Moving Target Defense in 5G and Beyond Networks: A Comprehensive Survey and Research Directions

Amir Javadpour, Forough Ja'fari, Tarik Taleb, and Chafika Benzaïd

***Abstract*—5G and beyond networks rely on SDN/NFV, network slicing, and multi-access edge computing (MEC) to support highly heterogeneous and mission-critical services; however, the same capabilities enlarge the attack surface and make static protection policies increasingly brittle under fast reconfiguration and adversarial adaptation. Moving Target Defense (MTD) is therefore particularly significant in this setting because it can continuously reshape exposed assets, communication paths, and virtualized functions to invalidate reconnaissance, reduce attacker dwell time, and raise the cost of exploitation. This paper presents a deployment-oriented survey of MTD for 5G, with special emphasis on learning-assisted designs, especially reinforcement learning (RL), that optimize what to mutate, when to mutate, and how to enforce mutation under slice-isolation, URLLC, and orchestration constraints. The reviewed literature is organized by research category and by 5G operational domain, and then linked to mutation targets, timing policies, SDN/NFV implementation components, and the attacks they are designed to mitigate. Beyond cataloguing prior work, the survey clarifies the trade-offs among security gain, reconfiguration overhead, policy stability, scalability, and reproducibility, and highlights benchmark, secure-learning, and deployment-realism gaps that currently limit operational adoption. The resulting taxonomy and synthesis provide a clearer foundation for designing robust, low-overhead, and AI-assisted MTD mechanisms for 5G and emerging 6G networks.**

***Index Terms*—Moving Target Defense; 5G/6G Security; SDN/NFV; Network Slicing; MEC/Edge; URLLC; Reinforcement Learning; Machine Learning; Artificial Intelligence; Network Security**

I. Introduction

MOVING target defense (MTD) intentionally changes selected system attributes so that attackers cannot rely on stable reconnaissance results or long-lived exploitation paths. Instead of defending a fixed attack surface, the defender continuously perturbs exposure points such as addresses, ports, routes, service placements, and virtualized resources. This is especially relevant to 5G and beyond networks, where SDN/NFV softwarization, network slicing, and MEC create flexible but highly dynamic infrastructures whose attack surfaces can be learned and abused if they remain predictable.

Most adversaries begin with reconnaissance to discover reachable hosts, open services, software fingerprints, and feasible lateral-movement paths. Once this information is collected, attacks such as scanning, DDoS preparation, route manipulation, or targeted compromise become easier to execute. MTD weakens this attack cycle by forcing the attacker to repeatedly re-discover the environment, thereby increasing uncertainty, operational cost, and time-to-success. In practice, the value of MTD in 5G is not only security improvement, but also its ability to operate as a proactive control layer that can be coordinated with SDN, orchestration, slicing, and edge management.

An illustrative example of how an MTD approach works is shown next. A sample network is shown in Figure 1, where six hosts are connected to a critical server, and the adversary wants to first compromise the hosts and then launch a denial of service (DoS) attack against the critical server using them.

The MTD approaches are concerned with one or multiple of the following questions.

- *What has to be moved?* The answer to this question specifies the targets.
- *Which targets have to be moved?* The answer to this question specifies the parameters.
- *When to move the targets?* The answer to this question specifies the moving intervals.
- *How to move?* The answer to this question specifies the moving implementation.

A. Contributions and novelty

Although several surveys discuss Moving Target Defense (MTD) in enterprise, cloud, or generic SDN settings, 5G introduces distinct operational domains (RAN, core, slicing, and MEC/edge) and strict service constraints that materially change the feasibility and impact of mutation actions. The present manuscript is intentionally centered on *5G-oriented MTD*; machine learning and reinforcement learning are treated as enabling mechanisms for adaptive mutation decisions rather than as an independent survey topic. Our first contribution is therefore a clearer scope definition that keeps the narrative focused on how MTD is designed, deployed, and evaluated in softwarized 5G systems.

Second, the survey organizes the literature through a consistent two-level logic. We classify prior studies into theoretical/framework, application-specific, RL-based, and survey/review categories, and then remap them to practical 5G domains and implementation layers. This allows us to connect *what* is mutated (targets), *how* actions are selected (parameters

Amir Javadpour (Senior Cybersecurity Researcher MOSA!C Lab / ICTFICIAL Oy, Finland). **Forough Ja'fari** (Sharif University of Technology). **Tarik Taleb** (Faculty of Electrical Engineering and Information Technology, Ruhr University Bochum, Bochum). **Chafika Benzaïd** (Faculty of Information Technology and Electrical Engineering, University of Oulu)

Corresponding author: Amir Javadpour (**a.javadpour87@gmail.com**)

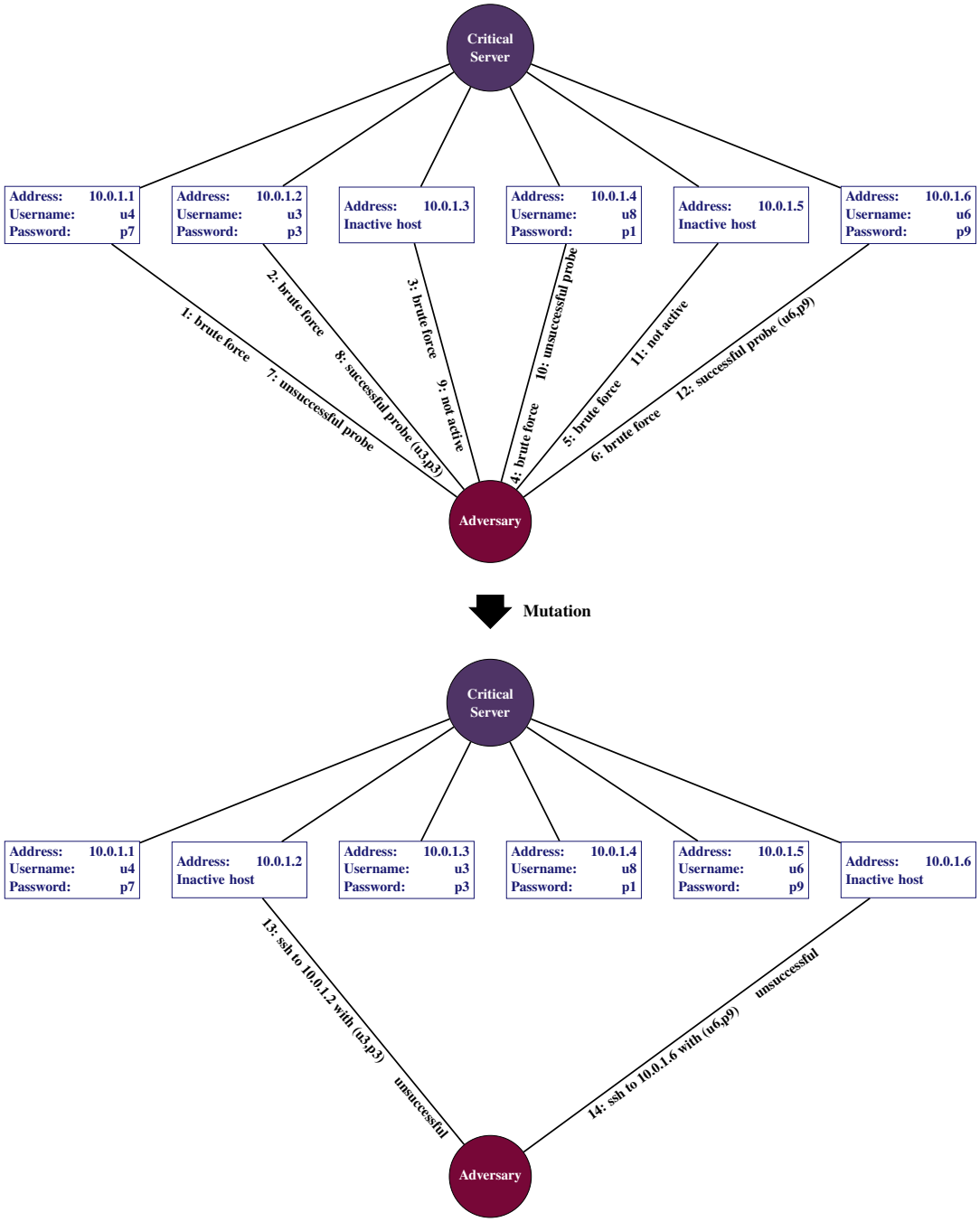
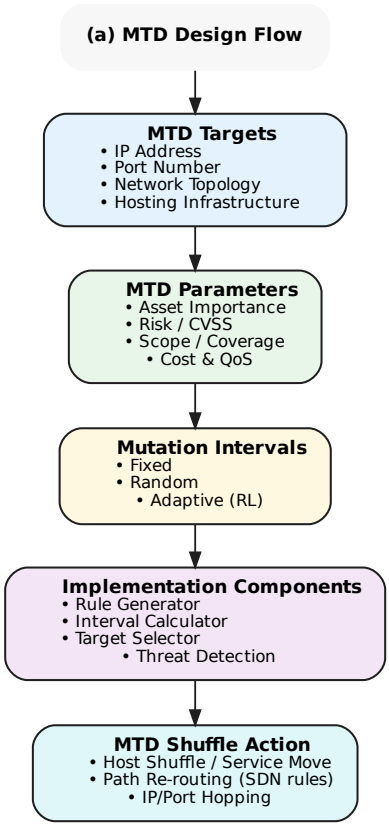


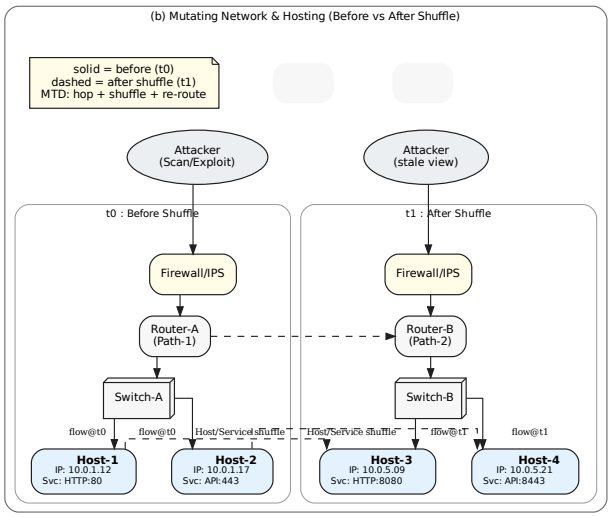
Fig. 1. Illustrative example of Moving Target Defense (MTD) through target mutation in a small network hosting a *critical server*. In the upper part (before mutation), an adversary probes multiple reachable hosts (shown with IP addresses and service credentials) and attempts to discover valid access paths; some probes fail while a subset may temporarily appear successful when the adversary targets an active host with matching credentials. The defense then triggers a *mutation* event (middle arrow), which changes the exposed configuration seen by the attacker e.g., swapping host roles (active vs. inactive/decoy), rotating or reassigning credentials, and/or remapping reachable endpoints. In the lower part (after mutation), the attacker's previously learned information becomes stale: hosts that were previously exploitable may become inactive or decoys, credentials and access mappings no longer hold, and subsequent probing attempts are rendered unsuccessful. This figure highlights the core MTD principle: continuously shifting the attack surface to disrupt reconnaissance-to-exploitation workflows and to increase attacker cost while preserving the protected critical service. (more detail in Fig. A in Appendix A.)

and decision logic), *when* mutations occur (interval and triggering strategies), and *how* the policy is enforced (SDN/NFV monitoring, control, and orchestration) to the classes of attacks being mitigated. As a result, heterogeneous proposals can be compared within one coherent survey structure rather than as isolated additions.

Third, we strengthen the analytical depth of the manuscript by synthesizing RL-assisted MTD as a closed-loop decision problem under 5G constraints. Instead of listing algorithms only descriptively, we examine how state/action/reward design, safety constraints, and mutation overhead align, or fail to align, with per-slice SLA preservation, URLLC latency/jitter sensi-



(a) MTD design flow



(b) Network-level shuffle & re-routing

Fig. 2. End-to-end survey workflow for Moving Target Defense (MTD) in SDN/NFV-based 5G systems. Panel (a) summarizes the MTD pipeline from *targets* (IP/port/topology/hosting) to *selection parameters* (asset importance, CVSS-based risk, scope/coverage, and cost–QoS trade-off), *mutation intervals* (fixed/random/adaptive), and *implementation components* (rule generation, interval calculation, target selection, and threat detection) culminating in the shuffle action. Panel (b) visualizes the core MTD principle at the network layer: the defender shuffles hosts/services and re-routes traffic paths via SDN control from t_0 to t_1, making reconnaissance stale and reducing exploit success, thereby mitigating DoS/DDoS, scanning, false data injection, and route manipulation.

tivity, scalability limits, and policy stability. Finally, the survey consolidates foundational and recent literature, highlights benchmark and reproducibility gaps, and outlines deployment-realistic research directions for robust, low-overhead, and AI-assisted MTD in 5G and emerging 6G networks.

B. Motivation

The evolution of 5G networks has brought significant improvements in bandwidth, latency, and connectivity, enabling mission-critical applications such as autonomous driving, industrial automation, and remote surgery. However, the complexity and openness of 5G architectures, driven by the integration of software-defined networking, network function virtualization, network slicing, and edge computing, have also expanded the attack surface, making them increasingly attractive targets for sophisticated cyber threats. Traditional static security configurations are inadequate in such dynamic environments, as they allow attackers sufficient time to carry out reconnaissance, exploit vulnerabilities, and execute persistent attacks. MTD has emerged as a promising paradigm for countering these threats by dynamically and unpredictably altering system configurations, thereby increasing the uncertainty and cost for adversaries [1]. Recent advancements in reinforcement learning and AI-driven decision-making have further enhanced the adaptability and efficiency of MTD mechanisms, motivating their integration into 5G security frameworks.

C. 5G-specific drivers and constraints for MTD

While MTD has been investigated in enterprise and cloud networks, 5G introduces architectural features and service requirements that make MTD both more necessary and more challenging. In 5G, security mechanisms must coexist with softwarized control (SDN/NFV), dynamic service composition, and heterogeneous deployment across RAN, core, and edge. As a result, MTD decisions cannot be assessed only by "security gain"; they must also be evaluated against operational constraints and service-level objectives.

a) Network slicing and SLA preservation.: A defining capability of 5G is *network slicing*, where multiple logical networks share the same physical infrastructure. From an MTD perspective, mutations such as routing perturbation, VNF re-placement, or address/identity hopping must preserve slice isolation and avoid cross-slice side effects. Moreover, each slice is governed by its own SLA/KPIs (e.g., latency, throughput, reliability), meaning that an MTD action that is beneficial for one slice may be unacceptable for another. Therefore, 5G-oriented MTD requires *per-slice* awareness and policies that explicitly account for isolation, resource sharing, and KPI compliance.

b) URLLC latency budgets and stability requirements.: URLLC services impose strict end-to-end latency and jitter constraints. Aggressive or frequent mutations may introduce transient packet loss, routing convergence delay, VNF cold-start time, or control-plane signaling overhead, all of which can violate URLLC guarantees. Hence, 5G deployments motivate *timing-aware* MTD strategies where mutation frequency,

scope, and rollback mechanisms are carefully designed. In practice, the feasibility of an MTD action depends not only on its security effect but also on its *reconfiguration cost* and *stability* (e.g., avoiding oscillatory policies).

c) Edge/MEC constraints and distributed orchestration.: 5G increasingly relies on MEC to place functions close to users for low latency. However, edge nodes often have constrained compute/storage, limited redundancy, and intermittent backhaul capacity. These conditions restrict the complexity of decision-making (e.g., heavy centralized optimization) and the overhead tolerated for reconfiguration. Furthermore, MEC introduces a distributed orchestration plane, where coordination delays and partial observability can limit how fast and how safely MTD can be applied. Consequently, MTD for 5G must be *resource-aware* and robust to partial information, especially for learning-assisted approaches.

d) Softwarized control: SDN/NFV overhead, convergence, and scale.: SDN/NFV enables flexible reconfiguration but also creates practical bottlenecks that become prominent under MTD. Frequent updates can trigger control-plane churn (e.g., repeated flow-rule updates), stress limited hardware resources (e.g., TCAM), and lead to configuration convergence issues. At scale, mutation-induced updates may increase policy conflicts and amplify instability, particularly when multiple controllers and domains are involved. Therefore, scalable 5G MTD must explicitly consider operational limits such as rule-space constraints, signaling overhead, convergence time, and multi-domain coordination.

e) Implications for MTD design and for this survey.: These 5G-specific realities motivate *constraint-aware and deployment-realistic* MTD designs that jointly optimize security improvement and service impact. In this survey, we use the above constraints to 1) structure the taxonomy by *domain* (RAN, core, slicing, MEC/edge), 2) highlight which mutation targets are practical under URLLC/MEC limitations, and 3) compare learning-based approaches not only by their threat mitigation performance but also by their overhead, stability, and scalability under 5G operational conditions. This perspective also clarifies why benchmark gaps and realistic evaluation settings are critical for progress in 5G MTD research.

D. Problem Statement and Goals

Moving Target Defense (MTD) is a promising proactive mechanism for strengthening the resilience of 5G networks; however, its deployment in production-grade 5G systems remains challenging. Practical implementations must operate across heterogeneous domains (RAN, core, slices, and edge/MEC) and within SDN/NFV-based control and orchestration, while respecting strict service requirements such as per-slice SLA compliance and URLLC latency/jitter constraints. Existing studies are often evaluated under simplified assumptions and may suffer from high reconfiguration and computational overhead, suboptimal mutation scheduling, limited interoperability across domains, and insufficient integration with slicing and orchestration. In addition, many learning-based MTD strategies rely on simulated environments and synthetic datasets that may not capture real traffic variability and attacker behavior, which limits generalizability. The lack of standardized interfaces and benchmark datasets further impedes fair comparison and large-scale adoption.

Given these gaps, the goal of this survey is to provide a structured and 5G-oriented analysis of MTD techniques, with an emphasis on learning-assisted (particularly reinforcement learning) and software-defined implementations. Specifically, we 1) classify and map existing MTD approaches to major 5G domains, including core networks, radio access networks, critical servers, end devices, and network slicing; 2) summarize the attack classes addressed by MTD (e.g., DoS/DDoS, scanning/reconnaissance, false data injection, and route manipulation) and discuss how different mutation strategies disrupt their underlying mechanisms; and 3) identify the key operational components required for deployment in SDN/NFV-enabled 5G infrastructures, such as rule generation, interval calculation, target selection, and threat detection. By consolidating fragmented findings and highlighting practical constraints and evaluation gaps, this survey aims to guide both researchers and practitioners toward low-latency, resource-efficient, and interoperable MTD designs for 5G and beyond.

Although optimization-based approaches also exist (e.g., Zhou et al. [2]), we focus primarily on learning-assisted MTD because it can adapt mutation decisions to evolving threats and non-stationary network conditions, and can explicitly optimize security QoS overhead trade-offs when designed with deployment constraints in mind.

E. Threat Model and Assumptions

We assume an active network adversary capable of reconnaissance (scanning and fingerprinting), volumetric and protocol-aware DoS/DDoS attacks, and limited lateral movement after foothold acquisition. The attacker may be off-path or gain partial on-path visibility through compromised hosts. The defender controls the SDN/NFV infrastructure (controller, orchestrator) and can mutate addresses, ports, routes, functions/VM placements, and slice configurations within policy and resource limits. Control channels (controller–switch and orchestrator–node) are trusted and authenticated; end-user cryptographic traffic remains opaque to the defender. Time is slotted at sub-second to multi-second granularity; detection signals may arrive with bounded delay and noise. Finally, service-level objectives (SLOs) constrain mutation frequency, allowable packet loss during cutover, and migration downtime.

To provide a clear overview of the organization and logical flow of this survey, Fig. I illustrates the proposed MTD framework adopted in this work. The framework begins by identifying the primary **MTD targets** in 5G networks, followed by the **selection parameters** that guide which assets are chosen for mutation. These parameters inform the design of **mutation intervals**, which define when and how frequently mutations are applied. The next stage focuses on the **implementation components** required to realize the selected MTD strategies in practice. Finally, the framework maps these defensive measures to the **types of attacks** they can effectively mitigate. This step-by-step representation not only summarizes the structure of our survey but also highlights the logical progression from threat surface identification to proactive defense mechanisms.

