COVER-MTD: Learning-Guided DDoS Defense for Edge-IoT

Amir Javadpour*, Forough Ja'fari†, Chafika Benzaïd‡, and Tarik Taleb§
*ICTFICIAL Oy, Espoo, Finland; Email: a.javadpour87@gmail.com
†Department of Computer Engineering, Sharif University of Technology, Tehran, Iran
‡Faculty of Information Technology and Electrical Engineering, University of Oulu, Finland
§Faculty of Electrical Engineering and Information Technology, Ruhr University Bochum, Germany

***Abstract*—Edge-IoT services are highly vulnerable to Distributed Denial of Service (DDoS) attacks, as compromised resource-constrained hosts can be abused to overload critical servers. Moving Target Defense (MTD) can reduce this risk by changing network-visible properties, but its effectiveness depends on selecting suitable mutation targets. Random mutation may waste resources, vulnerability-only mutation may ignore service reachability, and exposure-only mutation may overlook compromise likelihood. This paper proposes COVER-MTD, a reinforcement-learning-guided host-mutation framework that jointly considers host vulnerability and critical-service exposure. COVER-MTD ranks hosts through pairwise reinforcement learning, allowing the policy to compare local host risks without depending on a fixed network size. Mininet-based evaluation shows that COVER-MTD reduces server failures and attack success compared with random, vulnerability-only, and exposure-only MTD strategies, while maintaining comparable delay, packet loss, and host-selection overhead.**

***Index Terms*—Moving Target Defense, DDoS mitigation, Edge-IoT security, reinforcement learning, host mutation, service exposure, vulnerability-aware defense.**

I. Introduction

Edge-IoT networks are becoming an essential part of modern digital services, but their distributed and heterogeneous nature also increases the attack surface. Ordinary IoT hosts often interact with edge gateways, local controllers, and critical service nodes. Once a subset of these hosts is compromised, the adversary can coordinate them to generate malicious traffic and overload protected services. This makes DDoS mitigation particularly challenging, because the defender must react under strict latency, resource, and service-availability constraints [1, 2, 3].

Most conventional DDoS defenses are reactive. They inspect traffic, detect anomalies, filter suspicious flows, or enforce static access-control policies. These mechanisms are valuable, but they may still allow the attacker to collect information about reachable services, vulnerable hosts, and stable network paths before the countermeasure becomes active. In dynamic edge-IoT environments, a more proactive defense is needed to reduce the reliability of such reconnaissance information.

Moving Target Defense (MTD) provides this proactive capability by changing selected attributes of the protected system. Depending on the deployment, MTD may modify addresses, ports, forwarding paths, service placement, routing associations, or host reachability patterns [4, 5, 6, 7]. By making the attack surface less stable, MTD forces the adversary to repeat reconnaissance and reduces the usefulness of previously collected information. However, MTD is not free. Frequent or poorly targeted mutation may increase delay, packet loss, controller load, and operational complexity. Therefore, the central design question is not simply whether mutation should be used, but which hosts should be mutated.

Existing MTD strategies often answer this question using a single criterion. Random mutation is simple but blind to risk. Vulnerability-only mutation focuses on hosts that are easy to compromise, but it may select hosts that have little influence on critical services. Connection- or traffic-driven mutation considers structural exposure, but it may ignore whether the host is actually likely to be compromised. In DDoS attacks against critical services, these dimensions are coupled: a host is dangerous when it is both vulnerable and able to provide access to one or more protected servers [8, 9].

This paper proposes **COVER-MTD**, a **COnnection- and Vulnerability-aware Edge Reinforcement Moving Target Defense** framework. COVER-MTD ranks host mutation candidates using two complementary factors: normalized host vulnerability and critical-service exposure. The exposure value measures the number of critical servers reachable from a host. A host with high vulnerability and high exposure is therefore treated as a strong mutation candidate because its compromise can contribute to coordinated service disruption.

To avoid a fixed topology-dependent ranking rule, COVER-MTD formulates host prioritization as a reinforcement-learning-based pairwise comparison problem. The agent compares two hosts at a time and decides whether their order in the mutation list should be preserved or exchanged. Since the state is based on relative local information rather than the full topology dimension, the learned policy can be reused when the number of hosts changes. This property is important for edge-IoT scenarios where devices may join, leave, or change their reachability patterns.

The main contributions are as follows:

- We propose **COVER-MTD**, a host-level MTD framework for DDoS-resilient edge-IoT services.
- We define a critical-service exposure metric and combine it with host vulnerability to prioritize mutation candidates.
- We formulate mutation ranking as a pairwise reinforcement-learning problem, enabling topology-flexible host selection.
- We evaluate COVER-MTD in a Mininet-based SDN environment against random, vulnerability-only, and

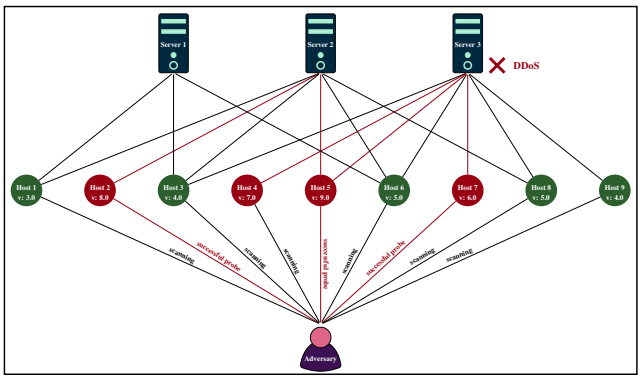


Fig. 1. DDoS-oriented edge-IoT scenario with ordinary hosts, critical servers, and compromised attack sources.

exposure-only MTD baselines using attack resilience, overhead, and learning-related metrics.

