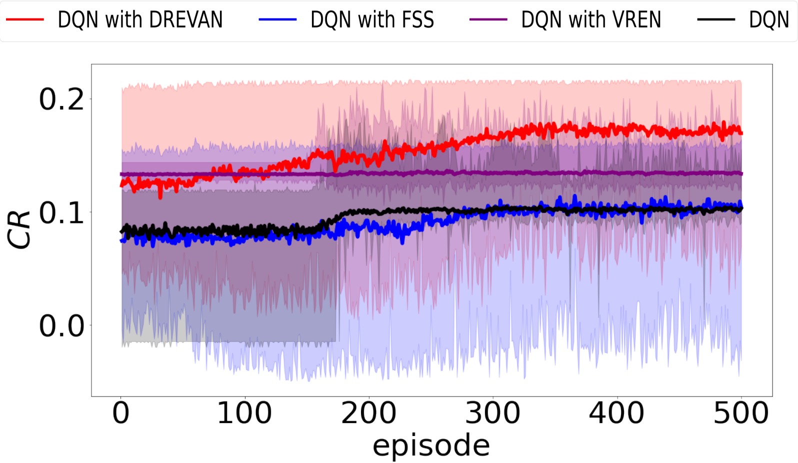
n_e : the training episode

B : the upper bound of total adaptation budget

T_{train} : the training time per episode

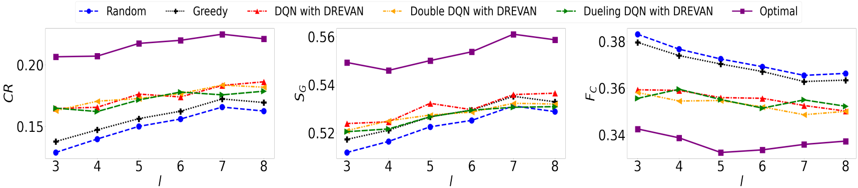
n_a : the attack simulation times

Converged Reward with respect to Training Episodes



- DQN with DREVAN performs the best.
- DQN with FSS can only learn a sub-optimal policy.
- DQN with VREN and DQN cannot learn well with LS.

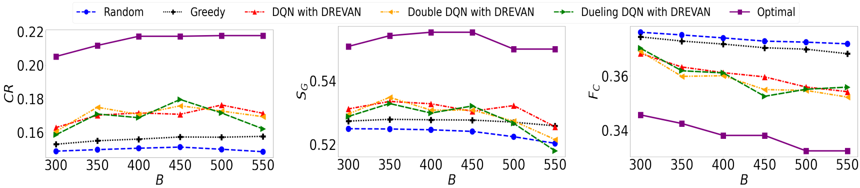
Effect of Varying the Number of Software Packages Available (I) under an ER Network



(a) Converged reward ($\mathcal{CR}$) (b) Size of the giant component ($\mathcal{S}_G$) (c) Fraction of compromised nodes ($\mathcal{F}_C$)

- As I increases, $\mathcal{F}_C$ drops and $\mathcal{S}_G$ increases.
- Overall performance order: Optimal $\geq$ DQN $\approx$ Double DQN $\approx$ Dueling DQN $\geq$ Greedy $\geq$ Random.

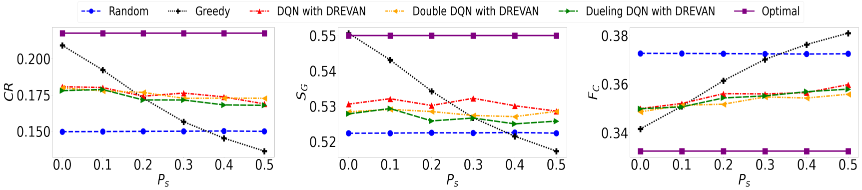
Effect of Varying the Upper Bound of the Total Adaptation Budget (B) under an ER Network



(a) Converged reward ($\mathcal{CR}$) (b) Size of the giant component ($\mathcal{S}_G$) (c) Fraction of compromised nodes ($\mathcal{F}_C$)

- Higher B increases $\mathcal{CR}$ while decreasing $\mathcal{F}_C$, but it does not necessarily improve $\mathcal{S}_G$.
- Overall performance order: Optimal $\geq$ DQN with DREVAN $\approx$ Double DQN with DREVAN $\approx$ Dueling DQN with DREVAN $\geq$ Greedy $\geq$ Random.
- Dueling DQN with DREVAN is more sensitive to B than DQN with DREVAN and Double DQN with DREVAN.

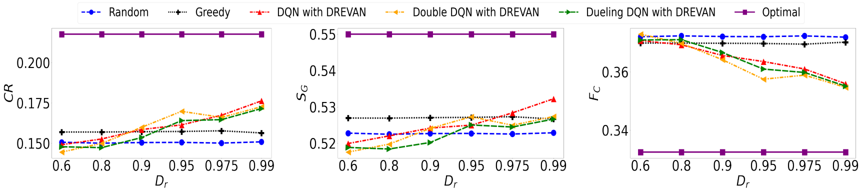
Effect of Varying Probability of State Manipulation Attacks (P_s) under an ER Network



(a) Converged reward ($\mathcal{CR}$) (b) Size of the giant component ($\mathcal{S}_G$) (c) Fraction of compromised nodes ($\mathcal{F}_C$)

- Higher P_s brings lower $\mathcal{CR}$ and $\mathcal{S}_G$ while introducing more $\mathcal{F}_C$.
- The Greedy scheme is more sensitive to P_s than DRL-based schemes.

Effect of Varying Detection Rate of State Manipulation Attacks (D_r) under an ER Network



(a) Converged reward ($\mathcal{CR}$) (b) Size of the giant component ($\mathcal{S}_G$) (c) Fraction of compromised nodes ($\mathcal{F}_C$)

- Higher D_r increases $\mathcal{CR}$ and $\mathcal{S}_G$ while lowering $\mathcal{F}_C$.
- DRL-based schemes only outperform with higher D_r.
- Overall DQN with DREVAN performs slightly better than Double DQN with DREVAN and Dueling DQN with DREVAN.

Conclusions

- Proposed a fractal-based environment (FSS) that can significantly reduce the training complexity of our DRL algorithms.
- Proposed a vulnerability-aware ranking algorithm (VREN) to strategically adapt edges for efficient and effective network configurations.
- Proposed a DRL-based framework, DREVAN, to minimize system vulnerability while maintaining comparable or better network connectivity.
- Showed the outperformance of three different types of Deep Q-learning algorithms against the counterpart and baseline schemes.

Any Questions?

Thank you!

Qisheng Zhang at
qishengz19@vt.edu

National Capital Region Campus
7054 Haycock Rd., Office 314
Falls Church, VA 22043