F. Scope and Narrative of the Survey

To keep the manuscript coherent, this survey follows one central narrative: MTD in 5G is treated as a deployment-constrained, closed-loop security-control problem. Accordingly, the paper does not attempt to become a generic survey of all 5G security mechanisms or all machine-learning methods for networking. Instead, ML and RL are discussed only insofar as they help decide mutation targets, timing, and enforcement under 5G-specific constraints.

The manuscript is therefore organized in a progressive way. We first position prior work and identify the main strands of the literature. We then analyze the core MTD design dimensions–targets, parameters, intervals, and implementations–before discussing attacks, evaluation methodology, reproducibility, deployment challenges, and future directions. This structure is meant to maintain a single survey identity throughout the paper: a 5G-oriented synthesis of MTD design, practical enforcement, and learning-assisted adaptation.

II. Review of Existing MTD Research

This section reviews the literature using a single organizing principle to avoid overlap and fragmentation. We begin with foundational surveys that define the architectural and practical design space of MTD, then move to telecom- and 5G-oriented studies, and finally examine learning-assisted proposals that optimize mutation decisions under dynamic conditions. This progression keeps the discussion centered on 5G-oriented MTD rather than treating MTD, generic 5G security, and machine learning as disconnected themes.

Foundational survey papers establish the architectural basis and practical taxonomy of MTD [3, 4, 5]. More recent works extend this perspective toward cloud/edge telco settings, secure network slicing, and 6G-oriented zero-trust integration, showing that MTD is increasingly being discussed in environments that closely match the operational assumptions of modern softwarized mobile networks [6, 7, 8]. Against this backdrop, our review highlights not only what has been proposed, but also how consistently those proposals address deployment realism, mutation cost, orchestration complexity, and reproducible evaluation.

Recent surveys have examined complementary slices of the MTD literature, including AI-oriented MTD overviews, game-theoretic strategy selection, domain-specific reviews for power grids and SCADA systems, and broader discussions of proactive/adaptive defense [9, 10, 11, 4, 12, 13]. Additional review articles focus on machine-learning-assisted MTD, AI-enhanced MTD for IoT, and the broader role of artificial intelligence in adaptive defense [14, 15, 16].

Taken together, these surveys confirm the maturity of the broader MTD field, but they also reveal a persistent gap: the literature still lacks a sufficiently integrated, 5G-centered synthesis that simultaneously addresses telecom architecture, slice-aware constraints, SDN/NFV enforcement, adaptive decision-making, and deployment realism. This gap motivates the remainder of our review.

Zhang et al. [17] have proposed an MTD solution for digital twin mobile networks. To protect productive servers in a software-defined network against DDoS attacks, Ribeiro et al. [18] have proposed an MTD approach that uses machine learning first to classify the network traffic and then forward the malicious requests toward a secondary server. To optimize MTD solutions in terms of balancing between their cost and effectiveness, Li and Wu [19] have mathematically formulated this optimization problem through an attack graph and then solved it using two deep reinforcement learning models. To improve the performance of MTD solutions that are based on the host address mutation technique, Zhang et al. [20] have proposed a method based on Advantage Actor-Critic (A2C) algorithm. MTD is modeled as a two-player game by Eghtesad et al. [21] and a reinforcement learning algorithm is proposed for finding the winning strategies of this game. The MTD technique in this game is to reimage the servers, which means to create a new image of that server and resetting it by operations like reinstalling its operating system and changing its configuration. Soussi et al. [22] have proposed an MTD-based architecture for beyond 5G networks to show how this security technique can be integrated with such networks. For deciding the optimal strategy, the authors have utilized machine learning algorithms for the MTD technique. Chowdhary et al. [23] have compared their work with Eghtesad et al. [21]. In this work, multi-agent reinforcement learning is used to find the winning strategy of the MTD game. Yoon et al. [24] have proposed a secure framework, called DESOLATER, for vehicular networks, in which an MTD approach utilizing a multi-agent reinforcement learning model is adopted. Chai et al. [25] have provided an MTD method for securing networks against DDoS attacks, called DQ-MOTAG. This method utilizes reinforcement learning to determine the optimal shuffling period. Gao and Wang [26] have proposed an MTD solution based on reinforcement learning. Li and Zheng [27] have proposed a two-stage reinforcement learning-based MTD solution, called meta-RL. Beyond the representative studies discussed above, the recent learning-assisted MTD literature has expanded toward CNN-supported detection and shuffling, federated and multi-agent reinforcement learning, Bayesian and topology-aware mutation design, closed-loop NFV/edge orchestration, physically informed adaptation, and security-of-learning-aware formulations [28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52]. Collectively, these works demonstrate the breadth of adaptive MTD design, but they also make clear that assumptions about telemetry quality, reward construction, action cost, and telecom deployment constraints vary substantially across papers. This heterogeneity is one of the main reasons a consolidated and explicitly 5G-oriented survey remains necessary.

N: the total number of nodes in the network I: the total number of available IP addresses S: the total number of slices

The core RL formulation (state/action/reward) used in Zhang et al. [20] is summarized in Table I.

The MTD RL formulation and reward design of Eghtesad et al. [21] are summarized in Table II.

Table III summarizes the SDN-based MARL MTD formulation (state/action/reward) in Chowdhary et al. [23].

The slice-aware MARL formulation proposed in Yoon et al. [24] is summarized in Table IV.

TABLE I
MODEL SUMMARY OF ZHANG ET AL. [20]

Environment State
$\mathcal{E} = (m_1, m_2, \ldots, m_N)$ $m_i = 0$ if the ith node is static (not mutated), otherwise $m_i = 1$.
Action Space
$\mathcal{A} = (a_{i,j})_{1 \le i \le N, 1 \le j \le I}$ $a_{i,j} = 1$ if IP address j is assigned to node i, otherwise 0.
Reward Function
$\mathcal{R} = \begin{cases} \alpha, & \text{if no node is scanned,} \\ -\beta \sum_{i=1}^{N} sc_i, & \text{if scanning is detected,} \end{cases}$ where sc_i is the number of scans on node i, and $\alpha, \beta > 0$.

Note: N is the number of nodes and I is the size of the available IP pool. The state vector $\mathcal{E}$ indicates whether each node is mutated, while the action space $\mathcal{A}$ encodes IP assignments. The reward encourages scan-free epochs and penalizes scanning activity.

TABLE II
MODEL SUMMARY OF EGHTESAD ET AL. [21]

Environment State
$\mathcal{E} = \big((x_1, y_1, z_1), (x_2, y_2, z_2), \ldots, (x_N, y_N, z_N)\big)$ • x_i: number of probes observed against node i since its last reimage. • $y_i \in \{\text{Defender}, \text{Adversary}\}$: controller of node i (whether compromised or not). • z_i: number of time steps node i has been offline due to reimaging.
Action Space
$\mathcal{A} = \{a \mid a \in \{1, \ldots, N\}\}$ a denotes the index of the node selected for reimaging at the current step.
Reward Function
$\mathcal{R} = \begin{cases} +\alpha, & \text{If the reimage restores a compromised node to control.} \\ -\beta, & \text{Reimaging may cause downtime without stop an attack.} \\ -\gamma, & \text{if an uncompromised node is unnecessarily reimaged.} \end{cases}$ where $\alpha, \beta, \gamma > 0$ represent defender's gain for recovery, cost of reimaging overhead, and penalty for unnecessary downtime, respectively.

Note: The state vector encodes probe history, compromise status, and reimaging downtime for each node. The action space is the set of nodes eligible for reimaging. The reward balances the benefit of restoring control with the operational cost of reimaging.

TABLE III
MODEL SUMMARY OF CHOWDHARY ET AL. [23]

Environment State
$\mathcal{E} = \text{AdversaryAccess}$ A scalar or vector that represents the estimated level of control or access the adversary currently has within the network (e.g., number of compromised flows, VMs, or hosts).
Action Space
$\mathcal{A} = \{\text{No action}, \text{VM migration}, \text{IP mutation}\}$ • *No action*: leave system unchanged. • *VM migration*: move selected virtual machine(s) to another host. • *IP mutation*: shuffle or reassign IP addresses of target nodes.
Reward Function
$\mathcal{R} = \begin{cases} +\alpha, & \text{if adversary's access decreases after the action,} \\ -\beta, & \text{if adversary's access increases (attack success),} \\ -\gamma, & \text{if the action causes overhead without reducing access.} \end{cases}$ where $\alpha, \beta, \gamma > 0$ represent security gain, security loss, and operational overhead, respectively.

Note: The reward encourages actions that minimize adversary access while penalizing both attack success and excessive migration/shuffling overhead.

TABLE IV
MODEL SUMMARY OF YOON ET AL. [24]

Environment State
$\mathcal{E} = \big((o_1, l_1, d_1, v_1), \ldots, (o_S, l_S, d_S, v_S)\big)$ • o_i: maximum traffic load observed in slice i. • l_i: cumulative packet loss in slice i. • d_i: average packet drop rate due to IP shuffling in slice i. • v_i: vulnerability score of slice i.
Action Space
$\mathcal{A} = (b, t)$ • b: allocated link bandwidth for the selected slice. • t: IP shuffling interval assigned to the slice.
Reward Function
$\mathcal{R} = \underbrace{+\alpha \cdot (1 - d_i - l_i)}_{\text{QoS gain}} - \underbrace{\beta \cdot v_i}_{\text{security risk}} - \underbrace{\gamma \cdot C(b,t)}_{\text{overhead cost}},$ where: • $\alpha > 0$: weight for service quality (minimizing loss/drop). • $\beta > 0$: weight for slice vulnerability penalty. • $\gamma > 0$: weight for mutation overhead. • $C(b, t)$: operational cost of allocating bandwidth b and applying shuffling with interval t.

Note: The reward promotes high QoS (low packet loss/drop), discourages vulnerable slice configurations, and penalizes unnecessary cost from frequent shuffling or bandwidth reallocation.

Table V summarizes the DQ-MOTAG problem formulation (state/action/reward) used in Chai et al. [25].

Table VII summarizes the Meta-RL formulation and objective trade-offs described in Li and Zheng [27].

The RESONANT DRL formulation, including the model-switching action space and cost-aware reward, is summarized in Table VIII.

Table IX summarizes the resource-aware DRL formulation proposed in Zhou et al. [54].

Zhou et al. [54] considered a cloud/edge environment, where multiple services are handled with containers. To mitigate DDoS attacks against these services, an MTD approach is proposed. This approach performs three types of MTD methodologies, which are scaling up or down the services, creating or removing service replicas, and port hopping. To help the MTD solution decides the optimal strategy for each of these three methods, a deep reinforcement learning model is used. The service/container-level MTD formulation considered in Zhou et al. [54] is summarized in Table X.

To mitigate packet drop attacks in vehicular networks, Zhang et al. [44] have proposed a multi-agent reinforcement learning model that optimizes route mutation strategies. Each of the vehicles in the network has an agent to decide the best

TABLE V
MODEL SUMMARY OF CHAI ET AL. [25]

Environment State

$\mathcal{E} = (s_{i,j})_{1 \le i \le P,\, 1 \le j \le U}$

- U: total number of users.
- P: total number of proxies.
- $s_{i,j} = 1$ if user j is currently connected to proxy i, otherwise 0.

Action Space

$\mathcal{A} = \{T_1, T_2, \ldots, T_k\}$

- Each action corresponds to selecting a shuffling period T (time interval between consecutive proxy reassignments).
- The DQ-learning agent learns the optimal T to balance defense strength vs. overhead.

Reward Function

$$\mathcal{R} = \beta \cdot \left(1 - \frac{\sum_{i=1}^{P} G_i}{U}\right) \; - \; \delta \cdot C(T),$$

where:

- G_i: number of DDoS attack attempts observed at proxy i.
- $C(T)$: operational cost of using shuffling interval T (e.g., reconfiguration overhead, user session disruption).
- $\beta > 0$: weight for attack mitigation benefit.
- $\delta > 0$: weight for shuffling overhead penalty.

Note: The reward is maximized when the shuffling period T effectively reduces the success of DDoS attacks while keeping migration/reconfiguration overhead low.

TABLE VI
MODEL SUMMARY OF GAO AND WANG [26]

Environment State

$\mathcal{E} = (s_1, s_2, s_3, s_4, s_5)$

- $s_i = 1$ if the network experienced an attack within the past i cycles, otherwise 0.
- This captures short-term attack history for anticipating future adversarial behavior.

Action Space

$\mathcal{A} = \{\text{Defend}, \text{No defend}\}$

- *Defend*: trigger an MTD action (e.g., IP shuffling, route mutation, VM migration).
- *No defend*: leave the system in its current configuration.

Reward Function

$$\mathcal{R} = \underbrace{+\alpha \cdot B}_{\text{defense benefit}} \; - \; \underbrace{\beta \cdot C}_{\text{defense cost}} \; - \; \underbrace{\gamma \cdot L}_{\text{system loss}},$$

where:

- B: reduction in attack success rate (benefit of defense).
- C: operational overhead of applying MTD (e.g., latency, reconfiguration).
- L: residual system loss if an attack succeeds despite defense.
- $\alpha, \beta, \gamma > 0$: weighting coefficients.

Note: The reward encourages defense actions that minimize long-term attack success while balancing operational costs and avoiding unnecessary reconfigurations.

next-hop vehicle for transmitting data packets.

The comparative summary in Table XI highlights the diversity of RL-based MTD strategies in terms of modeling, learn-

TABLE VII
MODEL SUMMARY OF LI AND ZHENG [27]

Environment State

$\mathcal{E}$ = Current system configuration

- Includes active VMs, network topology, IP/port assignments, and security status of nodes.
- Captures whether components are under attack or uncompromised.

Action Space

$\mathcal{A}$ = Next system configuration

- Choose the next configuration by mutating IPs, migrating VMs, or re-allocating resources.
- Each action transitions the system from $\mathcal{E}$ to a new configuration $\mathcal{E}'$.

Reward Function

$$\mathcal{R} = -\, C_{\text{mig}}(\mathcal{E}, \mathcal{E}') \; - \; \mu \cdot L(\mathcal{E}'),$$

where:

- C_{mig}: migration cost for reconfiguring from state $\mathcal{E}$ to $\mathcal{E}'$ (e.g., delay, packet drops, resource consumption).
- $L(\mathcal{E}')$: expected system loss if the new configuration is compromised.
- μ: estimated attack success probability under configuration $\mathcal{E}'$.

Note: The meta-RL agent learns policies that generalize across different environments, balancing defense effectiveness with migration overhead to achieve robust performance under diverse attack scenarios.

TABLE VIII
MODEL SUMMARY OF ABDEL MESSIH ET AL. [53]

Environment State

$\mathcal{E} = (M_1, M_2, \ldots, M_K)$

- M_k: performance and robustness indicators of classifier k (e.g., accuracy, false-positive rate, adversarial vulnerability).
- The state encodes the pool of available classifiers and their current trust/quality levels.

Action Space

$\mathcal{A} = \{$ select one classifier from the pool for this round $\}$

- At each decision epoch, the agent selects which classifier M_k to deploy.
- Possible actions: switch to a new model, keep the current model, or re-initialize a classifier.

Reward Function

$$\mathcal{R} = \underbrace{+\alpha \cdot \text{Acc}(M_k)}_{\text{classification accuracy}} \; - \; \underbrace{\beta \cdot \text{Vuln}(M_k)}_{\text{adversarial vulnerability}} \; - \; \underbrace{\gamma \cdot \text{Cost}(M_k)}_{\text{switching/overhead cost}},$$

where:

- $\text{Acc}(M_k)$: detection or classification accuracy of the chosen classifier.
- $\text{Vuln}(M_k)$: vulnerability score under adversarial/fraudulent inputs.
- $\text{Cost}(M_k)$: operational overhead of switching or retraining classifiers.
- $\alpha, \beta, \gamma > 0$: weighting factors balancing security, accuracy, and overhead.

Note: In RESONANT, the environment state represents the pool of candidate classifiers. The action is to pick one classifier for deployment in the current round, while the reward encourages high accuracy, low adversarial vulnerability, and minimal switching cost.

TABLE IX
MODEL SUMMARY OF ZHOU ET AL. [54]

Environment State

$$\mathcal{E} = \{(u_i, r_i, v_i) \mid i \in \mathbb{N}_S\}$$

- S: total number of slices.
- u_i: current utilization level of slice i (traffic load, CPU/memory usage).
- r_i: remaining resource quota for slice i (bandwidth, compute).
- v_i: vulnerability score of slice i based on threat intelligence and anomaly reports.

Action Space

$$\mathcal{A} = (a_i^{\text{mig}}, a_i^{\text{mut}})$$

- $a_i^{\text{mig}} \in \{0, 1\}$: migrate VNFs of slice i to another physical node or keep unchanged.
- $a_i^{\text{mut}} \in \{0, 1\}$: mutate IP/port of slice i or keep configuration.
- Joint action space allows hybrid adaptation (migration + mutation).

Reward Function

$$\mathcal{R} = \gamma \cdot G_{\text{sec}} - \lambda \cdot C_{\text{mig}} - \mu \cdot D_{\text{QoS}},$$

where:

- G_{sec}: security gain (reduction in attack surface or detected compromise).
- C_{mig}: migration overhead (latency, downtime, resource cost).
- D_{QoS}: service degradation caused by frequent reconfiguration.
- $\gamma, \lambda, \mu > 0$: weighting coefficients to balance security, overhead, and service quality.

Note: This formulation enables a DRL agent to dynamically trade off between **security improvement**, **migration overhead**, and **service quality**, making it directly applicable to multi-slice 5G/edge environments.

TABLE X
MODEL SUMMARY OF ZHOU ET AL. [54] (SERVICE/CONTAINER-LEVEL MTD)

Environment State

$$\mathcal{E} = \{(r_i, c_i, e_i, p_i) \mid i \in \mathbb{N}_S\}$$

- S: total number of services.
- r_i: binary indicator (1 if service i is running, 0 otherwise).
- c_i: number of active client connections to service i.
- e_i: resource consumption (CPU, memory, bandwidth) of service i.
- p_i: port number currently used by service i.

Action Space

$$\mathcal{A} = (a_i^{\text{sca}}, a_i^{\text{rep}}, a_i^{\text{hop}})$$

- $a_i^{\text{sca}} \in \{-1, 0, +1\}$: scale down, no change, or scale up service i.
- $a_i^{\text{rep}} \in \{-1, 0, +1\}$: remove replica, no change, or create a new replica of service i.
- $a_i^{\text{hop}} \in \{0, 1\}$: keep current port or change to a new port.

Reward Function

$$\mathcal{R} = \alpha \cdot G_{\text{sec}} + \beta \cdot Q_{\text{serv}} - \gamma \cdot C_{\text{scale}} - \delta \cdot C_{\text{hop}},$$

where:

- G_{sec}: security gain (e.g., reduced DDoS impact, attack evasion).
- Q_{serv}: service quality improvement (higher availability, throughput).
- C_{scale}: overhead of scaling or replicating services.
- C_{hop}: disruption cost due to port hopping (e.g., client reconnect delay).
- $\alpha, \beta, \gamma, \delta > 0$: weights balancing security, QoS, and overhead.

Note: The agent jointly optimizes scaling, replication, and port-hopping to maximize security and service quality while minimizing operational overhead. This makes the approach suitable for 5G/edge containerized environments.

ing algorithms, and operational focus. Lighter-weight solutions such as Gao and Wang [26] and Zhang et al. [20] provide interpretable formulations and demonstrate the effectiveness of RL in basic IP-shuffling or history-based defense scenarios. However, they lack the scalability and service-awareness required in real 5G deployments. Similarly, game-theoretic works such as Eghtesad et al. [21] establish solid theoretical foundations but are impractical for ultra-reliable low-latency communication (URLLC) due to the high downtime associated with reimaging.