The rest of this paper is organized as follows. Section II reviews representative MTD studies. Section III presents the system and attack formulation. Section IV introduces the COVER-MTD framework. Section V provides an illustrative shuffling case. Section VI reports the evaluation results. Section VIII concludes the paper.

II. Related Work

MTD has been studied as a proactive security mechanism for dynamic networks, cloud systems, SDN infrastructures, and IoT environments. Its main goal is to reduce the value of adversarial reconnaissance by changing selected system properties before or during an attack. Existing MTD studies can be discussed from three viewpoints: the mutation surface, the decision criterion, and the use of learning-based optimization.

At the mutation-surface level, some studies focus on network reconfiguration. These works modify routing paths, forwarding rules, or topology visibility to make the attacker's view unstable [10, 11, 12, 13]. Other studies target service or host attributes, including IP addresses, MAC addresses, port numbers, and service locations [14]. Lower-level mutation, such as memory-layout randomization, has also been investigated in cyber-physical and hardware-oriented contexts [15]. In contrast, this work focuses on host-level mutation for DDoS defense, where the defender must identify the ordinary hosts whose mutation most effectively reduces attack reachability toward critical services.

A first class of host-level methods uses randomization. DDuMaS combines SDN and MTD to defend against DDoS attacks by randomizing address and topology information [16]. SDN-based honeypot and shuffling approaches also use random host mutation and random shuffling intervals in IoT-related settings [17]. These methods are easy to implement and have low decision cost. Nevertheless, they do not distinguish between high-risk and low-risk hosts, which may lead to inefficient use of mutation resources.

A second class uses vulnerability information. CHAOS relies on CVSS-based information to guide IP and port mutation [18]. Other CVSS-driven methods select the most vulnerable host as the mutation target [19]. Attack-graph-based approaches extend this idea by finding paths that are likely to be exploited and mutating hosts on those paths [20]. Recent work has also used deep reinforcement learning over probabilistic attack graphs to optimize MTD deployment under cost constraints [21]. These methods are more risk-aware than random mutation, but vulnerability alone is not enough for DDoS-oriented host selection because a vulnerable host may not be connected to critical services.

A third class uses traffic, connection, or service-load information. Game-theoretic shuffling has been used to select virtual machines according to user connections, cost, and availability [22]. Deep reinforcement learning has also been adopted for adaptive MTD with host states such as probing activity, compromise state, and online/offline status [23]. DIVERGENCE uses flow rate and inspection-resource allocation to combine traffic inspection and mutation decisions [24]. More recent studies consider edge and industrial scenarios, including resource-efficient MTD for low-rate DDoS attacks in edge clouds [25] and adaptive MTD traffic management for IIoT DDoS protection [26]. These works highlight the importance of operational context, but many of them depend on traffic monitoring, service migration, or infrastructure-level reconfiguration rather than direct host ranking based on both vulnerability and critical-service reachability.

Recent domain-specific MTD studies further show the growing relevance of adaptive mutation. MTDNS uses SDN and NFV to redirect DNS traffic toward alternate DNS servers during flooding attacks [27]. In next-generation networks, deep reinforcement learning and MTD have been combined for slice-level security adaptation [28]. These studies address important service- or slice-level problems, while the problem considered here is different: identifying which ordinary edge-IoT hosts should be mutated when compromised hosts may jointly overload protected servers.

The above review shows that random methods are simple but risk-insensitive, vulnerability-based methods capture compromise likelihood but may ignore service exposure, and connection-based methods capture exposure but may overlook compromise likelihood. COVER-MTD addresses this gap by jointly using vulnerability and critical-service exposure and by learning pairwise mutation-priority decisions that can be reused across network sizes. Table I summarizes the positioning of representative MTD mechanisms and evaluated strategies with respect to mutation target, vulnerability awareness, exposure awareness, and learning capability.

III. System and Attack Formulation

A. Adversarial Setting

We consider an edge-IoT service network containing ordinary hosts and protected service nodes. The protected nodes are called critical servers because their failure directly affects service availability. Each ordinary host can communicate only with a subset of critical servers according to the access policy.