On the other hand, frameworks explicitly designed around SDN and network slicing, including Chowdhary et al. [23], Yoon et al. [24], and Zhou et al. [54], show greater promise for 5G applicability. These methods consider security gain, operational overhead, and QoS degradation, enabling fine-grained decisions in multi-slice environments. Meta-RL approaches such as Li and Zheng [27] further extend adaptability by learning generalized policies across heterogeneous settings, which is particularly relevant for the transition toward 6G. In addition, the RESONANT framework proposed in Abdel Messih et al. [53] introduces a novel perspective by applying MTD concepts to classifiers themselves; while less directly related to network infrastructure, such methods are important for protecting ML-driven services in 5G (e.g., fraud detection, intrusion detection). So, the most suitable strategies for 5G are those that 1) integrate with SDN/NFV orchestration, 2) are slice-aware and resource-conscious, and 3) balance security effectiveness with SLOs.

A. Domain-guided organization of prior work

To improve the readability and navigability of this review, we organize prior MTD studies primarily by their deployment context rather than presenting them as a single long sequence. The reason is that MTD design choices in 5G-era systems are strongly shaped by where the defense is implemented and what operational constraints apply. Accordingly, we group the literature into domain-centric clusters that reflect the dominant control mechanisms and constraints: IoT and end-device environments, where limited resources and partial visibility motivate lightweight mutation strategies; SDN-enabled networks, where centralized programmability supports rapid reconfiguration but introduces control-plane overhead, rule-space limitations, and convergence stability concerns; cloud/edge/MEC and NFV-based infrastructures, where service placement, migration, and orchestration decisions directly reshape the attack surface while imposing QoS and state-transfer costs; and 5G slicing and multi-tenant settings, where slice isolation and SLA/KPI preservation constrain both target selection and mutation scheduling. This domain-guided structure reduces repetition, creates clearer transitions between themes, and provides a direct bridge to the 5G-oriented taxonomy and comparative analysis developed in the subsequent sections.

TABLE XI
SUMMARY OF RL-BASED MTD APPROACHES: MODELS, SAR DESIGN, STRENGTHS, WEAKNESSES, AND 5G DEPLOYMENT CONSIDERATIONS

Reference	RL Algorithm	State / Action / Reward	Strengths	Weaknesses	5G Suitability and deployment limits
Zhang et al. [20]	A2C	S: Node mutation vector (m_i); A: Assign IPs; R: $+\alpha$ if not scanned, $-\beta \sum sc_i$.	+ Direct defense against scanning + Low training complexity	– Ignores QoS impact – Limited scalability	Useful for RAN/edge IP-hopping; overhead grows in dense 5G. *Deployment limits:* endpoint/mapping synchronization and control-plane churn may affect URLLC.
Eghtesad et al. [21]	Q-learning (game)	S: Probes, compromise flag, downtime; A: Select node to reimage; R: Recovery benefit – reimage cost.	+ Game-theoretic modeling + Captures attacker–defender dynamics	– High downtime – Impractical for URLLC	Low applicability to real-time 5G due to reimaging latency. *Deployment limits:* service downtime and stateful VNF dependencies can violate per-slice SLAs.
Chowdhary et al. [23]	Multi-agent RL	S: Adversary access level; A: No action / VM migration / IP mutation; R: Security gain – migration/overhead.	+ Joint optimization (migration + shuffle) + SDN-native design	– Migration overhead – Multi-agent instability	Highly relevant for SDN-based 5G slicing. *Deployment limits:* VNF migration time/state transfer, controller overhead, and multi-agent stability must be bounded.
Yoon et al. [24]	Multi-agent RL	S: Slice traffic load, loss, drop, vuln.; A: Bandwidth allocation + shuffle interval; R: QoS gain – vulnerability – overhead.	+ Slice-aware + Integrates QoS/security	– High model complexity – Needs real traffic datasets	Promising for slice-aware control. *Deployment limits:* high computation and reliance on realistic telemetry/datasets; may lag under fast URLLC dynamics.
Chai et al. [25]	Deep Q-Learning	S: User–proxy connections; A: Choose shuffle period; R: Mitigation benefit – shuffle overhead.	+ Optimizes shuffle timing + Scalable to proxy networks	– Simplified traffic assumptions – Ignores service delay	Partially relevant; extendable to 5G proxy/gateway defense. *Deployment limits:* must model mobility/session continuity and include latency in the reward.
Gao and Wang [26]	Tabular RL	S: Attack history (past 5 cycles); A: Defend / No defend; R: Benefit – defense cost – system loss.	+ Simple, interpretable + History-based adaptation	– Binary action space – Too coarse for complex loads	Baseline only; not practical at 5G scale. *Deployment limits:* binary actions and limited state ignore multi-slice QoS/overhead trade-offs.
Li and Zheng [27]	Meta-RL (two-stage)	S: Current system config; A: Next config (VM/IP migration); R: – migration cost – μ·system loss.	+ Generalizes across domains + Learns robust policies	– Migration overhead – Needs large training data	Promising for adaptive 5G/6G slice management. *Deployment limits:* data-hungry; online adaptation needs safety constraints; migration overhead remains.
Abdel Messih et al. [53]	Deep RL (classifiers)	S: Pool of classifiers with performance/vulnerability; A: Select classifier; R: Accuracy – vulnerability – switching cost.	+ Novel MTD on ML models + Useful against adversarial ML/fraud	– Indirect to 5G infra – More ML-security oriented	Indirect for 5G infrastructure, but relevant for ML-based 5G services. *Deployment limits:* focuses on model selection; requires integration with network control loops.
Zhou et al. [54] (slice-level)	DRL (policy optimization)	S: Slice utilization u_i, quota r_i, vuln. v_i; A: Migrate VNFs, mutate IP/port; R: Security gain – migration – QoS degradation.	+ Joint slice/resource security + Aligns with NFV/MANO	– Training complexity – Migration bottlenecks	Very promising for containerized 5G core/edge. *Deployment limits:* MANO integration, migration bottlenecks, and policy stability must be evaluated.
Zhou et al. [54] (service-level)	DRL (multi-action)	S: Service status (r_i, c_i, e_i, p_i); A: Scale up/down, replicate, port-hop; R: Security + QoS – scaling/port cost.	+ Combines scaling + replication + MTD + Fine-grained service protection	– Overhead of frequent scaling – Port hopping may disrupt clients	Directly applicable for microservice-based 5G edge clouds. *Deployment limits:* scaling/replication overhead; port-hop requires client update and MEC capacity.

Comparative discussion (Table XI). While Table XI summarizes RL-based MTD formulations (state/action/reward), differences in reported performance are primarily driven by a few design factors. Approaches that encode *richer state information* (e.g., threat indicators together with traffic load and QoS context) typically learn more stable policies than security-only states, because the agent can avoid disruptive mutations during congestion or latency-critical periods. In addition, methods with *feasible and well-scoped action spaces* (e.g., slice-aware or domain-scoped mutations) tend to outperform overly broad actions that may appear optimal in simplified simulations but introduce excessive SDN/NFV reconfiguration overhead in practice. Reward design is also decisive: policies that explicitly penalize SLA violations, control-plane update cost, and convergence time generally achieve better end-to-end outcomes than those maximizing attack disruption alone, especially under URLLC and MEC constraints. Finally, training realism affects generalization: models evaluated under diverse and non-stationary workloads (mobility, bursty traffic, slice elasticity) are more likely to remain effective after deployment than those trained on static traffic patterns. Overall, the best-performing RL-based MTD designs are those that balance security gain with bounded operational overhead and policy stability.

B. Recent studies and industrial guidance (2023 2025)

To reflect the most recent developments in 5G security and deployment practice, we complement the literature review with recent (2023 2025) academic studies and industrial guidance. Table XII provides direct links to key resources spanning standardization, operator-oriented security controls, and recent peer-reviewed research on SDN/NFV, network slicing, MEC/edge, and learning-assisted defenses. These additions align the survey with operator realities, where security controls must be enforceable in cloud-native infrastructures and multi-tenant slice settings while preserving strict SLA/KPI requirements.

From an industrial perspective, recent guidance emphasizes actionable security baselines for 5G deployments, including control-plane and virtualization hardening, slice isolation, auditing, and continuous risk management. This viewpoint

is essential for MTD because real deployments constrain how frequently and how broadly the system can be mutated without triggering orchestration churn, policy instability, or performance regressions. From an academic perspective, recent studies increasingly focus on slice-aware and edge-aware security and on learning-based control to optimize the security QoS overhead trade-off; however, they also highlight ongoing gaps in reproducible datasets, realistic 5G emulation/testbeds, and consistent reporting of control-plane overhead and per-slice KPI compliance.

1) Organization of the literature review and industrial anchors: To improve readability and synthesis, the remainder of this Literature Review is organized into three themes. First, we summarize *MTD approach families* according to what is mutated (e.g., identity/addressing, routing/forwarding state, and virtualization/service placement) and how the mutation is executed. Second, we review *5G threat surfaces and targeted attack types* that motivate these mutations, emphasizing how attacks manifest across the RAN, core, slicing, and MEC/edge domains. Third, we discuss *implementation contexts* in softwarized 5G (SDN/NFV and cloud-native slicing), highlighting practical enforcement points and operational constraints that shape real deployments. To ground the discussion in operator reality, Table XIII summarizes a set of industrial/standardization references that inform the feasibility boundaries of MTD (where mutation is implementable, what interfaces exist, and what operational constraints must be respected).

2) MTD approach families: what is mutated and how: MTD studies can be grouped by the primary mutation mechanism: *identity/address mutation* (e.g., IP/port hopping and endpoint obfuscation), *routing/forwarding mutation* (e.g., path perturbation and flow-rule diversification), and *virtualization/service mutation* (e.g., VNF/VM relocation, replica rotation, and dynamic placement). These families differ in both disruption strength and operational cost. Lightweight shuffling (identity or routing-level) is often easier to deploy and can be executed at higher frequency, but it may offer limited disruption against persistent adversaries who can re-identify targets via higher-layer fingerprints or by exploiting stable interfaces. In contrast, virtualization-layer mutations can change the *true* attack surface more substantially (e.g., relocating a vulnerable function or rotating replicas), but they typically introduce orchestration overhead, state transfer costs, and a higher risk of transient service impact. In 5G, the practical choice among these families depends on domain constraints: aggressive shuffling may conflict with strict URLLC budgets, while heavy relocation may be infeasible at the edge due to limited resources. *Synthesis:* the literature suggests that deployable 5G MTD often combines lightweight mutations for frequent disruption with heavier mutations triggered selectively under elevated risk, bounded by the operational constraints highlighted in Table XIII.

a) Summary and synthesis.: Taken together, the reviewed MTD families span a clear disruption cost spectrum. Identity and routing/forwarding mutations (e.g., hopping and path perturbation) are generally easier to deploy and can be executed more frequently, but their effectiveness may degrade against persistent adversaries who can re-learn the surface or exploit stable higher-layer identifiers. In contrast, virtualization/service mutations (e.g., relocation, replica rotation) can alter the *true* attack surface more substantially and may offer stronger disruption, yet they introduce orchestration overhead, state-transfer costs, and higher risk of transient QoS impact. Therefore, for 5G practice, the dominant takeaway is that deployable MTD should be evaluated as a security QoS overhead trade-off and is often best realized as a hybrid strategy: frequent lightweight mutations complemented by selective heavy-weight actions under elevated risk, constrained by per-slice SLA/KPIs.

3) 5G threat surfaces and targeted attack types: what is mitigated: The distributed and softwarized nature of 5G expands the threat surface across multiple domains and interfaces, enabling reconnaissance and scanning of exposed services, DoS/DDoS against open control or service interfaces, exploitation of virtualized network functions, control-plane abuse and policy manipulation, false-data injection against monitoring/management loops, and cross-slice or multi-tenant abuse where shared infrastructure becomes an attack amplifier. These attack types are not uniformly expressed across domains: for example, MEC/edge deployments are more exposed to localized probing and resource exhaustion, while core and SBA-facing services are attractive targets for API-level abuse and lateral movement. Consequently, the effectiveness of any mutation depends on whether it disrupts the attacker's *observable* surface (what the attacker can reliably learn) and whether the induced overhead remains acceptable for per-slice SLA/KPIs. *Synthesis:* the literature indicates that a 5G-aware MTD analysis must explicitly connect the targeted attack class to the domain where mutation is enforced and to the cost envelope that operators can tolerate, consistent with the industrial baselines summarized in Table XIII.

a) Summary and synthesis.: The surveyed literature consistently indicates that the effectiveness of MTD is highly domain- and threat-dependent in 5G. Reconnaissance disruption, DDoS resilience, and mitigation of control/management abuse require different mutation targets and enforcement points, and an approach that is effective in one domain (e.g., core or SDN-controlled fabric) may be infeasible or too costly in another (e.g., MEC/edge with tight latency budgets). A key synthesis is that meaningful comparison across proposals must explicitly connect 1) the targeted attack class, 2) the 5G domain where the attack manifests (RAN, core, slicing, MEC/edge), and 3) the acceptable operational envelope, especially for URLLC where transient loss, latency spikes, and instability can violate SLA. This motivates the taxonomy and mapping adopted in the subsequent sections.

4) Implementation contexts in softwarized 5G: where mutations are enforced: A defining feature of 5G security engineering is that many defenses are implemented through SDN/NFV and cloud-native management stacks, including SDN controllers (policy and forwarding control), NFV orchestration/MANO (lifecycle management, placement, scaling), and slice management (per-slice policy enforcement and KPI monitoring). This implementation reality directly shapes which MTD actions are feasible and how they can be executed:

TABLE XII
RECENT (2023–2025) 5G SECURITY AND MTD-RELATED RESOURCES (INDUSTRIAL GUIDANCE AND ACADEMIC STUDIES) WITH DIRECT LINKS.

Year	Type	Focus (industrial relevance / research theme)	Link
2023	Standard	5G security architecture and procedures (baseline security mechanisms; deployment anchor).	3GPP TS 33.501 (web)
2023	Standard	5G security architecture and procedures (downloadable PDF mirror).	ETSI TS 133 501 v17.11.01 (PDF)
2023	Guidance	Operator-oriented security controls for 5G (controls matrix; auditing/hardening).	ENISA 5G Controls Matrix
2023	Policy/Report	EU 5G Toolbox implementation progress (deployment/security posture; supply-chain view).	EU 5G Toolbox progress report (PDF)
2024	Guidance	Applying 5G cybersecurity and privacy capabilities (operator-facing guidance entry point).	NIST: Applying 5G capabilities (news)
2024	Guidance	NIST CSWP 36 (programmatic guidance series entry; private/enterprise 5G context).	NIST CSWP 36 (DOI)
2024	Survey	Network slicing security taxonomy: attacks, challenges, solutions, research directions.	IEEE COMST slicing-security survey (DOI)
2024	Analysis	Critical security analysis of SDN/NFV and slicing (practical deployment issues).	Springer: SDN/NFV+slicing analysis
2024	Survey	ML-enabled slicing lifecycle (automation/orchestration/security implications).	IEEE TNSM ML-enabled slicing (DOI)
2024	Article	DRL-enhanced MTD for network slicing security (empirical comparison; decision optimization).	CAMAD 2024 DRL+MTD (DOI)
2024	Article	DRL-based MTD for secure slicing in 5G and beyond (slice-centric MTD design).	WiMob 2024 DRL+MTD (DOI)
2023	Article	MTD + DRL for NFV/SDN environments (mutation policies under virtualized infrastructure).	NFV-SDN 2023 MTD+DRL (DOI)
2023	Preprint	O-RAN slicing and learning-assisted defense (RAN-side industrially relevant architecture).	O-RAN secured slicing (arXiv)
2024	Article	Practical security concerns and opportunities in 5G network slices.	IEEE Access: slice security (DOI)
2025	Article	DRL and MTD integrated framework for slice security (performance vs. baselines).	Springer: DRLandMTD slicing (2025)
2025	Survey	Open-access survey on slicing security challenges and attack vectors.	Open-access slicing-security survey (PMC)

TABLE XIII
INDUSTRIAL/STANDARDIZATION ANCHORS USED TO INTERPRET DEPLOYABILITY CONSTRAINTS IN 5G SECURITY AND MTD.

Theme in this review	5G domain emphasis	Industrial / standard references (clickable)	How used in this survey (practical lens)
Security baselines and 5G security architecture	Core, SBA interfaces, roaming, identity, key procedures	3GPP TS 33.501 (web) ETSI TS 133 501 v16.03.00 (PDF) 3GPP 33-series index	Defines security procedures and architectural touchpoints; used as an "implementation reality check" for where mutations/controls can be anchored without violating 5G procedures.
Operational event handling and monitoring expectations	Core, security operations / telemetry, incident workflows	3GPP TS 33.502 (security procedures)	Used to motivate the need for continuous telemetry and event-driven defense loops that can trigger/adapt mutation (especially for adaptive MTD policies).
EU operator-oriented security controls and compliance view	Multi-domain (RAN/Core/Slicing/MEC), supply-chain and controls	ENISA 5G Controls Matrix (page) ENISA 5G Controls Matrix (paper PDF) EU 5G Toolbox progress report (PDF)	Provides a control-centric, audit-friendly view; used to interpret which MTD mechanisms align with operator controls (hardening, isolation, monitoring) and where frequent mutation may conflict with stability/compliance requirements.
NIST guidance on applying 5G cybersecurity/privacy capabilities	Private/enterprise 5G, platform integrity, security capabilities	NIST CSWP 36 (project page) NIST CSWP 36 (IPD PDF) NIST CSWP 36E (IPD PDF)	Used as a deployment-oriented reference for capability-based security building blocks (trust, integrity, monitoring) that MTD can leverage while respecting operational constraints in enterprise/private 5G.
NFV security management and policy/monitoring considerations	NFV/MANO security controls, monitoring, security policy management	ETSI NFV-SEC 013 (PDF)	Used to frame feasibility constraints for virtualization-layer MTD (migration/relocation/replication) and the associated monitoring/policy lifecycle management realities.

routing/forwarding mutations can be realized via controller-driven rule updates, while virtualization-layer mutations require orchestrator coordination and may involve stateful cutovers. In practice, feasibility is bounded by operational constraints such as control-plane churn, convergence delay, rule-space/TCAM pressure, VNF migration time and state transfer cost, and multi-domain coordination complexity. These factors explain why some proposals remain evaluation-only while others move closer to deployable workflows. *Synthesis:* a deployment-realistic MTD survey should therefore assess not only what is mutated, but also *which plane enforces it* and *what overhead/stability envelope* it implies precisely the role of the industrial anchors summarized in Table XIII.

a) Summary and synthesis.: Across implementation-focused works, deployability is determined not only by the mutation concept but by the enforcement plane and its operational limits. SDN-driven mutations are bounded by rule-space/TCAM pressure, controller load, update bursts, and convergence behavior, while NFV/MEC-oriented mutations are bounded by orchestration latency, service-chain dependencies, state transfer, and cutover disruption. As a synthesis, evaluations that report only security improvement are insufficient for 5G; they must also report churn, convergence delay, and per-slice KPI impact to establish feasibility in production-grade networks. These observations directly motivate the benchmarking and reproducibility gaps discussed later and explain why constraint-aware (including RL-based) decision-making is emphasized in the survey.

C. Threat model summary

To improve clarity and make the threat model easier to scan, we summarize the assumed attacker goals, capabilities, knowledge, positioning, defender assumptions, and scope limitations in a single narrative. In our baseline setting, the attacker aims to compromise confidentiality, integrity, and/or availability of 5G services by progressing through the typ-

ical kill-chain stages of 1) reconnaissance (e.g., scanning and probing exposed IP/port/service surfaces and enumerating reachable VNFs/VMs/containers), 2) exploitation of software weaknesses or misconfigurations, 3) privilege escalation and lateral movement across virtualized components, and 4) impact actions such as service disruption (DoS/DDoS), route manipulation, or false-data injection, depending on the targeted 5G domain (RAN-edge, core functions, slices, or critical servers). We assume the attacker can generate both volumetric and application-layer DoS traffic (including botnet-driven traffic in the external case), attempt credential abuse (e.g., guessing, replay, or theft via compromised endpoints), and adapt tactics based on observed network changes and measured service behavior.