TABLE I
COMPACT COMPARISON OF REPRESENTATIVE MTD MECHANISMS AND EVALUATED STRATEGIES

Work / Strategy	Role in this paper	Target	Vuln.	Exp.	Learn.
DDuMaS [16]	Random-MTD basis	Net.	✗	✗	✗
MaSbH [17]	Random-MTD basis	Host	✗	✗	✗
CHAOS [18]	Vulnerability-only basis	Host/Port	✓	✗	✗
SbSMsiCN [19]	Vulnerability-only basis	Host	✓	✗	✗
BAP [20]	Vulnerability-only basis	Path/Host	✓	partial	✗
MTD-AG [21]	Vulnerability-only basis	Attack graph	✓	graph	✓
TgCeS [22]	Exposure-only basis	VM	✗	✓	✗
DIVERGENCE [24]	Exposure-only basis	Traffic/Host	✗	✓	✓
RE-MTD [25]	Edge/service-level reference	Edge service	✗	traffic	✓
MTDTM [26]	IIoT/service-level reference	IIoT service	✗	traffic	partial
MTDNS [27]	Service-level reference	DNS service	✗	service	✗
Slice-MTD [28]	Slice/service-level reference	Slice	✗	service	✓
COVER-MTD	**Proposed method**	**Host**	✓	✓	✓

Consequently, different hosts have different levels of service exposure.

The adversary is external. It performs reconnaissance, identifies reachable hosts, compromises vulnerable hosts, and then uses the compromised nodes as DDoS traffic sources. A critical server is considered failed when the number of malicious incoming host-server connections reaches a predefined threshold. The defender does not try to identify the external adversary directly. Instead, the defender mutates selected hosts to reduce the usefulness of the attacker's learned reachability information (Fig. 1).

B. Network Model

The network is represented as

$$\mathcal{G} = (\mathcal{H}, \mathcal{S}, \mathbf{A}, \boldsymbol{\nu}, \delta), \tag{1}$$

where $\mathcal{H} = \{1, \ldots, H\}$ is the host set, $\mathcal{S} = \{1, \ldots, S\}$ is the critical-server set, $\mathbf{A} \in \{0,1\}^{H \times S}$ is the host-server reachability matrix, $\boldsymbol{\nu}$ is the vulnerability vector, and δ is the crash threshold. The entry $a_{h,s} = 1$ means that host h can reach server s; otherwise, $a_{h,s} = 0$.

The vulnerability score of host h is ν_h. A normalized vulnerability indicator is defined as

$$\bar{\nu}_h = \frac{\nu_h}{10}. \tag{2}$$

A simple attacker-effort indicator can then be written as

$$\kappa_h = 100(1 - \bar{\nu}_h). \tag{3}$$

Thus, a more vulnerable host has a lower compromise effort.

The critical-service exposure of host h is

$$\eta_h = \sum_{s \in \mathcal{S}} a_{h,s}. \tag{4}$$

This value indicates how many protected servers can be reached from host h.

C. Mutation Objective

Let $\mathcal{X} \subseteq \mathcal{H}$ be a set of compromised hosts. Server s is overloaded when

$$\sum_{h \in \mathcal{X}} a_{h,s} \geq \delta. \tag{5}$$

The attacker succeeds at the network level if this condition holds for all critical servers:

$$\forall s \in \mathcal{S}, \quad \sum_{h \in \mathcal{X}} a_{h,s} \geq \delta. \tag{6}$$

The defender selects a mutation set $\mathcal{M} \subseteq \mathcal{H}$ to reduce the probability that compromised hosts can satisfy (6). Since enumerating all possible mutation subsets requires checking 2^H candidates, COVER-MTD follows a ranking-based strategy. Each host is assigned a local mutation priority:

$$\rho_h = \eta_h \bar{\nu}_h. \tag{7}$$

A high value of ρ_h indicates that host h is both vulnerable and exposed to several critical servers. The learning task is to reproduce this priority structure through pairwise host comparisons.

IV. PROPOSED COVER-MTD FRAMEWORK

A. Overview

COVER-MTD selects host mutation candidates by combining vulnerability and critical-service exposure. The method is based on a simple defense intuition: a host should be mutated with high priority if it is likely to be compromised and if its compromise can help the attacker reach multiple protected services. A host with only one of these properties may still be relevant, but it is not necessarily the strongest mutation candidate.

Instead of using a fixed sorting rule, COVER-MTD trains a pairwise comparison policy. The policy receives a compact state describing two hosts and returns a binary decision indicating whether their order should remain unchanged or be swapped. This allows the method to build a ranked mutation list without relying on a fixed-size action space.

B. Host Features and Actions

Each host is represented as

$$\mathbf{u}_h = [\eta_h, \bar{\nu}_h], \tag{8}$$

where η_h is critical-service exposure and $\bar{\nu}_h$ is normalized vulnerability. The binary action set is

$$\mathcal{A} = \{0, 1\}. \tag{9}$$

Action 0 means that the first host remains before the second host in the priority list, while action 1 exchanges their positions. Under the adopted pairwise comparison schedule, ranking H hosts requires $H(H-1)/2$ comparisons.

Algorithm 1 Pairwise State Encoding in COVER-MTD

Require: Reachability matrix $\mathbf{A}$, vulnerability vector $\boldsymbol{\nu}$, host pair (i, j)
Ensure: Encoded state $\mathbf{o}_{i,j}$
for $h = 1$ to H **do**
  $\eta_h \leftarrow \sum_{s=1}^{S} a_{h,s}$
  $\bar{\nu}_h \leftarrow \nu_h / 10$
Encode b_1 according to (11)
Encode b_2 according to (12)
$b_3 \leftarrow \mathbb{I}\left(\eta_i = \max_{k=i,\ldots,H} \eta_k\right)$
$b_4 \leftarrow \mathbb{I}\left(\bar{\nu}_i = \max_{k=i,\ldots,H} \bar{\nu}_k\right)$
return $\mathbf{o}_{i,j} = [b_1, b_2, b_3, b_4]$

C. Pairwise State

For hosts i and j, the RL state is

$$\mathbf{o}_{i,j} = [b_1, b_2, b_3, b_4]. \tag{10}$$

The first two components encode the relative exposure and vulnerability:

$$b_1 = \begin{cases} 0, & \eta_i = \eta_j, \\ 1, & \eta_i > \eta_j, \\ 2, & \eta_i < \eta_j, \end{cases} \tag{11}$$

and

$$b_2 = \begin{cases} 0, & \bar{\nu}_i = \bar{\nu}_j, \\ 1, & \bar{\nu}_i > \bar{\nu}_j, \\ 2, & \bar{\nu}_i < \bar{\nu}_j. \end{cases} \tag{12}$$

The remaining two components indicate whether host i has the maximum remaining exposure or vulnerability:

$$b_3 = \mathbb{I}\left(\eta_i = \max_{k=i,\ldots,H} \eta_k\right), \quad b_4 = \mathbb{I}\left(\bar{\nu}_i = \max_{k=i,\ldots,H} \bar{\nu}_k\right). \tag{13}$$

The pairwise state-encoding procedure used by COVER-MTD is summarized in Algorithm 1.

D. Reward and Training

The reward is defined according to the priority score ρ_h in (7). For hosts i and j,

$$\mathcal{R}(i, j, a) = \begin{cases} 0, & \rho_i = \rho_j, \\ +10, & \rho_i > \rho_j \text{ and } a = 0, \\ +10, & \rho_i < \rho_j \text{ and } a = 1, \\ -10, & \text{otherwise.} \end{cases} \tag{14}$$

The agent is therefore rewarded when it places the higher-priority host earlier in the mutation list. The training procedure of the COVER-MTD pairwise ranking policy is summarized in Algorithm 2.

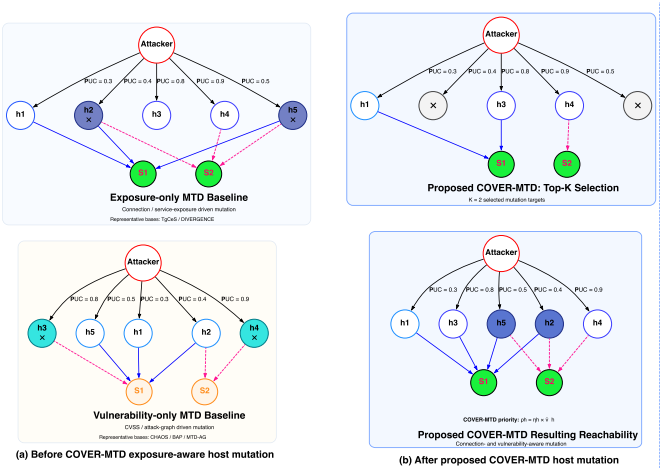
The final ranked list is used to select the top-K mutation candidates. The value of K can be chosen according to the defense budget, service policy, or operational constraints.

V. Illustrative Host-Shuffling Case

This section illustrates why exposure-aware host selection is useful for DDoS mitigation. A vulnerability-only strategy may select hosts that are easy to compromise but weakly connected to the protected services. In contrast, COVER-MTD prioritizes hosts that combine high compromise likelihood with high critical-service exposure.

Algorithm 2 Training of the COVER-MTD Pairwise Ranking Policy

Require: Episodes E, reachability matrix $\mathbf{A}$, vulnerability vector $\boldsymbol{\nu}$
Ensure: Trained policy π_θ
Initialize policy parameters θ
for $e = 1$ to E **do**
  Initialize host list $\mathcal{L}$
  for $i = 1$ to $H - 1$ **do**
    for $j = i + 1$ to H **do**
      $\mathbf{o}_{i,j} \leftarrow$ Algorithm 1$(\mathbf{A}, \boldsymbol{\nu}, i, j)$
      $a \leftarrow \pi_\theta(\mathbf{o}_{i,j})$
      Compute $\mathcal{R}(i, j, a)$ using (14)
      Update θ using $(\mathbf{o}_{i,j}, a, \mathcal{R})$
      if $a = 1$ **then**
        Swap hosts i and j in $\mathcal{L}$
return π_θ



(a) Before exposure-aware mutation.

(b) After COVER-MTD host mutation.

Fig. 2. Illustrative comparison between vulnerability-only mutation and COVER-MTD exposure-aware host mutation.

Fig. 2 compares the two situations. Before connection-aware mutation, several host-server paths remain exploitable. After COVER-MTD-guided mutation, the remaining topology provides fewer coordinated paths toward the protected servers. Thus, the adversary becomes less capable of using compromised hosts to crash multiple critical services simultaneously.

This example also shows why the most vulnerable host is not always the most important host for DDoS defense. The more relevant mutation candidate is often the host that is sufficiently vulnerable and also able to reach several critical servers.