Regarding attacker knowledge and visibility, we assume the adversary may have partial knowledge of common 5G/SDN/NFV architectures (e.g., typical control and data-plane roles and well-known service functions), but does not know future MTD mutations in advance; in particular, the attacker cannot predict the next target set, remapping, or mutation interval when randomized or adaptive policies are used. Consequently, information learned during reconnaissance becomes stale after mutation and the attacker must repeatedly re-discover effective targets, which increases time-to-exploit and operational cost. We consider two attacker positions: 1) an external attacker acting through Internet-facing surfaces and 2) an internal attacker originating from a compromised endpoint or tenant slice, which enables local probing and potential lateral movement.

On the defender side, we assume the SDN/NFV control and orchestration plane is trusted and remains uncompromised in the baseline model, and that MTD components (e.g., target selector, interval calculator, and rule generator) are protected and have sufficient authority to enforce configuration changes across the relevant domains. We further assume that monitoring and telemetry are available (at least at an aggregated level) to estimate attack indicators, operational overhead, and QoS/SLA impact, enabling risk-aware or learning-assisted mutation decisions. Finally, we clarify scope limitations: physical-layer threats such as jamming and hardware sabotage are outside the main scope unless explicitly stated, and complete compromise of the SDN controller or orchestration plane is treated separately as a worst-case scenario rather than the baseline assumption.

III. Evaluation Methodology and Metrics

We evaluate MTD along two axes: *security effectiveness* and *operational cost*. Effectiveness is captured by attack success rate reduction, time-to-reconnaissance inflation, and service availability under stress. Cost is captured by flow-rule churn, migration overhead (CPU/memory/network), cutover loss (packet drops/latency spikes), and control-plane load. Additional tenant-centric SLO metrics include p95/p99 latency, throughput, error rate, and time-to-recovery after a mutation. For learning-based methods, we report sample efficiency, convergence time, policy stability (oscillation), and safety violations under guardrails. Unless stated otherwise, experiments are repeated across random seeds and traffic mixes, and confidence intervals are reported.

IV. MTD Targets

First, we examine the 5G network architecture to investigate the possible targets.

As depicted in the architecture shown in Figure 3, a 5G network consists of five main components: the core network, the radio access network (RAN), the critical servers, the end devices, and the slices. It is worth noting that the sample 5G network shown in Figure 3 is implemented in a software-defined environment. Hence, in addition to the five components mentioned, it also contains OpenFlow switches and the SDN controller. Each of these components can be considered as the MTD target. The core network includes the infrastructure resources used to serve the user's custom requests. The slices are the virtual networks generated based on the user's requests and mapped on the core network nodes. Users join the network and utilize the services provided by end devices, which are connected to the main components via the RAN. There are also some critical servers in a 5G network, such as the network slice manager (NSM), that receive the user's requests, control the mapping processes, and generate appropriate responses. To improve readability and avoid visual clutter, we present the SDN/NFV-enabled 5G architecture in two complementary forms. Fig. 3 provides a complete end-to-end overview of the architecture, while Fig. 4 additionally offers a decomposed representation as four panels (a d) to expose fine-grained details that are difficult to read in a single dense diagram. Specifically, Fig. 4(a) highlights the RAN access segment, Fig. 4(b) summarizes the 5G core-network view, Fig. 4(c) illustrates the network slicing view, and Fig. 4(d) depicts the SDN/NFV control view (e.g., OpenFlow/gRPC interactions). This split enables consistent font sizing and spacing and allows readers to inspect labels, links, and functional roles more clearly without losing the global context captured in Fig. 3.

Having investigated the main components of 5G networks, we now discuss the potential targets, which are summarized in Table XIV,

The core network is not a special infrastructure that only exists in 5G networks. Similarly, any other computer networks, and hence, the MAC address of the physical machines, are possible targets to be mutated. Many researchers, such as Chowdhary et al. [58], have considered the IP address as the moving target. In addition to the address of this machine, their open ports for communicating can also be mutated. Mutating the port numbers is proposed by a few researchers, such as Zhou et al. [63]. The network topology is another possible target for being changed in the core network. The way the physical machines are connected together can be changed as a defensive mechanism, such as the one suggested by Steinberger et al. [59].

For securing the critical servers, the both IP address and port number are possible targets. For example, the NSM server can periodically change its IP address or the listening port to receive requests. Doing so, whenever flooded traffic is sent

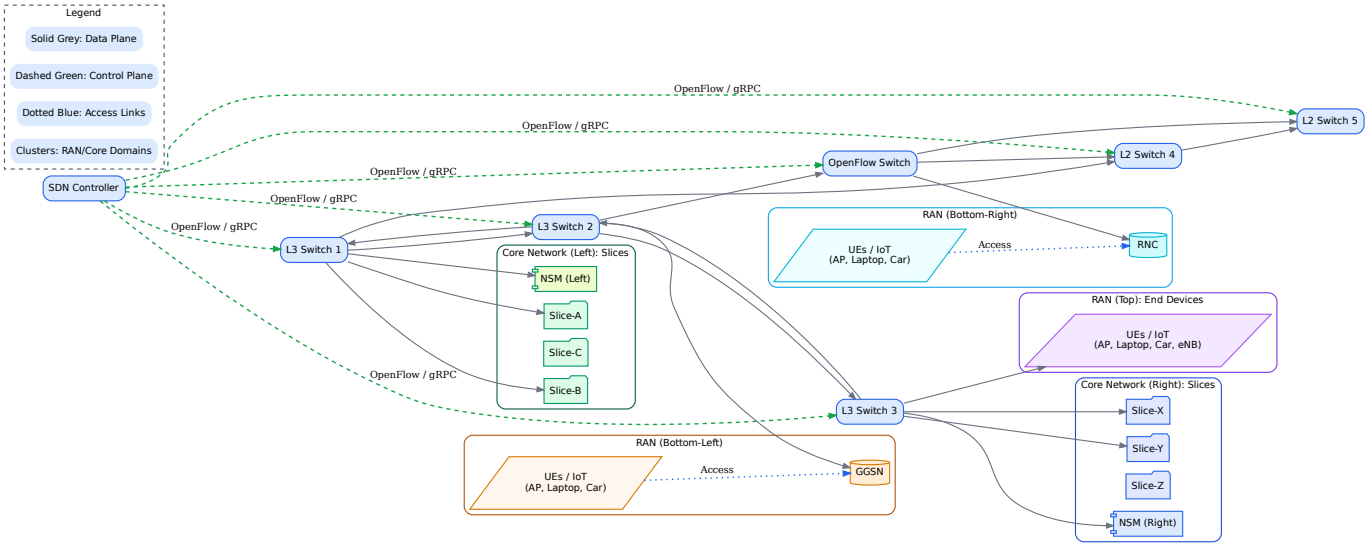


Fig. 3. The architecture of 5G networks implemented in a software-define environment

TABLE XIV
THE TAXONOMY OF MTD TARGETS (✓ SUPPORTED/USED; ✗ NOT SUPPORTED OR NOT REPORTED).

Category	References	Core network	RAN	Critical servers	End devices	Slices
IP address	[55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 20, 54]	✓		✓	✓	✓
Port number	[56, 65, 66, 63, 54]	✓	✓	✓		✓
Network topology	[59, 47]	✓				
MAC address	[55, 48]	✓			✓	
Function/VM migration	[55, 59, 62, 54]	✓	✓			✓
Route mutation	[44, 24]		✓			✓
Service endpoint randomization	[18, 53]			✓		
Hosting infrastructure	Future direction					✓

Note: Blank cells indicate the information was not reported in the referenced work.

Comparative discussion (Table XIV). Table XIV highlights that MTD targets differ substantially in security impact and operational cost. *Address/port hopping* is commonly adopted because it directly disrupts scanning and reconnaissance with relatively low implementation complexity, but its protection is typically limited when attackers can quickly re-discover endpoints or when higher-layer identifiers remain stable. *Topology/routing mutations* can increase attacker uncertainty more broadly, yet they must be carefully bounded in 5G to avoid convergence delays and control-plane churn that may degrade QoS. *Host/VNF relocation* and *service placement mutations* can provide stronger disruption against targeted exploitation by changing the actual attack surface, but they are constrained by MEC resource availability, migration time, and stateful function dependencies. In 5G slicing, target selection is further constrained by per-slice SLA requirements; therefore, practical designs often prioritize targets that 1) offer high disruption to reconnaissance and lateral movement, 2) can be executed with predictable overhead, and 3) minimize cross-slice side effects. This comparison clarifies why some targets are preferred in deployment-realistic 5G settings even when other targets may show higher security gains in simplified experiments.

toward the critical servers since they are mutated, the adversary has to launch another flooding attack to reach the servers.

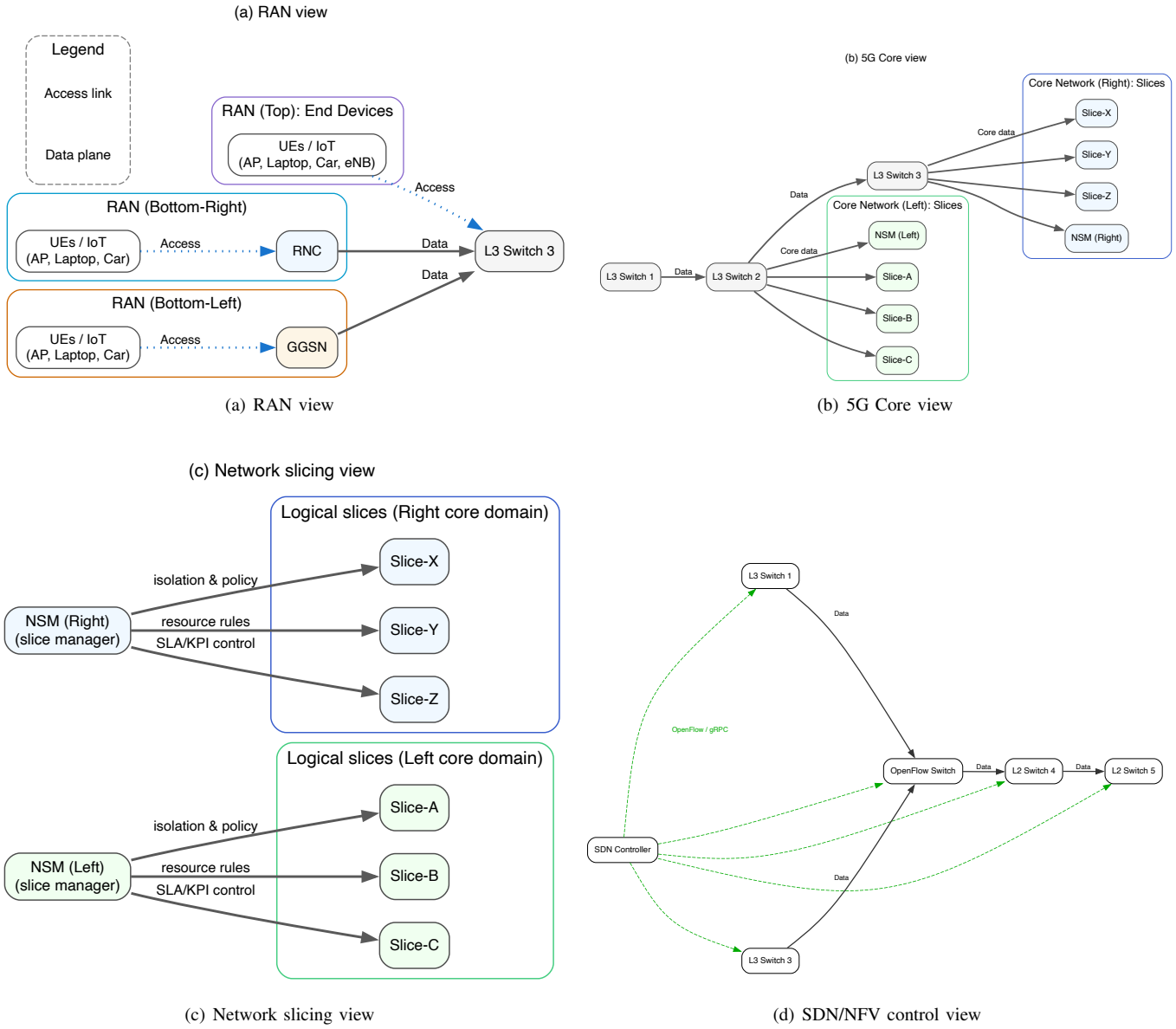
The only target possible for the end devices is their IP address in the network. Because the end devices are compromised based on their addresses, and no one uses their port. It is still possible to launch an attack against the common open ports, but mutating them requires large-scale configurations, as the ports are on lower networking layer than IP addresses. Using SDN, it is not time-consuming to change the IP addresses of the packets. But the port numbers are not as easy.

The slices can be mutated based on their IP address and open ports, similar to the critical servers. There is also a future direction here. Each virtual node of a slice is mapped on a physical machine in the core network based on some resource constraints. To mutate them, one can change the infrastructure node of that slice. However, this change is not similar to virtual machine migration because there are some constraints based on the other virtual nodes of that slice.

V. MTD PARAMETERS

A significant challenge in MTD approaches is determining how to balance the trade-off between the MTD cost and the security level it provides. It is obvious that when we move many targets, we can prevent many attacks and their side effect. But, on the other hand, the complexity of these movements, such as the extra delay generated, may become too high, which is unsuitable for a network. Hence, an ideal MTD approach moves as low as possible targets with a major impact on network security.

We give an example to show why selecting the optimal set of targets is important. In Figure 5, there are four virtual machines (VMs) with the same vulnerabilities that are located



(a) RAN view (b) 5G Core view

(c) Network slicing view (d) SDN/NFV control view

Fig. 4. Decomposed views of the SDN/NFV-enabled 5G architecture (Fig. 3) to improve readability. Panel (a) highlights the RAN-side access segment and UE/IoT connectivity, (b) summarizes the core-network functions and inter-domain forwarding, (c) illustrates logical network slicing and per-slice separation/management, and (d) shows SDN/NFV control interactions (e.g., OpenFlow/gRPC) with switching elements. The panels are provided as vector PDFs to remain sharp under zoom and to make labels and connectors easier to read.

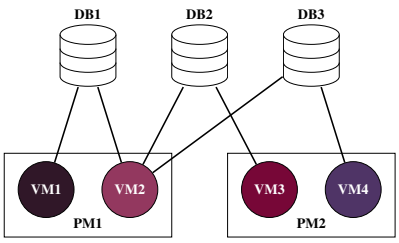


Fig. 5. A sample scenario to show the importance of selecting the optimal set of targets.

on two physical machines (PMs). Each of the VMs has a stricted access to the three databases (DBs). While VM2 has access to all DBs, VM1, VM3, and VM4 have access to only one. Now assume that an adversary tries to access all of the DBs. Since the adversary's resources are limited, he wisely puts its effort on compromising VM2. Because once it is compromised, the adversary gains access to all the DBs. Similarly, migrating VM2 to another physical machine, as an MTD solution, is more efficient than migrating VM1, VM2, and VM3. These two migrations have the same impact on securing the DBs, while the latter requires fewer changes in the configurations. When a single VM is migrated, it must connect to a DHCP server to obtain the appropriate IP address, which generates an extra delay. Moreover, the traffic packet toward its previous IP address will drop, because of the change in the address. These are just two of the cost-related events that may occur. Accordingly, it is suggested that the optimal set is always selected to reduce the MTD cost.

Different parameters are currently considered for selecting the optimal set of targets to be moved. Soussi et al. [22] have considered the importance and risk level of the slices as the moving parameter in 5G.

Now, we review some of the techniques for selecting the optimal set of targets.

Chowdhary et al. [55] have proposed a scalable MTD framework for SDN-based cloud networks. This framework consists of three main components. The first component gathers the information about the vulnerabilities and analyzes them. An attack graph is constructed using this information and a low complexity algorithm is proposed to analyze it. This algorithm finds minimum k-cut of AG to partition it and process each partition with a different processor to distribute this process and reduce its time consumption. This partioning helps the solution be scalable and have an acceptable performance for large-scale networks. The second component selects the appropriate countermeasure by choosing the most vulnerable VM and migrate it to another physical machine. This migration is performed in a way that reduces the total vulnerability score of the system. The third component checks the conflicts in flow rules that are set on the switches. The rules are consistent with both the network configuration and the security policies. The total complexity order of this solution is $O((N/p)^2)$, where N is the number of nodes and p is the number of processors.

Shi et al. [56] have proposed an MTD mechanism in SDN environment, the main mutation parameter of which is the hosts risk levels. A vulnerability degree, say d, is assigned to each of the hosts based on Common Vulnerability Scoring System (CVSS). Then, a random number is generated, and if it is less than d, that host must be mutated. In other words, the probability of a host being mutated is directly related to its vulnerability level. As a result, the complexity of this mechanism is $O(N)$, where N is the number of hosts. Moreover, this mechanism has considered both IP addresses and open ports to be mutated. The former is for mitigating man-in-the-middle attacks, and the latter is for mitigating port scanning attacks, respectively. Chowdhary et al. [65] have also utilized port hopping technique, based on the CVSS score of the virtual machines in a cloud environment.

Steinberger et al. [59] have proposed an MTD defensive strategy for mitigating DDoS attacks against Internet service provider networks. The main focus of this work is to support the MTD strategy on large-scale networks considering their requirements, such as ease of deployment and scalability. While it is said that the way of selecting hosts to mutate or optimizing the mutation time interval is crucial, the implemented strategy used an interval of 20 minutes for mutation, due to the average DDoS attack interval of 60 minutes. The changing targets in this work are the hosts IP addresses and the network topology. For changing the network topology is different situations, the routers are utilized.

The main focus of the MTD mechanism proposed by Liu et al. [66] is on the synchronization problem. In this mechanism, the port hopping scheme is implemented in an SDN environment using a time synchronization method to maintain consistency in the hopping policy. The clients who want to be served, first send a request to the SDN controller, and the controller will authenticated them. After that, the controller sends the calibration information to the clients. All of these procedures are performed within an encrypted channel. For evaluating these mechanisms, a time interval of 10 seconds is chosen.

Macwan and Lung [60] have presented an SDN environment that deploys an MTD approach. In this work, the details of OpenFlow rules for deploying the MTD mechanism are mentioned. This method is based on mutating the IP addresses. The authors have assigned a virtual IP address to each of the hosts to mutate them Narantuya et al. [61] have also investigated the deployment of MTD in SDN with multiple controllers. The shuffling intervals are different under different situtaions. For the case of one, two, and three controllers, it is 12, 8, and 6 seconds.

Luo et al. [62] have proposed an SDN-based MTD approach, that shuffles the IP address of random hosts during random shuffling intervals. In this work, honeypots are also deployed to detect and mitigate the attacks. The main focus of Debroy et al. [67] is to minimize the shuffling interval, and also to select the best VM for migration.

Hyder and Ismail [68] have shuffled both control plane and data plane in SDN. The controllers are shuffled in this work to prevent the reconnaissance attacks against them. In the data plane, it is assumed that the requests are first forwarded to the load balancers and then the webservers. The IP address of the load balancers and the port number of the web servers are shuffled every 120 and 60 seconds, respectively.

VI. MTD Intervals

In some MTD approaches, there is not a specific action that triggers the movements. For example, an MTD approach may consider that the selected target have to be mutated every 5 seconds to avoid any possible attacks with a period of 5 seconds affecting the network performance. The 5G network MTD solution suggested by Soussi et al. [22] consider this kind of approaches. The first point that must be considered for this type of MTD approaches is the process of forcing all the hosts particpating in the network have scanning periods higher than the movement interval length to guarantee that a maclicious user's information in the i^{th} interval cannot be used for launching an attack in that interval. Among these techniques, some use random intervals, some of the others use fixed-length intervals, and the others select the interval length from a specific set of numbers. Steinberger et al. [59] have proposed an MTD strategy for Internet service providers, the mutation length of which is 20 minutes. In other words, the IP address of the host in the network are changed every 20 minutes. In the MTD mechanism proposed by Liu et al. [66], port hopping is carried out every 10 seconds. Narantuya et al. [61] have proposed an MTD solution in software-defined networks, where the interval length varies based on the number of controllers in the network. In the case of having one, two, and three controllers, the IP address mutation is 12, 8, and 6 seconds, respectively. Luo et al. [62] have suggested completely random mutation intervals.