VI. Performance Evaluation

A. Evaluation Setup

COVER-MTD is evaluated in an SDN-enabled Mininet environment. Fig. 3 shows the representative SDN-based Mininet topology used to evaluate host mutation and DDoS resilience. The compared strategies are Random-MTD, which

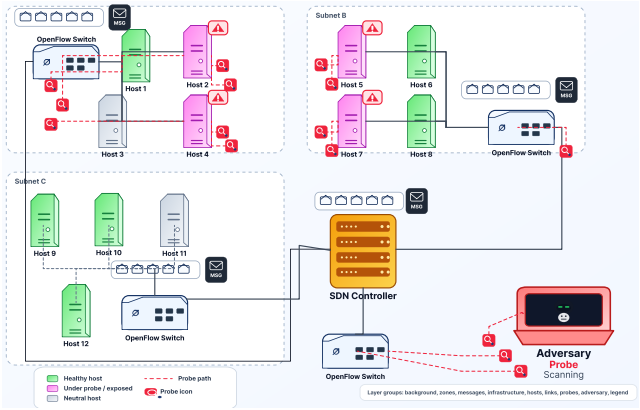
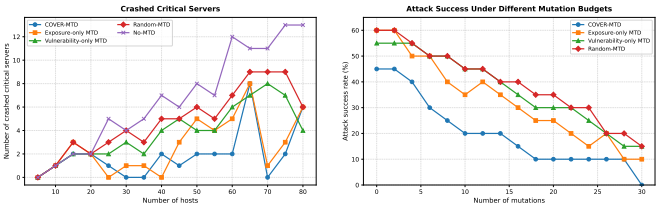


Fig. 3. Representative SDN-based Mininet topology used for host mutation and DDoS evaluation.



(a) Crashed critical servers. (b) Attack success under different mutation budgets.

Fig. 4. DDoS-resilience comparison of COVER-MTD, Exposure-only MTD, Vulnerability-only MTD, Random-MTD, and No-MTD.

mutates hosts without using risk information; Vulnerability-only MTD, which prioritizes hosts with higher vulnerability scores; Exposure-only MTD, which prioritizes hosts with higher critical-service reachability; and the proposed COVER-MTD, which jointly considers vulnerability and exposure. These baseline categories are motivated by representative MTD studies: DDuMaS and MaSbH are used to represent random host mutation, CHAOS, BAP, and MTD-AG motivate vulnerability-aware mutation, and TgCeS and DIVERGENCE motivate exposure-, connection-, or traffic-aware mutation [16, 17, 18, 20, 21, 22, 24]. The learning component of COVER-MTD is implemented using PyTorch.

The simulated networks use random host-server connectivity. The number of ordinary hosts is four times the number of critical servers. Host vulnerabilities and reachability patterns are randomly assigned. In each run, the attacker scans the network, compromises vulnerable hosts, and launches DDoS traffic toward critical servers. All RL-based methods are trained with the same number of episodes.

B. DDoS Resilience

The first evaluation dimension measures whether the defense reduces service disruption. We use two indicators: the number of crashed critical servers and the attack success rate. The attack success rate is the fraction of crashed critical servers over all critical servers.

Fig. 4 shows the resilience results. COVER-MTD produces fewer server failures because it selects hosts that are both vulnerable and structurally exposed. Compared with the no-defense case, the average numbers of protected servers are 5.30, 3.95, 2.40, and 1.75 for COVER-MTD, Exposure-only MTD, Vulnerability-only MTD, and Random-MTD, respectively. The corresponding average attack success rates are 18%, 30.16%, 35.66%, and 37.33%. These results show that joint vulnerability-exposure selection is more effective than single-factor mutation.

C. Network and Decision Overhead

Security improvement should not come at the cost of unacceptable operational overhead. Therefore, we evaluate delay, packet loss, and host-selection time. Delay reflects additional forwarding time after mutation. Packet loss reflects transient forwarding disruption. Host-selection time measures the computational cost of choosing mutation candidates.

Fig. 5 shows that COVER-MTD stays close to the baselines. The average additional delay values are 13.62, 13.37, 13.34, and 13.36 ms for COVER-MTD, Exposure-only MTD, Vulnerability-only MTD, and Random-MTD, respectively. The packet loss rates are 6.79%, 7.00%, 6.90%, and 7.06%. Hence, COVER-MTD does not obtain better security by sacrificing packet delivery.

The average host-selection times are 574 ms, 554 ms, and 562 ms for COVER-MTD, Exposure-only MTD, and Vulnerability-only MTD, respectively. COVER-MTD is slightly slower because it processes two risk dimensions, but the additional cost is small compared with the reduction in attack success.