There are also some other MTD approaches, such as the one proposed by Shi et al. [56], that are assisted by another

security tool and the aided information specifies the movement intervals. One of the ways of doing so is to use a threat detection system, that alerts the MTD component about the malicious event occuring in the network. In this situation, the MTD component starts its intervals or makes them faster. Christopoulou et al. [69] have suggested an MTD framework for 5G networks that utilizes an anomaly detection component.

The different types of approaches for slecifying the mutation intervals are summarized in Table XV.

VII. MTD Implementations

Software-define networking plays the key role in implementing the MTD approaches in 5G networks. These networks are large-scale and have numerous complex components, making it nearly impossible to control and monitor them using traditional methods. The controller in an SDN can easily support the challenging tasks of traditional networks. As a result, this section focuses on implementing an MTD approach in SDN. The suggested components that may be considered for this implementation are described in Table XVI.

Table XVI.

Due to the non-stop changes caused by an MTD approach, the first concern is the forwarding rules of the OpenFlow switches. The rules must be up-to-date to properly forward the packets based on the changes, with the lowest possible number of packet losses. We can consider a rule generator component for this task.

The rule generator has to know the mutation intervals. Hence, we consider another component, interval calculator, to generate the time intervals. This component communicates with the rule generator, and announces the beginning of each mutation interval. In the cases that the MTD approach is aided by another security tool, interval calculator must also communicate with that tool.

The target selector is another component for MTD implementation. This component utilizes custom algorithms and techniques, including machine learning and optimization algorithms, to select the optimal set of targets for mutation. Target selector requires collecting the information of the network, or the part that must be mutated. Hence, a shared database with the controller is considered that stores the network statistics or other reports received by the controller. Using this information, the target selector finds the optimal set and then passes it to the rule generator. Then, the rule generator can define the appropriate rules based on the optimal set of targets.

VIII. Standardization and Integration

Practical MTD requires well-defined interfaces to management and orchestration stacks. Policies translate to orchestrator intents (e.g., VNF migration, scaling) and SDN flow updates via controller north/southbound APIs. Clear contracts for telemetry schemas, policy conflict resolution, and admission control ensure that security actions remain compliant with tenant SLAs and operational change windows.

IX. Mitigated Attacks

Different types of attacks can be effectively mitigated by MTD approaches (Table XVII). The most common and dangerous one is DoS/DDoS attacks. In a DoS attack, the adversary floods the network toward a specific target to make it unavailable and out of access. In a DDoS attack, the source of this flooded traffic is a massive number of compromised hosts. The distributed nature of DDoS attacks, makes them harder to detect. When the targets of a DDoS attack are mutated, the impact of such attacks becomes smaller.

Scanning attacks are another threat that MTD can mitigate. These attacks are the initators of many other attacks such as DDoS. The advesary scans the network and recognizes different features of the network, such as vulnerabilities and the security breaches he could abuse. When an MTD approach mutates the targets, the previously collected data by the adversary becomes invalid.

X. Datasets, Testbeds, and Reproducibility

Experiments rely on emulated SDN fabrics and containerized NFV chains with controllable background traffic and attack generators. Traffic mixes include benign web/DB flows and adversarial scans/DDoS at varying rates. Each experiment publishes: 1) configuration snapshots (controller, orchestrator, switch tables), 2) mutation timelines, 3) telemetry schemas, and 4) analysis notebooks to recompute all figures/tables from raw logs. This packaging enables like-for-like comparison across MTD strategies and fosters reproducibility.

XI. Discussion

A central finding of this survey is that the literature becomes much more coherent when it is interpreted through one lens: 5G-oriented MTD as a closed-loop, deployment-constrained security-control problem. Under this lens, theoretical models define the mutation logic, application-specific works reveal domain constraints, and RL-based studies contribute adaptive actuation policies. These strands should therefore be read as complementary layers of one design space rather than as competing narratives. The concentric classification presented in Fig. 6 provides an intuitive yet analytically rigorous mapping of the MTD literature. The innermost core, representing *Survey/Review* papers, consolidates foundational taxonomies and meta-analyses that have structured the domain's conceptual landscape. These works act as navigational anchors, guiding researchers toward both established paradigms and underexplored areas.

The *Theoretical/Framework* layer captures general-purpose MTD models and architectures, often mathematically formulated to ensure scalability, robustness, and reduced attack surfaces. These studies tend to be domain-agnostic, enabling adaptation across diverse network infrastructures. In contrast, the *Application-specific* layer contextualizes MTD concepts within targeted domains such as IoT, vehicular networks, and cloud computing addressing unique operational constraints and threat models. The outermost *RL-based* ring reflects the

TABLE XV
TAXONOMY OF MTD MUTATION INTERVALS WITH POLICIES, SELECTION, SYNCHRONIZATION, EXAMPLES, AND DEPLOYMENT NOTES

Category	Subcategory	Interval policy / trigger	Selector / algorithm	Sync / keying	Representative values & references	Overhead & QoS notes	Typical targets / scope
solo	fixed-length	Constant T independent of state/load	Operator policy; controller timer; static config	Time-based (NTP/PTP); controller push before epoch change	[59]: T=20 min (IP shuffling); [66]: T=10 s (port hopping); [68]: LB IP 120 s, web port 60 s	Low control-plane cost; predictable (learnable) by attacker if monitored; choose T to bound exposure/churn	Host/service IPs, ports, VIPs, controller roles
	set of intervals	$T \in \{T_1, \ldots, T_k\}$ chosen by policy or learned over discrete set	Rule-based (e.g., depending on #controllers); DRL over discrete actions	Pre-shared set; epoch advertised by controller; tolerant windowing	[61]: $T \in \{12, 8, 6\}$ s (by #controllers); : RL chooses among preset T values	Balances predictability/agility; simpler admission control; discrete choice eases sync	IP/port shuffling in SDN, slice-facing gateways
	random	Sample T from distribution (e.g., uniform on $[T_{\min}, T_{\max}]$)	PRNG seeded per host/slice; enforce min dwell time	Token/epoch IDs with validity window; controller issues updates on change	[62]: random periods and random host selection	Maximizes unpredictability; care for jitter, re-auth churn; tune $T_{\min}/T_{\max}$ to SLA	End-host IPs, decoy/honeypot rotations, ephemeral routes
accompanied	threat-driven	Intervals start/speed up upon alert; back-off after quiet period	Anomaly/IDS score thresholds; hysteresis/back-off controller logic	Event-driven controller push; TTL-coded epochs in handshake	[69]: anomaly-triggered mutations; [56]: CVSS-informed triggering	Low overhead in benign periods; FP spikes cause churn, FN raises exposure; good for URLLC with cautious ramps	LB IPs, service ports, slice endpoints, controller role shuffle
	learning-optimized	T adapted online to cost/benefit (security vs. QoS)	(Deep) RL/meta-RL chooses T (continuous/discrete) or mutation cadence	Same as above; policy/epoch broadcast before new cadence	[25]: RL selects shuffling period; [36]: DRL for optimal shuffling period; [27]: meta-RL (implicit interval via migration cadence); [54]: DRL with port-hopping cadence	Tracks attacks/load; may add inference latency; needs safe exploration and SLA guards	IP/port shuffling, VNF/VM migration cadence, service replicas

Notes: When a paper omits numeric T, the policy/trigger is summarized. $T_{\min}/T_{\max}$ bound churn and exposure. LB = load balancer; PRNG = pseudo-random number generator; URLLC = ultra-reliable low-latency communications; FP/FN = false positive/negative.

Comparative discussion (Table XV). Table XV shows that the mutation interval is a key lever controlling the security overhead trade-off. *Fixed-period* policies are simple and predictable, but they can be learned and timed by adaptive adversaries, and they may trigger unnecessary churn during high-load periods. *Randomized* intervals increase attacker uncertainty and reduce predictability, yet they can still violate QoS if mutations occur during latency-sensitive service windows. *Adaptive* intervals (e.g., triggered by risk level, anomaly signals, or RL policies) tend to perform better in 5G settings because they align mutations with context: they can increase mutation frequency when threat likelihood is high while reducing mutations under URLLC/MEC constraints to protect SLA compliance. However, adaptive designs must explicitly address synchronization and stability (e.g., avoiding oscillatory behavior and ensuring configuration convergence). Overall, practical 5G MTD favors interval policies that are threat-aware, bounded by QoS constraints, and engineered to limit control-plane update rates and convergence overhead.

growing adoption of reinforcement learning to enable dynamic, context-aware MTD policies that can adapt to evolving adversarial behaviors in real time.

This layered representation allows us to discern three important insights. First, there is a temporal trend showing that while theoretical contributions dominate earlier works, RL-based methods have gained significant traction post-2019. This aligns with broader advances in deep learning and computational optimization. Second, there is evident cross-pollination between categories, with theoretical models frequently informing RL-based implementations, and application-specific deployments often leveraging generalized frameworks. Third, the visual and categorical synthesis enables researchers to identify gaps, such as the limited number of RL-based solutions tailored for multi-domain, heterogeneous environments, highlighting promising directions for future investigations.

By juxtaposing Fig. 6 with the attack-specific mapping in Table XVIII, we create a dual analytical perspective: the figure emphasizes methodological evolution and inter-domain relationships, while the table provides a direct mapping between MTD approaches and the classes of cyberattacks they mitigate. Together, they provide a comprehensive lens through which the maturity, adaptability, and applicability of MTD strategies can be evaluated in both academic and operational contexts.

A. Design Guidelines and Best Practices

Adopt hybrid cadences (background and event-driven), prioritize minimum effective sets for mutation, stage updates with versioned rules, and enforce guardrails around learned policies. Track a small set of universal KPIs (security gain, churn, cutover loss, time-to-recovery) and wire automatic rollback when SLOs regress. Treat the MTD control plane as a protected asset with authentication, rate-limits, and auditability.

1) Why reinforcement learning is emphasized for decision-making in 5G MTD: A central reason for emphasizing reinforcement learning (RL) in this survey is that MTD in 5G is inherently a *sequential* and *feedback-driven* control problem rather than a one-shot configuration choice. In operational 5G, the defender repeatedly decides what to mutate (targets such as IP/port/route mappings or VNF placement), when to mutate (mutation intervals and dwell times), and how aggressively to mutate (scope and granularity), while the effects unfold over time through attacker adaptation, orchestration delays, and transient service disruptions. This makes purely static schedules or ad-hoc heuristics vulnerable to predictability and suboptimal under non-stationary conditions such as mobility, bursty traffic, slice elasticity, and MEC resource variability. RL is well matched to this setting because it learns policies that optimize long-term outcomes under uncertainty, enabling context-aware decisions that integrate both *security signals* (e.g., scan intensity, compromise likelihood) and *operational signals* (e.g., per-slice KPIs, controller load, convergence

TABLE XVI
PRACTICAL COMPONENTS FOR IMPLEMENTING MTD IN 5G

Component	Purpose	Key inputs	Core functions / algorithms	Interfaces / APIs	Latency / scale constraints	Failure modes & mitigations
Rule generator	Install correct forwarding after each mutation	Current topology, target set, epoch/timer, policy constraints	Flow synthesis; dependency ordering; two-phase updates; fast-reroute; transactional install; versioned tables (cookie); *SDN/MTD IP/route shuffling:* [55, 56, 61, 59, 57, 64]	OpenFlow (FLOW_MOD), P4Runtime, gRPC/REST to SDN (ONOS/ODL), NETCONF/YANG	Apply time <50–100 ms/hop; batch size caps; switch TCAM limits	Blackholes/loops due to partial install -to- staged rollout, atomic groups, drain-before-cutover; switch overload -to- rate-limit updates
Interval calculator	Compute mutation cadence (fixed/random/learned)	Threat score, traffic load, SLA class, learned policy state, operator policy	Timer wheel; stochastic sampling ($T \sim \mathcal{D}$); *RL/DRL cadence control:* [24, 48, 25, 21, 19, 46, 4, 16]; hysteresis/backoff; safety guards (min dwell)	Northbound REST to Policy Manager; pub/sub bus (Kafka/NATS); signal to RuleGen/Selector	Jitter <10 ms for fast shuffles; clock sync across domains	Clock skew/desync -to- NTP/PTP and grace windows; oscillation -to- hysteresis; bursty triggers -to- token-bucket gating
Target selector	Pick optimal assets to mutate under cost/risk	Asset inventory, CVSS/vuln db, attack graph, telemetry (NetFlow/sFlow), budget/SLA	Heuristics (greedy, k-cut); ILP/MILP (budgeted max-cover); *Attack-graph/optimization/DRL selection:* [57, 19, 32, 33, 31, 17, 28]; graph RL	DB/Feature store; REST to Policy Manager; gRPC to RuleGen; SIEM feeds	Online compute <200 ms (tight-loop); offline precompute allowed	Flapping/oscillation -to- damping, budget caps; stale inventory -to- periodic reconcile; myopic choices -to- look-ahead window
Threat detection module	Detect scans/DDoS/anomalies to trigger/shape MTD	Flow stats, IDS alerts (Snort/Suricata), logs, RAN counters, host EDR signals	Signature IDS; anomaly (autoencoder, IsolationForest); sketching (Count-Min); alert correlation; sliding-window scoring; *SDN flow classification / hybrid ML detectors / malware-aware MTD:* [18, 70, 71, 72, 68]	SPAN/mirror; Kafka; gRPC/REST to Interval/Selector; SIEM integration	Detection latency $\ll$ interval length; FPR bounded per SLA	FP bursts -to- dampening, quorum; FN risk -to- companion fixed cadence; poisoning -to- model canaries
Policy manager	Enforce intents, SLAs, constraints, conflict resolution	High-level intents, tenant SLAs, regulatory rules, maintenance windows	Policy parsing; conflict detection (precedence lattice); admission control; ABAC constraints; what-if checks; *Telco policy/closed-loop MTD orchestration:* [22, 40, 73, 41]	Northbound REST/GUI; southbound to Selector/RuleGen; AuthN/Z (OIDC/RBAC)	Deterministic decision; sub-100 ms path for online ops	Policy deadlocks -to- fallback precedence; unsafe intents -to- deny with explanation/log; drift -to- conformance checks
State & log database	Persist states, actions, metrics; enable learning/replay	Topology snapshots, mutation logs, KPIs, flow stats, labels	Time-series (Prometheus/Timescale); feature store; TTL/retention; replay/simulation exporters; *Visualization/telemetry for MTD ops:* [74]	SQL/TSDB; object store; CDC/streaming (Kafka); Grafana dashboards	Sustain ingest $\geq$ 100k events/s; query p95 < 200 ms for dashboards	Data lag/loss -to- WAL, replication; skewed clocks -to- write-side timestamping; bloat -to- tiered storage
Orchestration interface	Execute MTD actions on infra (migrate/scale/hop)	Action plan (migrate VNF/VM, scale replicas, port-hop), resource map	Translate to MANO/K8s/OpenStack; canary/blue-green; live migration w/ pre-copy; rollback on SLO breach; *NFV/telco-cloud slice/security orchestration:* [40, 39, 69, 42, 38]	ETSI NFV SOL003/005; OpenStack; Kubernetes/Helm; SR-IOV/CNI APIs	Downtime < SLO (e.g., sub-100 ms for URLLC slices); quota-aware	Partial migration, pod evictions -to- canary and health gates; SLO breach -to- auto-rollback; capacity shortfall -to- reschedule/deferral
Key mgmt & sync (optional)	Distribute epoch tokens/keys to legit clients for hopping	AAA/PKI, device identities, epoch schedule, revocation lists	Token derivation (HKDF); rotating keys; overlapping epochs; resync protocol; DTLS/TLS channels; *Token/TOTP-based MTD & IP hopping:* [75, 76]	AAA (RADIUS/LDAP/OAuth2), EAP, mTLS, KMS (e.g., HashiCorp)	Handshake <100 ms; tolerate brief overlap/clock skew	Client desync -to- overlap windows; key leak -to- rotate+revoke; storm -to- rate-limit, backoff
Telemetry & safety guard (optional)	Safe exploration & A/B; guardrails for RL policies	Live KPIs, shadow policy outputs, guard thresholds	Shadow-mode policy; canary cohorts; constraint wrappers; emergency freeze switch; *Surveys & frameworks for safe, adaptive MTD (policy learning/guarding):* [4, 16, 15, 74]	Feature flags service; policy bus; roll-out controller	Shadow-to-live cutover within guard bounds; abort on SLO breach	Policy regression -to- auto-freeze and rollback; drift -to- periodic re-baseline

Abbreviations: RL = Reinforcement Learning; DRL = Deep RL; ILP/MILP = (Mixed Integer) Linear Programming; ABAC = Attribute-Based Access Control; TCAM = Ternary CAM; SIEM = Security Info & Event Mgmt; MANO = Management & Orchestration; URLLC = Ultra-Reliable Low-Latency Communications; KMS = Key Mgmt Service; SLO/SLA = Service Level/Agreement; EDR = Endpoint Detection & Response.

TABLE XVII
THE ATTACKS THAT CAN BE MITIGATED USING MTD APPROACHES

Attack	References
Denial of Service / Distributed Denial of Service (DoS/DDoS)	[38, 64, 63, 77, 54, 25, 26, 24, 20, 19, 50, 49]
Scanning / Reconnaissance attack	[72, 78, 79, 76, 29, 20]
False Data Injection attack	[80, 81, 82, 48]
Route manipulation / Packet drop attacks	[44, 24, 47]
Malware / Application-layer intrusion	[71, 18, 53]
Resource exhaustion / Service degradation	[54, 27, 46]

delay).

Moreover, 5G introduces a particularly strict *security QoS overhead* trade-off. A mutation can reduce attack success probability but simultaneously increase control-plane churn, rule-update bursts, and convergence time, which may violate URLLC latency/jitter budgets or degrade slice-level KPIs. RL provides a natural mechanism to encode these trade-offs through reward and constraint design, allowing the policy to prefer low-overhead actions when the network is congested or when URLLC slices are active, and to increase mutation intensity when risk rises and slack exists. This motivation is consistent with the surveyed RL-based studies that use RL for mutation scheduling decisions Chai et al. [25], slice-aware multi-agent control Yoon et al., Chowdhary et al. [24, 23], scanning-disruption strategies Zhang et al. [20], and robustness/generalization across configurations Li and Zheng [27]. At the same time, we stress that RL is not a replacement for other paradigms: supervised/unsupervised ML is commonly used to detect attacks or estimate risk (supporting the ob-

TABLE XVIII
COMPARISON OF REVIEWED MTD-RELATED WORKS CATEGORIZED BY DOMAIN (AS IN FIG. 6)

Domain	References	Main Focus / Contribution	Technique Type	Year Range	Targeted Attacks
Survey / Review (core)	[14, 9, 74, 70, 10, 15, 11, 4, 16, 12, 13]	Comprehensive surveys of MTD in various contexts (AI-driven, IoT, power grids, game theory, etc.)	Literature review/taxonomy	2020 2025	General coverage of MTD-related threats
Theoretical / Framework (inner)	[55, 56, 65, 59, 60, 61, 66, 62, 57, 63, 67, 68, 64, 75, 73, 72, 78, 79, 76, 81, 82, 69, 77]	Proposed generic architectures, frameworks, and cost-effective strategies for MTD deployment in SDN, IoT, and 5G	Non-learning-based frameworks	2016 2021	DoS/DDoS, scanning, false data injection
Application-specific (middle)	[1, 83, 58, 68, 38, 71, 18, 44, 54, 37, 43, 84]	Tailored MTD solutions for specific domains (5G slicing, vehicular networks, IoT, Android malware, edge clouds)	Mix of heuristic and learning-based methods	2020 2024	DDoS, malware, routing attacks, relay selection
RL-based (outer)	[21, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 53, 35, 36, 39, 40, 41, 42, 45, 46, 47, 48, 49, 50, 51, 85, 52, 20, 19]	Use of deep/multi-agent RL, federated learning, and meta-RL for optimizing MTD strategies in various environments	Reinforcement learning-based optimization	2019 2024	DDoS, route mutation, IP/port shuffling, deception, resource allocation

servation step), and optimization-based methods can provide strong constraint satisfaction when accurate models exist (e.g., Zhou et al. [2]). The emphasis on RL in this survey is therefore driven by its suitability for the *closed-loop actuation policy* of MTD in dynamic 5G environments, not by excluding complementary detection or optimization components.