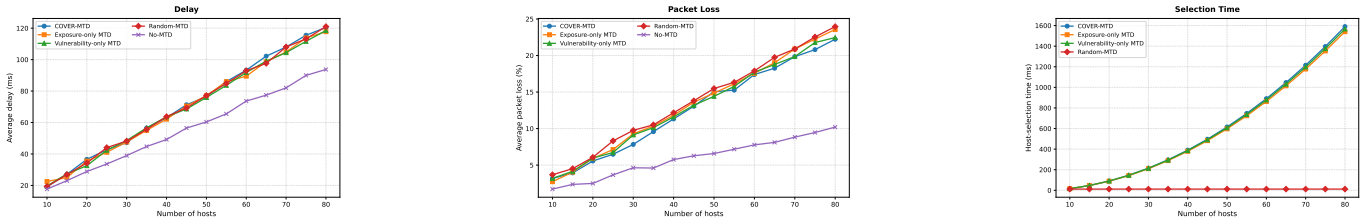
Fig. 7 summarizes the trade-off between selection time and attack success. Random-MTD is computationally cheap but weak. COVER-MTD requires slightly more selection time, but it provides substantially stronger protection.

D. Policy Reuse Across Network Sizes

The final evaluation examines whether the learned pairwise policy can be reused when the number of hosts changes. This is important because edge-IoT networks are dynamic and retraining for every topology is impractical. Several models are trained with different host counts and evaluated on networks of different sizes. Fig. 6 shows that a model trained with fewer hosts remains effective for larger networks. This behavior is due to the pairwise state representation, which is independent of the total input dimension. Training time increases with the number of hosts because the number of comparisons grows as $H(H-1)/2$, but the reward curves stabilize after a limited number of episodes. Overall, the results show that COVER-MTD improves DDoS resilience while maintaining practical overhead. Its main advantage is that it removes high-risk attack paths by selecting hosts that are simultaneously vulnerable and exposed to critical services.

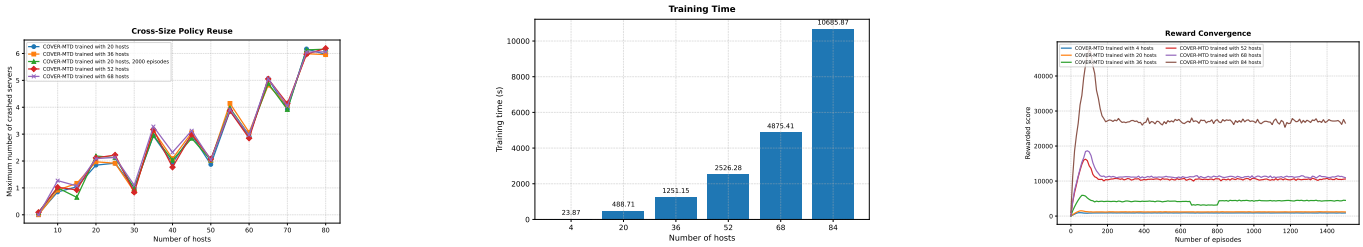
VII. Discussion

Fig. 8 provides an illustrative comparison of the host-mutation behavior of the evaluated MTD strategies. The



(a) Delay. (b) Packet loss. (c) Selection time.

Fig. 5. Overhead comparison in terms of delay, packet loss, and host-selection time.



(a) Cross-size policy reuse. (b) Training time. (c) Reward convergence.

Fig. 6. Learning behavior and cross-size applicability of the COVER-MTD pairwise ranking policy.

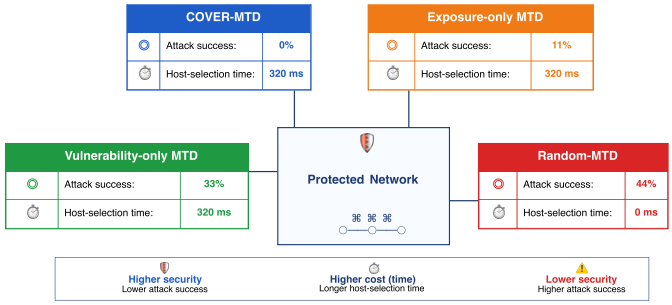


Fig. 7. Security–cost trade-off between host-selection time and attack success rate.

figure summarizes how Random-MTD, Exposure-only MTD, Vulnerability-only MTD, and the proposed COVER-MTD select mutation candidates under the same attacker-to-host compromise probabilities and host-to-server reachability relations. In this illustration, blue links represent service paths toward S_1, while dashed magenta links represent service paths toward S_2. The marked hosts indicate the hosts selected for mutation under each strategy.

As shown in Fig. 8, Random-MTD selects mutation targets without using either vulnerability information or critical-service exposure. This makes the method simple and computationally inexpensive, but it may spend the mutation budget on hosts that are not the most dangerous from the DDoS-defense perspective. This behavior is consistent with random host-shuffling mechanisms used in earlier MTD-based DDoS protection studies [16, 17].

Exposure-only MTD improves over random selection by considering how strongly a host is connected to critical services. Therefore, hosts with more service reachability are treated as more important mutation candidates. This is useful because a host that can reach multiple protected servers may provide the attacker with more opportunities to contribute to service failure. However, exposure-only selection does not directly consider whether the host is easy to compromise. As a result, it may select structurally exposed hosts even when their compromise likelihood is relatively low. This limitation is related to connection-, traffic-, or service-exposure-driven MTD mechanisms [22, 24].