Table XIX summarizes the decision elements in 5G MTD and shows why RL is frequently adopted for those elements in the surveyed literature, along with the key deployment caveats that must be considered to make RL-based MTD practical at scale.

B. What this survey teaches us

This survey shows that Moving Target Defense (MTD) becomes truly effective in 5G when it is treated as an operationally constrained control problem rather than a purely security-driven mechanism. A first lesson is that *domain context dominates feasibility*. The same mutation action can be realistic in one part of the 5G system and infeasible in another: RAN- and edge-facing components are constrained by tight latency budgets and mobility dynamics, the core is shaped by VNF chaining and orchestration dependencies, and slicing introduces tenant-level isolation and per-slice KPI enforcement. Therefore, the practical value of any MTD technique depends on explicitly mapping mutation targets and enforcement points to the correct domain (RAN, core, slicing, MEC/edge) and designing policies around the constraints and interfaces available there.

A second lesson is that *security gains are inseparable from operational cost*. Many studies demonstrate that shuffling addresses, perturbing routes, migrating VNFs, or reconfiguring policies can reduce attack success rates and increase time-to-compromise, but the same actions introduce measurable overhead: control-plane updates, convergence delays, transient packet loss during cutover, and potential throughput degradation. In 5G, these costs must be interpreted through per-slice SLA/KPI requirements, because an MTD policy that is acceptable for eMBB may still be unacceptable for URLLC due to latency/jitter sensitivity, and an action that is safe in the core may be costly at MEC due to limited resources. The central takeaway is that MTD should be evaluated as a *security QoS overhead trade-off*, and results that report only security metrics are insufficient to assess deployability.

A third lesson concerns *scalability and stability*. At 5G scale, the limiting factors are often not the existence of a mutation policy but the infrastructure's ability to sustain frequent change. Rule-space constraints (including flow-table/TCAM pressure), controller CPU load, and bursts of southbound updates can dominate feasibility, especially when mutations are fine-grained or synchronized across slices. Moreover, reactive policies and learning-driven controllers can induce policy oscillation when decisions respond to delayed or noisy telemetry, when traffic is non-stationary, or when multi-domain coordination is imperfect. As a result, practical MTD must incorporate rate limits (minimum dwell time), stability penalties, and explicit convergence constraints, and must report churn, convergence time, and transient disruption alongside security benefit.

A fourth lesson is that *learning-assisted MTD is promising but must be constraint-aware and robust*. Reinforcement learning and other adaptive methods can align mutation decisions with evolving threats and dynamic network conditions, but many existing formulations optimize attack disruption without enforcing QoS/SLA constraints or bounding control-plane overhead. Furthermore, policies trained in simplified simulators may fail under realistic 5G conditions such as mobility, bursty load, slice elasticity, and adaptive attacker behavior. For learning-assisted MTD to be deployable, it must incorporate explicit constraint handling (SLA-aware rewards or safe RL), realistic and diverse training conditions, and stress-testing against adaptive adversaries. Equally important, action spaces must reflect what can be executed in SDN/NFV stacks at scale, rather than assuming arbitrarily frequent migrations or rule rewrites.

A fifth lesson is that *benchmarking and reproducibility are currently a bottleneck*. The survey reveals a persistent

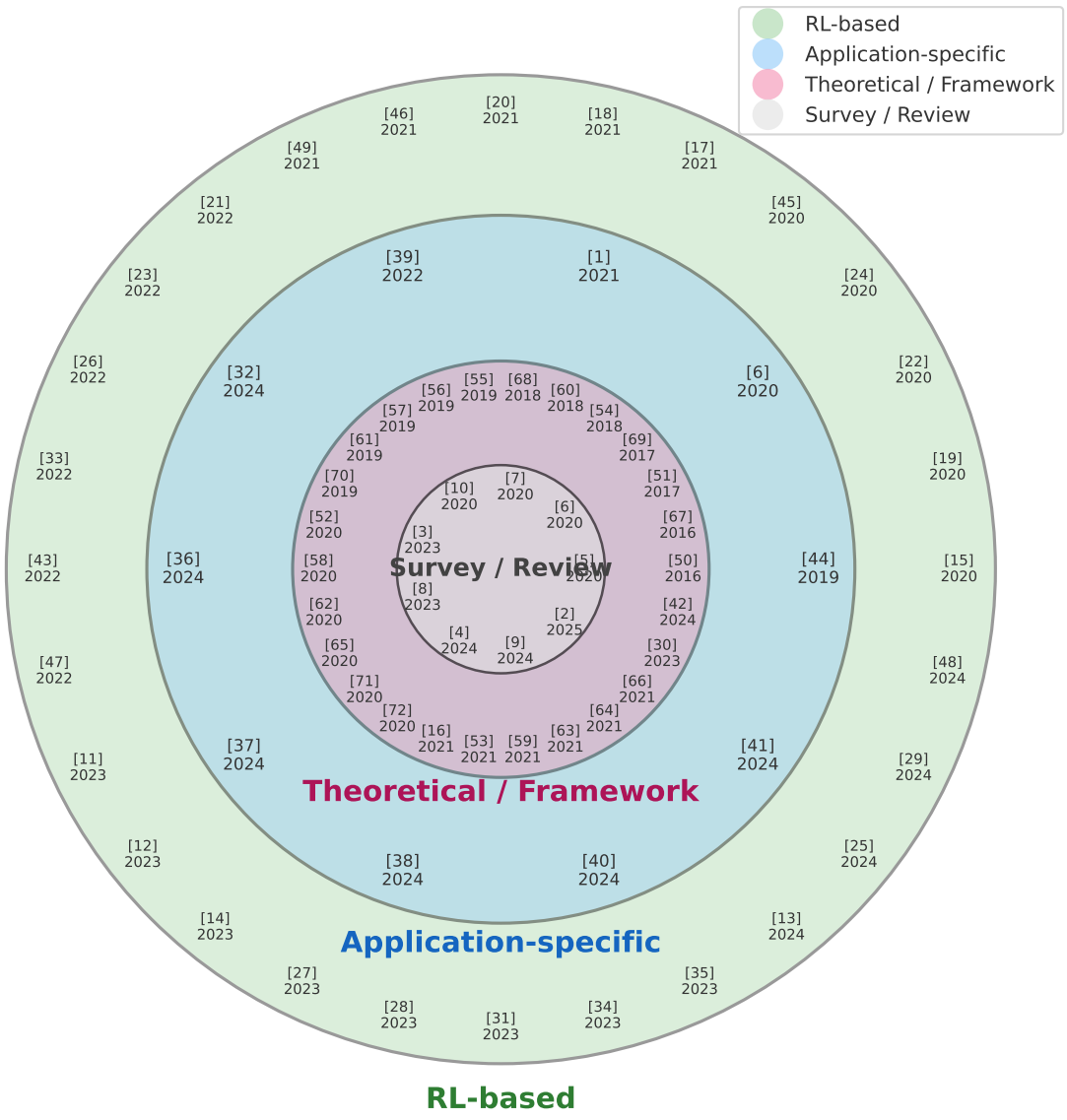


Fig. 6. Concentric circle diagram categorizing the reviewed references into four layers: the innermost circle represents *Theoretical/Framework* studies, the middle ring includes *Application-specific* works, the outer ring contains *RL-based* approaches, and the central node indicates *Survey/Review* papers. Colors highlight the thematic domains, while chronological ordering within each layer illustrates the temporal evolution of research in this field. (More details in Figure 20 in Appendix D)

gap: there is no consolidated dataset that jointly captures realistic 5G traffic dynamics, attack traces aligned with 5G threat surfaces, and ground-truth labels for both security events and service impact. Likewise, evaluations often rely on non-standard emulation/testbed setups with different topologies, controller choices, virtualization stacks, and mutation-enforcement mechanisms, making cross-paper comparison difficult. This lack of standard benchmarks slows progress because it obscures which improvements are truly algorithmic and which stem from experimental design. Establishing reference workloads, standard emulation environments, and unified reporting templates is therefore a prerequisite for rapid, reproducible advances.

A sixth lesson is that *mutation design must be guided by attacker adaptation*. Many approaches implicitly assume that attackers are stationary or only weakly adaptive; however, realistic adversaries observe network behavior and adjust scanning strategies, exploitation timing, and target selection. This implies that MTD should be evaluated not only by immediate attack disruption, but also by how quickly attackers can re-learn the surface and how mutation strategies remain effective over repeated cycles. This points to the importance of adversary-aware evaluation, longer-horizon experiments, and strategies that avoid predictable periodicity.

Finally, the survey indicates that *5G MTD should be engineered as a coordinated, multi-layer mechanism*. Network-layer hopping alone is often insufficient if higher-layer identifiers remain stable; conversely, virtualization-layer changes

TABLE XIX
WHY RL IS EMPHASIZED FOR 5G MTD DECISION-MAKING: DECISION ELEMENTS, 5G-SPECIFIC DIFFICULTY, RL ADVANTAGE, REPRESENTATIVE SURVEYED EXAMPLES, AND DEPLOYMENT CAVEATS.

MTD decision element	**Why it is difficult in 5G**	**Why RL fits (core benefit)**	**Examples in surveyed works**	**Industrial/deployment caveats**
Mutation interval / timing (when to mutate)	Non-stationary traffic (mobility, bursts), slice elasticity, and tight URLLC latency/jitter budgets make fixed schedules either too costly or too weak.	Learns context-aware timing that balances long-term security benefit vs. overhead; adapts mutation frequency to observed risk and network state.	RL for shuffling/scheduling decisions Chai et al. [25].	Must bound churn and enforce dwell times; evaluate convergence delay and transient loss under realistic controller/orchestrator latencies.
Target selection (what to mutate)	Large action space across domains (core, edge/MEC, slicing); multi-tenant constraints and cross-slice side effects complicate choices.	Optimizes sequential target choices under uncertainty; can incorporate risk and KPI signals to prioritize high-impact, low-cost targets.	Scanning-disruption policies Zhang et al. [20]; multi-action SDN/NFV decisions Chowdhary et al. [23].	Action feasibility depends on SDN/NFV interfaces; target changes may require synchronization with endpoints and slice policies.
Joint security QoS control (how aggressively to mutate)	Security actions can degrade per-slice KPIs; URLLC is sensitive to transient disruptions; MEC resource limits constrain feasible actions.	Reward/constraint design encodes security vs. QoS vs. cost; policy learns to avoid SLA violations while still disrupting attackers.	Slice-aware multi-agent control integrating QoS/security Yoon et al. [24].	Needs SLA-aware constraints and stability penalties; must report p95/p99 latency/jitter, loss bursts, and per-slice violations.
VNF/VM/service relocation (virtualization-layer MTD)	Migration/replication changes the true attack surface but incurs orchestration delay, state transfer, and potential cutover loss; hard to model statically.	Learns long-horizon trade-offs where short-term cost may reduce long-term compromise risk; can adapt relocation decisions to load/risk.	Robust/generalizable RL across configurations Li and Zheng [27]; SDN-native multi-action (migration and mutation) Chowdhary et al. [23].	Stateful chain dependencies and warm-up time must be modeled; migration may violate URLLC unless guarded by strict constraints.
Multi-domain / multi-agent coordination	5G spans multiple control loops (RAN/core/edge/slices); coordination delay and partial observability can destabilize reactive policies.	Multi-agent RL can coordinate decisions across domains/slices; learns cooperative policies under shared objectives.	Multi-agent slice-aware RL Yoon et al. [24]; multi-agent SDN settings Chowdhary et al. [23].	Risk of oscillation; requires bounded update rates, robust telemetry, and explicit stability evaluation under delayed measurements.
Comparison with alternatives (why not only optimization / heuristics)	Heuristics can be predictable; optimization needs accurate models and can be expensive to recompute under fast dynamics and attacker adaptation.	RL directly optimizes the closed-loop actuation policy; alternatives remain valuable as complements (detection, constraints, or fallback).	Optimization-based complementary perspective Zhou et al. [2].	Hybrid designs are often most practical: ML detection and constrained decision layer and safe RL actuation with fallback modes.

alone may be too costly without careful scheduling. The most robust direction is coordinated MTD that links target choice, mutation interval, and enforcement mechanism across domains while remaining SLA-aware and stability-bounded. These lessons collectively motivate the open problems discussed in this paper, including benchmark development, safe and stable learning-assisted control, cross-domain coordination, and forward-looking mutation targets for next-generation (5G/6G) architectures.

C. Industry and policy momentum: why MTD is timely and deployable in 5G

Beyond academic proposals, multiple *public signals* indicate that Moving Target Defense (MTD) is not merely conceptual, but an increasingly operational security direction. First, MTD appears in authoritative security guidance and resilience engineering frameworks, which treat controlled change as a systematic means to increase attacker uncertainty and reduce windows of opportunity. In parallel, telecom-oriented security documents (e.g., EU-level 5G security controls and 3GPP-aligned guidance) emphasize the need for dynamic and automation-friendly defenses to address softwarized and distributed 5G infrastructures. These policy and standards signals matter for 5G because they translate high-level resilience principles into concrete security engineering expectations: automation, repeatability, and measurable trade-offs against service performance.

A second signal is the presence of MTD explicitly in EU R&I programs and 5G/6G project outputs. Recent SNS JU call topics and project communications list moving target defence among targeted security capabilities, while 5G PPP/SNS-related deliverables and experiments report MTD controllers and use cases aligned with NFV/SDN and slicing workflows. This suggests that MTD is being treated as a practical building block within operator-grade security roadmaps, rather than a standalone academic technique. At the same time, the maturity signal does not remove research gaps; it strengthens them: real deployments require constraint-aware decision-making (latency/jitter budgets, per-slice SLA/KPIs, control-plane churn, rule-space pressure, orchestration delay), and these constraints are often under-modeled in the literature.

Third, commercial and platform-level adoption reinforces feasibility. The existence of a dedicated Automated Moving Target Defense (AMTD) market category, together with major endpoint-security vendor discussions of AMTD, indicates productization paths for moving-target concepts. Even more broadly, widely deployed platform mitigations such as address-space randomization in mainstream operating systems demonstrate that "moving target" mechanisms can be engineered and maintained at scale. Overall, these signals collectively support the argument that MTD is both relevant and under active development, while also highlighting why 5G-specific benchmarking, reproducible evaluation environments, and deployment-grounded designs remain important open needs. Representative public sources supporting these maturity signals are listed in Table XX.

D. Evidence of momentum and market pull for MTD in 5G/6G

While the preceding sections focus on technical designs and evaluation trade-offs, it is also important to justify why

TABLE XX
PUBLIC SIGNALS OF MATURITY AND MOMENTUM FOR MTD/AMTD (STANDARDS/POLICY, EU PROGRAMS, INDUSTRY, AND PLATFORM-LEVEL ADOPTION).

#	Category	Public source (clickable keyword)
1	Standards/Definition	NIST glossary: Moving Target Defense
2	Standards/Guidance	NIST SP 800-172 (Enhanced Security)
3	Resilience Engineering	NIST SP 800-160 v2 (Cyber Resiliency)
4	Government Research	NIST: Simulation-based MTD effectiveness
5	Government Research	NIST: Applying MTD to network security
6	Resilience Frameworks	MITRE: Cyber resiliency techniques
7	Resilience Frameworks	MITRE: Engineering Aid framework (update)
8	EU 5G Controls	ENISA: 5G Security Controls Matrix
9	EU 5G Controls (PDF)	ENISA: 5G controls booklet (PDF)
10	EU 5G Specs Guidance	ENISA: Security in 5G specifications
11	3GPP Security Spec	3GPP TS 33.501 (Security architecture)
12	3GPP Security Events	3GPP TS 33.502 (Security procedures)
13	ETSI/3GPP PDF (33.501)	ETSI PDF: TS 133 501 v18.06.00
14	ETSI NFV Security	ETSI NFV-SEC 013 (Security guidance)
15	EU Call Topic (SNS)	Horizon SNS call topic (Stream B)
16	SNS JU Achievements	SNS JU: Key achievements (2025)
17	SNS JU News/Updates	SNS JU: Newsflash (Nov 2025)
18	5G PPP / EU Brochure	5G PPP: 10 Technology Priorities (2023)
19	5G PPP Experiment	INSPIRE-5Gplus: MTD experiment page
20	5G PPP Deliverable (MTD controller)	INSPIRE-5Gplus D3.4: Smart 5G security
21	CORDIS Project Results	CORDIS: Project results (ID 871808)
22	SNS Project Deliverable	NATWORK D3.1: Secure-by-design O&M (PDF)
23	Market Category (AMTD)	Gartner: AMTD market category
24	Major Vendor (AMTD)	Sophos: AMTD blog (pioneering)
25	Vendor Reference (AMTD)	Morphisec: AMTD overview
26	Financial-sector Focus (public article)	Morphisec: Financial-sector regulatory pressure
27	Public Case/Stories (vendor portal)	Morphisec: Customer stories
28	Platform Adoption (Microsoft ASLR)	Microsoft: Exploit protection (ASLR)
29	Platform Adoption (Apple ASLR)	Apple Platform Security: ASLR
30	Platform Adoption (Linux KASLR)	Linux kernel: Self-protection (KASLR)

MTD is a timely and practically relevant direction for 5G and beyond. Publicly accessible signals show that MTD is increasingly discussed in telecom-security guidance, standardization roadmaps, and EU R&I agendas, and that "moving-target" style mechanisms are considered compatible with automation-centric operations in softwarized networks. These signals do not eliminate the research challenges; rather, they motivate more deployment-grounded work, including benchmarking, reproducible experimentation, and rigorous reporting of security QoS overhead trade-offs under slice-level SLA constraints.

Table XXI consolidates representative public references across standards/policy, EU programs/projects, and academic/industry artifacts. The intent is not to claim universal adoption, but to demonstrate sustained and multi-stakeholder attention to MTD/AMTD concepts and to highlight that the area is mature enough to attract operational interest while still presenting clear open research gaps (e.g., cross-domain enforcement, stable policies under churn, and realistic datasets/emulation environments). For completeness, a small portion of evidence is vendor-published case material; such entries are included as market signals and should be interpreted as self-reported.

Overall, these public signals support two complementary conclusions. First, MTD/AMTD is increasingly discussed as an automation-friendly resilience mechanism for softwarized networks and adjacent security stacks. Second, despite this momentum, the lack of standardized benchmarks, consolidated datasets, and deployment-grade reporting of KPI impact means that substantial research remains necessary to translate MTD from concept demonstrations into reproducible, operator-ready defenses for 5G/6G environments.

E. Additional concluding perspective

Taken together, the evidence reviewed in this survey suggests that the next phase of MTD research should prioritize deployment realism over isolated security gains. In particular, future work should 1) design mutation policies that explicitly preserve per-slice KPIs and URLLC constraints, 2) quantify overhead and convergence behavior under multi-slice load and large-scale SDN/NFV control, 3) develop benchmark suites that couple realistic traffic, attacks, and service-impact labels, and 4) validate robustness against adaptive adversaries and non-stationary operating conditions. These steps would transform MTD from a promising concept into a practical and measurable security capability for production-grade 5G and beyond.

XII. CHALLENGES IN DEPLOYING MTD IN 5G NETWORKS

Despite significant progress in MTD research for 5G networks, several challenges hinder large-scale deployment. Real-time decision-making remains a major issue, as RL-based methods such as [25, 26, 24, 38] can suffer from inference delays that conflict with ultra-reliable low-latency communication (URLLC) requirements. Scalability and resource overhead are also critical concerns, with SDN/NFV-based approaches like [59, 61, 63, 27] facing bottlenecks in handling frequent flow rule updates and virtual function migrations in multi-domain networks. Furthermore, integrating MTD mechanisms with network slicing and orchestration frameworks is complex; while slice-aware strategies have been explored in [53, 64, 77], ensuring isolation and SLA compliance across multiple tenants remains unresolved. Interoperability across heterogeneous architectures presents another challenge, as highlighted in [56, 66, 65], where vendor-specific APIs and proprietary configurations limit policy unification. Moreover, most evaluations, such as [72, 78, 81, 82], are conducted in simulation or small-scale testbeds that fail to capture real-world traffic diversity and multi-vector attack patterns. Security–performance trade-offs persist, with studies like [62, 80, 58, 57] showing that frequent mutations can degrade QoS if not optimally tuned. Finally, ML- and RL-based MTD methods rely on large-scale, realistic datasets, yet data scarcity and privacy constraints force many works [76, 60, 63] to depend on synthetic traffic, which may limit generalization. Addressing these challenges will require cross-layer design, standardized APIs, privacy-preserving data sharing, and large-scale experimentation in production-grade 5G/6G testbeds.