Vulnerability-only MTD follows the opposite logic. It prioritizes hosts with higher compromise likelihood, which is useful when the defender wants to reduce the number of easily exploitable attack sources. Nevertheless, vulnerability alone is not sufficient for DDoS-oriented host mutation. A highly vulnerable host may have limited reachability to critical servers and may therefore contribute less to coordinated service disruption. This explains why vulnerability-aware methods, including CVSS- or attack-graph-guided approaches, can still miss important exposure effects if service reachability is not jointly considered [18, 20, 21].

In contrast, the proposed COVER-MTD jointly considers both dimensions. A host receives high mutation priority only when it is both sufficiently vulnerable and structurally exposed to critical services. This joint criterion allows COVER-MTD to focus the mutation budget on hosts that are more useful to the attacker for DDoS propagation. Consequently, COVER-MTD can remove or mutate more security-critical host-service paths than single-factor baselines. The illustrative behavior in Fig. 8 therefore supports the main intuition of the proposed framework: effective host mutation should not be based only

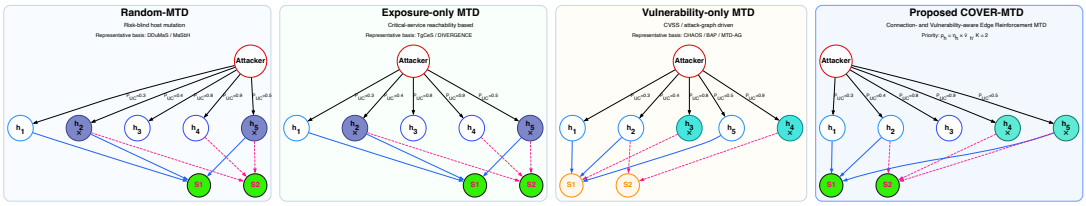


Fig. 8. Comparison of Random-MTD, Exposure-only MTD, Vulnerability-only MTD, and the proposed COVER-MTD under the same host compromise.

on compromise likelihood or only on connectivity, but on their combined effect.

VIII. Conclusion

This paper presented COVER-MTD, a reinforcement-learning-guided host-mutation framework for DDoS-resilient edge-IoT services. The proposed method addresses the limitation of single-factor mutation strategies by combining host vulnerability with critical-service exposure. This joint view allows the defender to prioritize hosts that are not only easy to compromise but also useful for attacking protected servers. COVER-MTD formulates host prioritization as a pairwise ranking problem. The agent learns local comparison decisions between two hosts rather than relying on a fixed topology-dependent action space. This design supports policy reuse across networks with different host counts and makes the framework suitable for dynamic edge-IoT deployments. Simulation results in Mininet show that COVER-MTD reduces the number of crashed servers and the attack success rate compared with random, vulnerability-only, and exposure-only MTD strategies. The security gain is achieved while keeping delay, packet loss, and host-selection time close to baseline methods. Future work will investigate faster ranking mechanisms, adaptive mutation scheduling, and cost-aware protection against Economic Denial of Sustainability attacks.

Acknowledgment

The work in this paper was supported in part by the Federal Ministry of Research, Technology, and Space (BMFTR), Germany, through the Project 6GEM+ under Grant 16KIS2411; and in part by the European Union through the 6G-Path project under Grant 101139172.

References

[1] A. Javadpour, F. Ja'fari, C. Benzaïd, and T. Taleb, "An optimized reinforcement learning based mtd mutation strategy for securing edge iot against ddos attack," *Journal of Information Security and Applications*, vol. 93, p. 104138, 2025.

[2] A. Javadpour, F. Ja'fari, T. Taleb, M. Shojafar, and C. Benzaïd, "A comprehensive survey on cyber deception techniques to improve honeypot performance," *Computers & Security*, p. 103792, 2024.

[3] A. Javadpour, F. Ja'fari, T. Taleb, F. Turkmen, and C. Benzaïd, "Beyond reinforcement learning for network security: A comprehensive survey and tutorial," *Journal of Information Security and Applications*, vol. 96, p. 104294, 2026.

[4] A. Javadpour, F. Ja'fari, T. Taleb, M. Shojafar, and B. Yang, "Scema: an sdn-oriented cost-effective edge-based mtd approach," *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 667–682, 2022.

[5] A. Javadpour, F. Ja'fari, T. Taleb, and M. Shojafar, "A cost-effective mtd approach for ddos attacks in software-defined networks," in *GLOBECOM 2022-2022 IEEE Global Communications Conference*. IEEE, 2022, pp. 4173–4178.

[6] J.-H. Cho, D. P. Sharma, H. Alavizadeh, S. Yoon, N. Ben-Asher, T. J. Moore, D. S. Kim, H. Lim, and F. F. Nelson, "Toward proactive, adaptive defense: A survey on moving target defense," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 1, pp. 709–745, 2020.

[7] A. Javadpour, F. Ja'fari, T. Taleb, M. Shojafar, and B. Yang, "Scema: an sdn-oriented cost-effective edge-based mtd approach," *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 667–682, 2022.

[8] A. Javadpour, F. Ja'Fari, T. Taleb, and C. Benzaïd, "Moving target defense for ddos mitigation with shuffling of critical edge (s) connections," *Journal of Information Security and Applications*, vol. 97, p. 104347, 2026.