TABLE XXI
PUBLIC EVIDENCE SIGNALS INDICATING MOMENTUM FOR MTD/AMTD IN 5G/6G (POLICY/STANDARDS, EU AGENDAS, RESEARCH/TESTBEDS, AND MARKET SIGNALS).

#	Source type	Evidence signal (what it demonstrates)	Public URL
1	Industry guidance	GSMA practitioner-facing 5G security guide (operator-oriented security priorities)	GSMA 5G Security Guide v3.0
2	Industry library	GSMA curated cybersecurity document library (sustained 5G security ecosystem outputs)	GSMA Cybersecurity Document Library
3	Standards roadmap	ITU-T compilation referencing major 5G security guidance (standardization landscape signal)	ITU-T ICTS 2022 tutorial (PDF)
4	EU agency report	ENISA report on NFV security in 5G (practical risks/controls for virtualized 5G)	ENISA: NFV Security in 5G (PDF)
5	EU agency report	ENISA 5G security standards mapping (evidence of active standardization and guidance)	ENISA: 5G Standards mapping (PDF)
6	EU agency update	ENISA publication on 5G security measures under EECC (regulatory/measure perspective)	ENISA: 5G measures under EECC
7	ETSI spec (PDF)	NFV security management requirements (monitoring/policy lifecycle constraints)	ETSI NFV-IFA 026 (PDF)
8	ETSI work item	NFV security mgmt update work item (signals ongoing evolution of NFV security management)	ETSI Work Item 58648
9	ETSI ecosystem	ETSI NFV technology hub (living ecosystem and continuously updated specs/test suites)	ETSI NFV Technology Hub
10	NIST controls (PDF)	NIST SP 800-53 controls include concealment/misdirection family (foundation for MTD-like controls)	NIST SP 800-53r5 (PDF)
11	EU program page	SNS JU key achievements explicitly highlighting MTD among innovations (program-level momentum)	SNS JU achievements 2025
12	EU program update	SNS JU newsflash highlighting MTD among key innovations (continued emphasis in 2025)	SNS JU newsflash (Nov 2025)
13	EU program (PDF)	SNS Journal 2025 (broad project portfolio mentioning MTD among security directions)	SNS Journal 2025 (PDF)
14	EU program (PDF)	SNS Journal 2024 (community/roadmap signal that includes MTD in security context)	SNS Journal 2024 (PDF)
15	EU whitepaper (PDF)	SoftNet whitepaper mentioning integration of SOAR and MTD (operations and automation context)	SoftNet whitepaper 2024 (PDF)
16	EU project (PDF)	NATWORK benchmark/SoA report referencing MTD as orchestration-era defense option	NATWORK D2.1 SoA & Benchmark (PDF)
17	Open-source/testbed	ETSI OSM research notes referencing applying MTD in MEC ecosystem (hands-on feasibility signal)	ETSI OSM research notes
18	CORDIS project	5GROWTH results page listing MTD-related outputs (EU project dissemination signal)	CORDIS: 5GROWTH results
19	CORDIS project	INSPT project results including MTD-related slice protection output (EU research signal)	CORDIS: INSPT results
20	CORDIS project	CORDIS results listing IPv6 shuffling for MTD in SDN (continued R&I outputs)	CORDIS: Project 101139285 results
21	Academic (telco/MEC)	Peer-reviewed MTD for cloud/edge telco environments (MEC/NFV relevance)	Peer-reviewed telco/MEC MTD (ScienceDirect)
22	Academic metadata	DOI/bib entry for telco/MEC MTD paper (traceability and citation anchor)	Bibliographic entry (IT.pt)
23	Academic (edge services)	Peer-reviewed TOTP-style port mutation MTD for sensitive network services	Port-mutation MTD (ScienceDirect)
24	Academic (B5G/6G)	IEEE-style article PDF on MTD as proactive defense element beyond 5G	MTD beyond 5G/6G (PDF)
25	Academic (DRL+MTD)	DRL-based routing randomization MTD (programmable networks and learning signal)	DRL routing randomization MTD (ScienceDirect)
26	Academic (slicing)	DRL+MTD for network slicing security (explicit 5G slicing context)	DRL+MTD for slicing (Springer)
27	Academic (O-RAN)	MTD-based secured slicing in O-RAN (RAN intelligence + MTD for ML robustness)	O-RAN secured slicing (arXiv)
28	Industry case (finance)	Vendor-published AMTD case study in banking (market signal; self-reported)	Morphisec case: Merrick Bank (PDF)
29	Industry case (finance)	Vendor-published AMTD case study in hedge fund environment (market signal; self-reported)	Morphisec case: Hedge Fund (PDF)
30	Industry case (finance)	Financial services case study page describing AMTD-style "hopping" (market signal; self-reported)	HOPR case: Financial services

A. *Benchmark gaps and reproducible evaluation for 5G MTD*

A key barrier to progress in 5G-oriented Moving Target Defense (MTD) is the lack of shared benchmarks that enable reproducible and fair comparison across studies. In the reviewed literature, evaluations are often conducted using custom simulators or small-scale SDN/NFV testbeds, with different topology sizes, traffic assumptions, attack implementations, and mutation-enforcement mechanisms. As a result, reported improvements can be difficult to interpret across papers, and it remains unclear whether gains stem from the MTD policy itself or from choices in the experimental setup. This issue is particularly critical in 5G networks because MTD decisions must jointly respect security objectives and strict operational constraints (e.g., per-slice SLA/KPI compliance, URLLC latency/jitter sensitivity, and MEC resource limits). Table XXII summarizes the main benchmarking gaps observed in existing studies and outlines the minimum components needed for a deployment-relevant and reproducible evaluation pipeline.

Beyond documenting these gaps, the benchmark requirements in Table XXII also indicate how future work can make results more deployment-relevant. First, evaluations should explicitly model slice-level objectives and constraints by reporting per-slice KPIs and SLA violations under mutations, especially when actions involve routing changes, rule updates, or VNF/service migration (e.g., slice-aware decision-making in [24, 54]). Second, mutation feasibility should be validated under realistic control-plane and orchestration overheads, since frequent reconfiguration can trigger controller load, convergence delays, and policy oscillation at scale (e.g., SDN-based rule-update pipelines in [61, 63]). Finally, RL-based MTD should be evaluated for robustness and generalization under non-stationary workloads rather than single fixed scenarios, because 5G traffic dynamics and attacker adaptation can invalidate policies learned in simplified environments (e.g., [25, 20, 27]). Establishing open, standardized benchmarks along these dimensions would enable fair comparison, accelerate reproducible research, and support practical adoption of adaptive MTD in 5G and beyond.

1) Typical 5G operating ranges for contextualizing metrics: Although the evaluation metrics used for MTD (e.g., attack success reduction, time-to-compromise, reconfiguration overhead, convergence delay, and QoS impact) are well-defined, interpreting them requires a practical sense of what constitutes "acceptable" performance in operational 5G networks. In practice, operators enforce different SLA/KPI targets depending on the service class; therefore, the same MTD action may

TABLE XXII
BENCHMARKING GAPS FOR 5G MTD AND RECOMMENDED BENCHMARK COMPONENTS (EXAMPLES ARE NON-EXHAUSTIVE).

Gap (what is missing)	**Why it matters in 5G**	**What a practical benchmark should provide**	**Examples of current practice**
Consolidated datasets that couple traffic, attacks, and ground truth	RL/ML-based MTD needs realistic, non-stationary data (mobility, bursty load, slice elasticity). Without labels for security events and service impact, policies are hard to validate and compare.	A curated dataset set containing benign multi-service traffic, attack traces (scan/DoS/DDoS/FDI/route manipulation), and aligned labels for both security outcomes and QoS/SLA impact.	Many studies use synthetic or scenario-specific traces for RL-based decisions and SDN-based shuffling, limiting cross-paper comparability (e.g., [25, 24, 20, 63]).
Standardized 5G reference topologies and domain mappings	MTD feasibility depends on where it is applied (RAN/core/slices/MEC) and on domain-specific constraints. Different topologies can dominate the measured overhead and convergence behavior.	Reference topology templates that explicitly map RAN, core, slicing entities, and edge/MEC, including multi-tenant slices and realistic service chains; multiple scales (small/medium/large) for stress testing.	Evaluations often rely on custom SDN/NFV setups with different sizes and assumptions (e.g., SDN-based multi-controller settings [61], DDoS-oriented SDN studies [59], and slice-aware RL frameworks [24, 54]).
Standard emulation/testbed environment for SDN/NFV-enabled 5G MTD	The same MTD action may have very different cost depending on controller/orchestration delays, rule-installation latencies, and virtualization/migration mechanisms, which are rarely standardized.	A reference emulation stack (SDN + NFV/MEC) with a common interface to execute mutations (targets/parameters/intervals), log control-plane actions, and replay identical workloads/attacks across methods.	Many works implement MTD with bespoke SDN rule logic or custom orchestration assumptions (e.g., IP/port mutation in SDN [56, 60, 63], migration-based MTD [55, 54]).
Unified evaluation protocol that jointly reports security benefit and operational cost	In 5G, security improvements are not sufficient if they violate URLLC/MEC constraints or degrade per-slice KPIs. Without consistent metrics, "better" results are ambiguous.	A reporting template that includes both security metrics (e.g., time-to-compromise, reconnaissance disruption, attack success reduction) and operational metrics (e.g., control-plane churn, convergence time, rule-space/TCAM pressure, cutover loss, p95/p99 latency, jitter, throughput, SLA violations).	Security-only reporting is common, while QoS and control-plane costs are inconsistently reported across SDN shuffling and RL-based scheduling studies (e.g., [62, 25, 24]).
Reproducibility artifacts (config snapshots, mutation timelines, seeds, and logs)	Without artifacts, it is difficult to reproduce results, verify assumptions, and study sensitivity to traffic mixes, randomization, and attacker adaptation.	Release of 1) topology and controller/orchestrator configs, 2) mutation schedules/timelines and rule-update traces, 3) traffic/attack generation scripts, and 4) raw logs plus analysis scripts.	Most papers describe setups at a high level but do not provide sufficient artifacts to replay experiments exactly, especially for multi-step mutation strategies and RL training pipelines (e.g., [27, 20]).

be feasible for one slice while unacceptable for another. This is particularly important in 5G, where mutation events (e.g., routing/flow updates, IP/port hopping, VNF migration, or slice reconfiguration) can introduce transient disruption such as short packet loss bursts, brief latency spikes, or control-plane churn. To contextualize the reported metrics in this survey, Table XXIII summarizes representative, order-of-magnitude ranges for common 5G service expectations. These indicative ranges help the reader assess whether an MTD policy that improves security also remains compatible with service requirements, especially for URLLC and edge/MEC deployments where latency/jitter budgets and resource constraints are strict.

In addition to service-level KPIs, deployment-realistic evaluation should report the operational overhead introduced by MTD, including the rate of control-plane updates, configuration convergence time, and any transient packet loss during cutover. When RL-based MTD is used, it is also important to report whether the learned policy explicitly incorporates SLA constraints (e.g., penalties for latency/jitter violations) and whether it remains stable under non-stationary conditions such as mobility, bursty traffic, and slice elasticity. These contextual ranges therefore serve as a practical reference point for interpreting the security QoS overhead trade-offs discussed throughout the survey.

2) Scalability constraints in SDN/NFV-enabled 5G: Scalability is a key practical constraint for deploying MTD in 5G, because many mutation actions ultimately translate into frequent changes in forwarding rules, service-function chains, slice policies, and/or VNF placements across a large number of devices and tenants. A first bottleneck is the limited rule capacity of data-plane elements: programmable switches and virtual switches maintain finite flow tables and, for high-speed matching, may rely on expensive TCAM resources; therefore, fine-grained or large-scale mutations (e.g., per-flow shuffling, frequent path perturbation, or wide-scope access-control updates) can quickly inflate the number of rules, increase lookup complexity, or force rule compression/aggregation that weakens policy expressiveness. A second bottleneck is control-plane churn. In SDN/NFV-based 5G, each mutation can trigger bursts of controller-to-switch updates, orchestration actions, and consistency checks; at scale, this raises controller CPU load, consumes southbound bandwidth, and increases config-

TABLE XXIII
REPRESENTATIVE 5G SERVICE EXPECTATIONS (INDICATIVE RANGES) USED TO CONTEXTUALIZE MTD EVALUATION METRICS. EXACT THRESHOLDS VARY BY OPERATOR AND APPLICATION, BUT THE TABLE PROVIDES PRACTICAL ORDER-OF-MAGNITUDE GUIDANCE.

Service	**Latency / jitter expectations**	**Reliability / loss expectations**	**Implication for MTD evaluation**
URLLC	Very low end-to-end latency (order of a few ms to $\sim$10 ms) and tight jitter tolerance	Very high reliability (often "five nines" or higher) and extremely low loss (typically 10^{-5} order or lower for critical traffic)	Mutation-induced transient disruption must be tightly bounded; convergence delay, packet-loss bursts, and frequent rule updates can quickly violate SLA.
eMBB	Moderate latency tolerance (often tens of ms) with moderate jitter sensitivity	Reliability requirements lower than URLLC; performance dominated by throughput stability	MTD should avoid sustained throughput drops; large-scale reconfigurations and heavy control-plane churn should be rate-limited.
mMTC	Latency can be relaxed (often hundreds of ms, sometimes more depending on the application)	Reliability varies; large-scale connectivity and energy efficiency are central	Scalability and overhead dominate: rule-space growth, controller/orchestrator load, and per-device mutation cost must remain bounded.

uration convergence time, which may manifest as transient packet loss, latency spikes, or short service interruptions effects that are particularly harmful for URLLC slices and for latency-sensitive edge/MEC services. A third issue is stability: reactive or learning-driven mutation policies can create policy oscillation when updates are applied too aggressively or when the system operates with delayed telemetry and non-stationary traffic (e.g., mobility, bursty workloads, slice elasticity). In multi-controller or multi-domain settings, coordination delays and conflicting objectives can further amplify oscillations, causing repeated reconfigurations that degrade both security and QoS. These scalability realities imply that deployment-realistic MTD should explicitly bound mutation rates (e.g., minimum dwell time and rate limits), prefer domain- or slice-scoped actions when rule space is tight, and incorporate operational costs (rule-space growth, update rate, convergence delay, and transient loss) into the optimization objective especially for RL-based methods. In addition, practical designs should evaluate worst-case update bursts under multi-slice workloads, include safeguards such as graceful degradation (fallback to less frequent mutations under congestion), and avoid synchronized cross-slice mutations that can create correlated churn on shared infrastructure.

B. Critical analysis across major MTD categories

Beyond summarizing prior work, it is essential to clarify recurring limitations that affect the deployability of MTD in 5G-oriented environments. Although many studies show that MTD can disrupt reconnaissance, increase attacker uncertainty, and reduce attack success probability, the reported gains are often sensitive to operational constraints, attacker adaptivity, and evaluation realism. To strengthen the analytical depth of this survey, we provide a category-level critique that explains where existing approaches tend to fall short and what should be reported to support deployment-realistic assessment. Table XXIV summarizes the main limitations and the corresponding evaluation items that are most informative for 5G practice.

a) IoT and end-device-centric MTD.: IoT-focused MTD commonly prioritizes lightweight actions (e.g., identity/address changes, protocol-level perturbations, or endpoint-level diversification) to respect constrained compute, energy, and connectivity. However, evaluations frequently simplify the heterogeneity that dominates real IoT deployments, including mixed vendor stacks, uneven patch levels, intermittent connectivity, and legacy protocols that constrain feasible mutations. Another recurring limitation is incomplete modeling of attacker persistence and adaptation: even if low-layer identifiers change, higher-layer fingerprints (e.g., protocol behaviors, timing patterns, application identifiers) may remain stable, enabling repeated re-identification. In addition, limited endpoint telemetry can reduce the reliability of risk estimation and lead to poorly timed mutations that waste energy or disrupt service. Consequently, results may not transfer directly to large-scale device fleets unless studies explicitly model device diversity, energy overhead, and attacker re-identification strategies.

b) SDN-based MTD in programmable networks.: SDN enables rapid policy enforcement through centralized control, making it attractive for implementing MTD via address/port hopping, route perturbation, dynamic ACLs, or flow-rule randomization. Nevertheless, many SDN-based papers focus on security benefits while providing limited quantification of operational costs that become dominant at 5G scale. In practice, frequent mutations can inflate rule-space usage, increase controller CPU load, and generate bursts of southbound updates; these effects can lengthen convergence time and cause transient packet loss or latency spikes. Such transient disruptions are especially problematic for URLLC slices and for edge services with tight latency/jitter budgets. Stability is another concern: when mutation decisions react to noisy telemetry or non-stationary traffic, policies can oscillate (rapidly switching configurations) and amplify control-plane churn. Therefore, deployment-realistic SDN-based MTD must evaluate bounded update rates, convergence behavior, and rule-space pressure under multi-slice workloads, and should explicitly discuss how policies avoid oscillation.

c) Cloud/Edge/MEC and NFV/virtualization-aware MTD.: Virtualization-aware MTD can provide strong disruption against targeted exploitation by changing the actual hosting surface through VNF/VM/container migration, service relocation, replica rotation, or dynamic placement across edge and cloud. However, many works simplify the costs and feasibility constraints that arise for stateful and chained network functions, including state-transfer overhead, warm-up time, dependency constraints along service-function chains, and orchestration latency. These constraints are amplified in MEC/edge deployments where capacity is limited and distributed coordination is harder, so mutation actions that appear feasible in a centralized cloud setting may become costly or unsafe at the edge. Another frequent simplification is assuming that migration or replication can occur without measurable service disruption; in reality, cutover periods can produce brief loss spikes or delay increases, which may violate per-slice SLA/KPIs. As a result, meaningful evaluation should report migration time, cutover loss, orchestration delays, and the impact on end-to-end KPIs under realistic workloads and resource constraints.

C. MTD for 5G slicing and multi-tenant environments.

Slice-aware MTD is inherently constrained by per-slice isolation requirements and SLA/KPI preservation, yet the literature sometimes assumes ideal isolation and static tenant behavior. In operational systems, slices share physical infrastructure and often share portions of control-plane capacity, so security-driven changes in one slice may indirectly affect others through shared rule space, shared controller/orchestrator load, or shared edge resources. A recurring limitation is that cross-slice side effects and fairness considerations are not always quantified, even though they determine whether an MTD policy is acceptable in multi-tenant settings. Moreover, slice behavior is not static: elasticity (scale up/down), mobility-driven demand changes, and dynamic policy coordination can change feasible mutation schedules over time. Therefore,

evaluations should report per-slice KPIs and SLA violations, cross-slice interference indicators, and the sensitivity of results to slice elasticity and multi-tenant contention.

D. RL-based (learning-assisted) MTD.

Learning-assisted MTD provides adaptive decision-making under evolving threats and complex network dynamics, but several limitations recur across RL-based formulations. First, many models optimize attack disruption without explicitly enforcing QoS/SLA constraints, which is risky for URLLC and latency-sensitive MEC services; unless reward functions include explicit penalties for latency/jitter violations and instability, policies may select mutations that are security-optimal but operationally unacceptable. Second, training and evaluation often rely on simplified simulators and synthetic traces; consequently, policies may not generalize to non-stationary 5G conditions such as mobility, bursty traffic, slice elasticity, and shifting attacker strategies. Third, practical feasibility is sometimes under-discussed: assumed action spaces may be too broad relative to SDN/NFV scalability limits, and aggressive mutation rates can induce control-plane churn and policy oscillation, especially under delayed telemetry and multi-controller coordination. Finally, attacker modeling is frequently stationary or weakly adaptive, whereas realistic adversaries can learn mutation patterns and adjust their timing. For these reasons, RL-based MTD should be evaluated with explicit overhead and stability metrics, constraint-aware objectives, robustness tests under non-stationary workloads, and adversary-aware scenarios.