[9] A. Javadpour, P. Pinto, F. Ja'fari, and W. Zhang, "Dmaidps: a distributed multi-agent intrusion detection and prevention system for cloud iot environments," *Cluster Computing*, vol. 26, no. 1, pp. 367–384, 2023.

[10] M. Rawski, S. Kukliński, P. Sapiecha, M. Pelka, G. Przytuła, P. Wojs, K. Szczypiorski *et al.*, "Mmtd: Mano-based moving target defense for corporate networks," in *2020 World Conference on Computing and Communication Technologies (WCCCT)*. IEEE, 2020, pp. 79–87.

[11] Z. Karim, A. Sebbar, Y. Baddi, and M. Boulmalf, "Secure multipath mutation smpm in moving target defense based on sdn," *Procedia Computer Science*, vol. 151, pp. 977–984, 2019.

[12] A. Aydeger, M. H. Manshaei, M. A. Rahman, and K. Akkaya, "Strategic defense against stealthy link flooding attacks: A signaling game approach," *IEEE Transactions on Network Science and Engineering*, 2021.

[13] C. Xu, T. Zhang, X. Kuang, Z. Zhou, and S. Yu, "Context-aware adaptive route mutation scheme: a reinforcement learning approach," *IEEE Internet of Things Journal*, vol. 8, no. 17, pp. 13 528–13 541, 2021.

[14] Z. Liu, Y. He, W. Wang, S. Wang, X. Li, and B. Zhang, "Aeh-mtd: Adaptive moving target defense scheme for sdn," in *2019 IEEE International Conference on Smart Internet of Things (SmartIoT)*. IEEE, 2019, pp. 142–147.

[15] B. Potteiger, A. Dubey, F. Cai, X. Koutsoukos, and Z. Zhang, "Moving target defense for the security and resilience of mixed time and event triggered cyber–physical systems," *Journal of Systems Architecture*, vol. 125, p. 102420, 2022.

[16] J. Steinberger, B. Kuhnert, C. Dietz, L. Ball, A. Sperotto, H. Baier, A. Pras, and G. Dreo, "Ddos defense using mtd and sdn," in *NOMS 2018-2018 IEEE/IFIP Network Operations and Management Symposium*. IEEE, 2018, pp. 1–9.

[17] X. Luo, Q. Yan, M. Wang, and W. Huang, "Using mtd and sdn-based honeypots to defend ddos attacks in iot," in *2019 Computing, Communications and IoT Applications (ComComAp)*. IEEE, 2019, pp. 392–395.

[18] Y. Shi, H. Zhang, J. Wang, F. Xiao, J. Huang, D. Zha, H. Hu, F. Yan, and B. Zhao, "Chaos: An sdn-based moving target defense system," *Security and Communication Networks*, vol. 2017, 2017.

[19] A. Chowdhary, S. Pisharody, and D. Huang, "Sdn based scalable mtd solution in cloud network," in *Proceedings of the 2016 ACM Workshop on Moving Target Defense*, 2016, pp. 27–36.

[20] S. Yoon, J.-H. Cho, D. S. Kim, T. J. Moore, F. Free-Nelson, and H. Lim, "Attack graph-based moving target defense in software-defined networks," *IEEE Transactions on Network and Service Management*, vol. 17, no. 3, pp. 1653–1668, 2020.

[21] Q. Li and J. Wu, "Optimizing the effectiveness of moving target defense in a probabilistic attack graph: A deep reinforcement learning approach," *Electronics*, vol. 13, no. 19, p. 3855, 2024.

[22] Y. Zhou, G. Cheng, S. Jiang, Y. Zhao, and Z. Chen, "Cost-effective moving target defense against ddos attacks using trilateral game and multi-objective markov decision processes," *Computers & Security*, vol. 97, p. 101976, 2020.

[23] T. Eghtesad, Y. Vorobeychik, and A. Laszka, "Deep reinforcement learning based adaptive moving target defense," *arXiv preprint arXiv:1911.11972*, 2019.

[24] S. Kim, S. Yoon, J.-H. Cho, D. S. Kim, T. J. Moore, F. Free-Nelson, and H. Lim, "Divergence: Deep reinforcement learning-based adaptive traffic inspection and moving target defense countermeasure framework," *IEEE Transactions on Network and Service Management*, 2022.

[25] Y. Zhou, G. Cheng, Z. Ouyang, and Z. Chen, "Resource-efficient low-rate ddos mitigation with moving target defense in edge clouds," *IEEE Transactions on Network and Service Management*, vol. 22, no. 1, pp. 168–186, 2025.

[26] Swati, S. Roy, J. Singh, and J. Mathew, "Securing iiot systems against ddos attacks with adaptive moving target defense strategies," *Scientific Reports*, vol. 15, p. 9558, 2025.

[27] A. Aydeger, P. Zhou, S. Hoque, M. Carvalho, and E. Zeydan, "Mtdns: Moving target defense for resilient dns infrastructure," *arXiv preprint arXiv:2410.02254*, 2024.

[28] A. Andreou, C. X. Mavromoustakis, E. Markakis, A. Bourdena, and G. Mastorakis, "Enhancing network slice security with deep reinforcement learning and moving target defense strategies," *Discover Internet of Things*, vol. 5, p. 67, 2025.