As summarized in Table XXIV, deployment-realistic MTD evaluation in 5G requires jointly reporting security benefits and operational costs, explicitly modeling per-slice QoS/SLA constraints (especially for URLLC and MEC), and validating stability and scalability under realistic SDN/NFV conditions. These considerations motivate the benchmarking requirements and open research directions discussed in the subsequent sections.

E. Industrial users and real-world deployment of Moving Target Defense (MTD)

In real industrial environments, the primary users of Moving Target Defense (MTD) are operational teams who must maintain continuous service delivery while systematically reducing exploitable exposure. Typical users include (i) SOC/NOC teams in mobile network operators and private-5G operators, (ii) enterprise IT/OT security teams in manufacturing, logistics, energy, and healthcare, and (iii) platform and site-reliability engineers operating cloud-native workloads at the edge (MEC) and in core data centers. For these users, MTD is considered valuable only when it behaves as an engineered operational control (with observability, change windows, and rollback), rather than a disruptive security experiment that risks uptime and SLA violations.

In practice, deployed MTD appears as a layered set of "safe-to-move" surfaces that can change without breaking the service contract. At the endpoint and workload layer, it is often realized through runtime randomization and exploit-mitigation mechanisms (e.g., memory-layout randomization and hardening controls) that continuously increase attacker cost and fragility of exploits. At the network and service layer, it is implemented via controlled mutation of reachable surfaces (e.g., rotating exposure points, shuffling placements of virtualized functions, or periodically refreshing service instances). In 5G and edge deployments, MTD naturally aligns with SDN/NFV and cloud-native lifecycle management, where network functions and services are already instantiated, scaled, and updated continuously enabling MTD to be integrated into automation pipelines instead of being performed manually.

Operational success in the real world typically follows a disciplined playbook: define a mutation budget (what can move, how frequently, and under what triggers), instrument end-to-end monitoring (latency/jitter baselines, alarms, and fault isolation), and enforce governance (authorization, exceptions, and rollback procedures). Organizations commonly start with low-risk scopes (non-critical slices, limited endpoint groups, or edge test clusters), validate that mutation does not degrade QoS, then scale coverage while keeping a human-in-the-loop for policy tuning. In this sense, real-world MTD is best understood as a controlled "continuous change" capability whose effectiveness is evaluated not only by security gains, but also by operational compatibility, auditability, and predictable recovery behavior. All public sources and website links supporting these real-world signals are compiled in Table XXV.

Figure 7 summarizes how Moving Target Defense (MTD) is applied in real industrial operations as a controlled and measurable "continuous change" capability rather than an ad-hoc security add-on. In practice, the main users include operator SOC/NOC teams (public and private 5G), enterprise IT/OT security teams in industrial environments, and platform/SRE teams operating cloud-native workloads at the edge. Deployed mechanisms typically combine endpoint/workload-level randomization and exploit mitigation with network/service-level controlled mutation, executed through automation and lifecycle management aligned with 5G/NFV and cloud operations. To keep these changes compatible with compliance and service constraints, organizations commonly rely on publicly available baselines and architectural references; all supporting public sources and website links are provided in Table XXV.

XIII. Integrated Perspectives: ML Support, Secure Learning, and 6G Outlook

The final part of the manuscript consolidates three themes that are essential for a realistic MTD agenda in 5G and beyond: the supporting role of machine learning beyond RL, the security of learning-assisted MTD itself, and the transition from 5G deployments to 6G-oriented research directions. Presenting these themes together preserves the central identity of the survey by treating them as extensions of the same MTD control problem rather than as detached side topics.

In particular, supervised and unsupervised learning support telemetry interpretation and risk estimation, RL supports sequential actuation, and secure-learning analysis explains how these adaptive defenses can themselves become attack

TABLE XXIV
CATEGORY-LEVEL CRITIQUE: RECURRING LIMITATIONS AND RECOMMENDED EVALUATION/REPORTING ITEMS FOR DEPLOYMENT-REALISTIC MTD IN 5G.

Category	Recurring limitations (typical gaps)	What should be explicitly evaluated/reported
IoT / end-device MTD	Device heterogeneity simplified; limited telemetry; attacker persistence and re-identification under-modeled; energy/connectivity constraints not fully quantified.	Device diversity assumptions; energy/battery cost; connectivity constraints; resistance to fingerprinting/re-identification; scalability to large fleets.
SDN-based MTD	Security gains emphasized without full operational cost; rule-space growth/TCAM pressure, controller load, update bursts, and convergence delay inconsistently reported; oscillation risk under reactive control.	Rule-space/TCAM utilization; controller CPU and south-bound update rate; convergence delay; transient loss/latency spikes; stability under multi-controller and multi-slice load.
Cloud/Edge/MEC and NFV MTD	Migration/relocation feasibility and stateful dependencies simplified; orchestration latency under-modeled; MEC resource limits and cutover disruption often omitted.	Migration time and cutover loss; state-transfer overhead; service-chain dependency constraints; orchestration delays; edge capacity limits and backhaul variability.
5G slicing / multi-tenant MTD	Cross-slice side effects and fairness not fully quantified; per-slice SLA/KPI preservation sometimes incomplete; slice elasticity dynamics under-modeled.	Per-slice KPIs and SLA violations; cross-slice interference indicators; shared control-plane/data-plane contention; elasticity scenarios; fairness across tenants.
RL-based MTD	QoS/SLA constraints often missing; training in simplified simulators; limited robustness/generalization; feasibility at SDN/NFV scale unclear; churn/oscillation risk.	Constraint-aware objectives (QoS/SLA); stability penalties and bounded mutation rates; generalization under non-stationary workloads; overhead and convergence metrics; adaptive-attacker evaluation.

TABLE XXV
REAL-WORLD MTD USERS AND DEPLOYMENT SIGNALS (ALL PUBLIC LINKS ARE PROVIDED INSIDE THE TABLE).

No.	Surface / signal	Typical industrial users	How it is used in the real world (practical interpretation)	Public link(s)
1	MTD baseline definition	Security architects, governance/audit teams	Used to align scope and terminology (what counts as MTD in requirements, assurance, and internal policy).	NIST glossary: MTD
2	Engineering cyber resilience (design guidance)	Critical-infra operators, telecom vendors, system engineers	Provides an engineering framing for proactive/architectural measures; often used to justify MTD-style controls as part of resilience-by-design.	NIST SP 800-160 v2r1 (Cyber Resiliency)
3	Resiliency technique catalog (MTD-relevant families)	Enterprise security architecture, mission/industrial systems	Used to map MTD mechanisms to resilience objectives and to reason about trade-offs (security benefit vs. operational impact).	MITRE: Cyber resiliency techniques
4	5G security control baselines	MNO/private-5G operators, regulators, assurance teams	Used as control baselines; supports introducing "controlled change" without violating telecom compliance expectations.	ENISA 5G controls matrix (PDF)
5	EU policy pressure toward verifiable controls	Operators, national authorities, regulated sectors	Drives adoption of implementable and auditable controls; indirectly favors deployable MTD-like measures compatible with assurance.	EU 5G cybersecurity policy page
6	5G system security architecture reference	5G core/edge implementers, telecom vendors	Serves as the "do-not-break" reference: any MTD mutation must remain compatible with 5G security procedures and architecture.	ETSI TS 133 501 (PDF)
7	NFV security management/monitoring reference	NFV/MANO teams, telecom cloud operators	Used to structure safe mutation: lifecycle control, monitoring, and governance to avoid unstable or non-auditable changes.	ETSI NFV-SEC 013 (PDF)
8	Endpoint exploit-mitigation as "always-on" MTD primitives	Endpoint engineering, IT/OT security teams	Deployed at scale as prevention/hardening layers that reduce reliability of memory-based exploitation with minimal ops overhead.	Microsoft: Exploit protection (ASLR)
9	OS-level randomization (ASLR) in production platforms	Platform engineering, secure OS baselines	Treated as built-in MTD: continuous randomization of memory layout increases exploit cost without changing application logic.	Apple: ASLR (Platform Security)
10	Linux kernel self-protection (incl. randomization context)	Linux server/edge operators, OT gateways	Adopted via hardened images and kernel configurations to raise the bar for kernel-level exploitation in edge/industrial nodes.	Linux: Kernel self-protection
11	Cloud-native "refresh by redeploy" operational pattern	Platform/SRE teams, edge operators (MEC)	Practical MTD-like deployment: frequent rolling updates/immutable deployments reduce dwell time and limit stable attacker footholds.	Kubernetes: Deployments (rolling updates)
12	Commercial AMTD category signal (endpoint-focused)	CISOs, MDR/MSPs, procurement	Indicates real market adoption: MTD delivered as a product capability aimed at prevention with low operational friction.	Morphisec: AMTD overview
13	Industrial/enterprise customer-story signals (deployment narrative)	Security leadership, enterprise operations	Public adoption narratives showing how organizations position AMTD/MTD as a preventive layer complementary to EDR/SOC tooling.	Morphisec: Customer stories
14	5G vertical trials signal (operational constraints reality check)	5G vertical operators, industrial consortia	Demonstrates that 5G is deployed across verticals with strict KPIs; MTD must be engineered to respect these constraints.	5G PPP: Verticals cartography

surfaces. The 6G outlook then extends the same logic to more programmable and AI-native environments, where mutation policies must remain robust under even tighter latency, coordination, and trust constraints.

1) Machine learning tools as defensive enablers in 5G (beyond RL): Machine learning is increasingly used as a practical defense layer in 5G systems because many attacks manifest as changes in telemetry patterns before they become service-

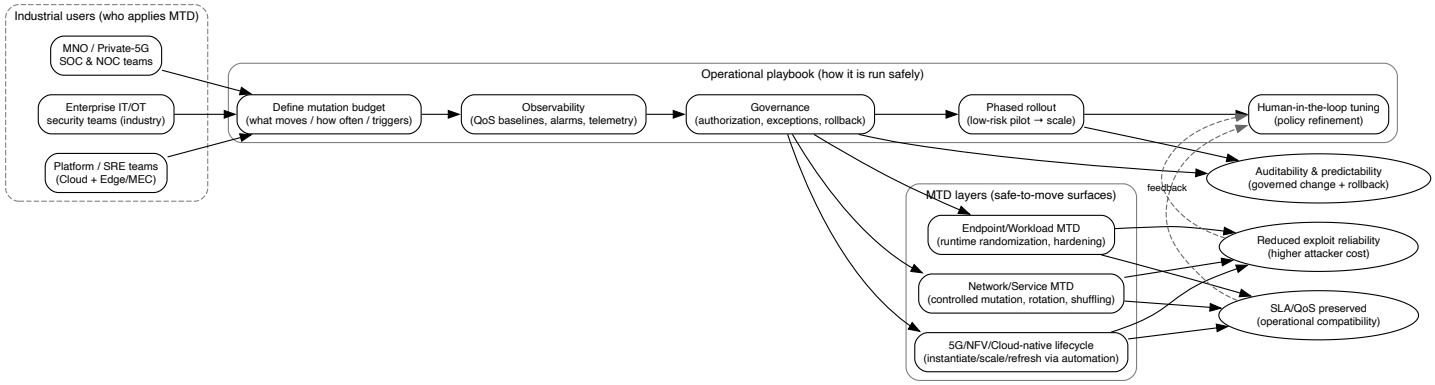


Fig. 7. Industrial view of real-world MTD deployment: primary users, an operational playbook (mutation budget, observability, governance, phased rollout), and layered "safe-to-move" surfaces across endpoint/workload and network/service domains.

impacting incidents. In operational deployments, ML is commonly applied to detect intrusions and anomalies, estimate risk, and forecast near-term attack intensity by learning from multi-source telemetry such as flow statistics, control-plane signaling events, slice-level KPIs, radio measurements, and virtualization logs. These tools are particularly valuable in 5G because the environment is dynamic (mobility, bursty traffic, elastic slices, and distributed MEC), making static signatures and fixed thresholds brittle. Depending on where telemetry is available and where latency budgets permit, ML components may run at the edge/MEC (fast local anomaly alarms), within the core/cloud (global correlation across domains), or alongside SDN and orchestration components (policy-aware detection and response).

Importantly, ML is not only a standalone defense; it also acts as a decision-support layer for MTD. Detection models can provide alerts and risk scores that trigger mutation, rank high-risk targets, or adapt mutation intensity, while forecasting models can predict attack spikes or congestion periods to schedule safer mutation windows. In learning-assisted MTD, RL is used primarily for the actuation policy (selecting what and when to mutate), whereas supervised and unsupervised ML often support the observation layer (detecting attacks and estimating state). The practical value of ML-enabled defense in 5G therefore depends on end-to-end integration: telemetry quality, drift robustness under non-stationary traffic, privacy and data-governance constraints in multi-tenant slicing, and the ability to quantify false alarms and their downstream cost (e.g., unnecessary reconfiguration churn). Table XXVI summarizes common ML defensive tasks, their typical deployment points, how they map to 5G attack types, and how they can be coupled with MTD.

Table XXVII provides a concise mapping between security-of-learning attack classes, RL failure modes, and their MTD-specific operational consequences, along with representative references. Fig. 8 illustrates the RL-driven MTD pipeline and highlights the main adversary intervention points as well as the corresponding defense insertion points.

A. Security of Learning: Attacks and Defenses for RL-Driven MTD

Reinforcement-learning-driven Moving Target Defense (MTD) pipelines rely on telemetry, state construction, and learned policies to trigger network reconfigurations at runtime; consequently, attacks on the learning process can directly translate into unsafe or strategically harmful MTD actions. Following the adversarial ML taxonomy in NISTIR 8269 Tabassi et al. [86] and operational threat perspectives such as MITRE ATLAS MITRE [87], we treat the learning stack (data, model, and reward channel) as a first-class attack surface. Table XXVII maps major attack classes to RL failure modes and MTD-specific consequences, while Fig. 8 illustrates where an adversary can intervene and where defenses can be inserted.

1) Test-time evasion via observation manipulation: Adversaries can degrade a deployed DRL policy by perturbing its observations (e.g., falsified alerts, manipulated flow statistics, or poisoned feature vectors) to induce suboptimal actions without needing to compromise the policy parameters. Early evidence shows that adversarial-example style perturbations can significantly reduce performance of neural network policies at test time, including in black-box settings Huang et al. [88]. Similarly, DQN-based agents can be vulnerable to adversarial observation perturbations with transferability across models, which is especially problematic when similar encoders are reused across sites Behzadan and Munir [89]. In MTD, this can manifest as premature or delayed shuffling/migration decisions that increase operational cost while failing to reduce attacker dwell time.

2) Training-time poisoning and policy induction: If the attacker can influence training data streams (e.g., telemetry logs used for offline RL, simulated episodes, or replay buffers), they can mount poisoning or policy-induction attacks that shape the learned policy toward attacker-favorable behavior. Behzadan and Munir demonstrate policy manipulation/induction against DQN learning via adversarial perturbations during learning Behzadan and Munir [89]. More broadly, poisoning can be analyzed under formal robustness guarantees and certified bounds, as in certified defenses for data poisoning Steinhardt et al. [90]. For RL-based MTD, the practical risk is learning

TABLE XXVI
MACHINE-LEARNING TOOLS FOR CYBER DEFENSE IN 5G AND HOW THEY CAN SUPPORT MTD ACTUATION.

ML defensive task	Typical inputs (telemetry)	Where in 5G	Outputs	Attack types supported	How it enables MTD (examples)
Supervised intrusion detection / classification	Labeled flow features, packet/transaction summaries, API logs, VNF events, known indicators of compromise	Core/cloud IDS, SDN controller analytics, security function VNFs	Attack class or intrusion probability; confidence score	Scanning, malware/C2, DoS patterns, policy abuse, known exploits	Trigger mutation on high-confidence alarms; prioritize targets for mutation; condition RL state with risk score
Unsupervised anomaly detection	Flow counters, signaling rates, per-slice KPIs (latency/jitter/loss), radio counters, resource usage	Edge/MEC (fast alarms), core analytics, slice management plane	Anomaly score; change-point alarms; top contributing features	Zero-day behaviors, stealthy probing, low-rate DoS, data exfiltration indicators	Increase mutation rate only when anomaly persists; select conservative mutations during suspected URLLC impact
Time-series forecasting / early warning	Historical traffic demand, signaling load, DDoS indicators, slice elasticity events	Core/cloud analytics; operator monitoring plane	Forecast of load/attack intensity; confidence interval	DDoS surges, flash-crowd confusion, coordinated probing waves	Schedule mutations in predicted low-risk windows; preemptively harden high-risk slices with low-overhead shuffling
Graph/relational analytics (graph ML)	Topology, service chains, dependency graphs, lateral movement signals, communication graphs	Core and orchestration (NFV/MANO), SDN policy layer	Suspicious subgraph, compromised path likelihood, critical node ranking	Lateral movement, multi-stage attacks, route manipulation campaigns	Select mutation targets that break likely kill-chain paths; relocate/migrate functions that become high-risk chokepoints
Representation learning / self-supervised features	Raw multi-domain telemetry (flows, logs, KPIs) without labels	Core/cloud training; lightweight edge inference	Compact state representation; embeddings for downstream detection/control	Generalizable detection under drift; unknown patterns	Improve RL state quality (more stable features); reduce reliance on brittle hand-crafted state definitions

a policy that appears effective on validation traces but systematically under-reacts (or over-reacts) under attacker-chosen conditions.

3) Backdoors (Trojans) in DRL policies: Backdoor attacks can implant a hidden trigger that causes a DRL agent to switch into malicious behavior while retaining near-normal performance otherwise, which makes them difficult to detect using standard evaluation. TrojDRL shows that Trojan attacks can be realized for DRL agents through minuscule data poisoning and in-band reward modification, yielding policies that behave benignly until a trigger is present Kiourti et al. [91]. In an RL-driven MTD controller, triggers could be crafted as rare traffic patterns, specific alert sequences, or synthetic "incident" signatures that force disabling MTD, selecting predictable configurations, or repeatedly oscillating actions to amplify downtime.

4) Reward channel attacks: reward hacking and tampering: Security issues also arise from the reward specification and reward delivery path. "Reward hacking" is a known failure mode where agents exploit reward misspecifications to achieve high return without achieving the intended objective Amodei et al. [92]. Beyond misspecification, reward tampering considers an agent's incentive to manipulate the reward process itself; Everitt et al. formalize reward function tampering and reward-input tampering and propose design principles to reduce these incentives Everitt et al. [93]. In MTD, reward channels might be implemented via proxy metrics (e.g., IDS alert rate reduction), which can be gamed by attackers who suppress detection signals or create misleading patterns that increase reward while the attacker progresses.

5) Defenses: robust training, certification, and runtime monitoring: Defenses span robust training, formal guarantees, and runtime safeguards. Robust Adversarial RL (RARL) trains an agent in the presence of an adversary to improve robustness to disturbances and model mismatch Pinto et al. [94]. For observation perturbations, principled robustness frameworks for state-adversarial settings can improve resilience across DRL algorithms Zhang et al. [95], while approaches like policy smoothing aim for provable robustness guarantees against adaptive adversaries Kumar et al. [96]. Complementarily, detection-and-denoising pipelines can defend observation-space attacks without risky online adversarial sampling Xiong et al. [97]. For RL-driven MTD, practical deployments should also enforce hard constraints ("shields"), rate-limit reconfigurations, and maintain rollback paths if policy outputs deviate from safe operational envelopes.

6) Implications for RL-driven MTD evaluations: Because learning attacks can invalidate naive comparisons, surveys should explicitly report the assumed attacker capabilities (test-time vs training-time), observation access, and whether the reward channel is trustworthy [98]. When comparing papers, it is important to separate "policy robustness to noisy telemetry" from "robustness to adaptive adversarial manipulation," and to report overhead metrics (e.g., action oscillation rate, migration churn) alongside security gains. We recommend using Table XXVII as a checklist for what each surveyed work covers (or omits) and using Fig. 8 to align reported defenses with the point(s) of intervention in the RL pipeline.

7) Reporting checklist for reproducible and secure learning: To make results comparable and reduce hidden failure modes, each RL-based MTD study should 1) publish the exact state/action/reward definition and reward proxies, 2) provide telemetry integrity assumptions, 3) evaluate at least one adversarial condition (observation perturbation, poisoning, or backdoor), 4 quantify stability/overhead (reconfiguration churn, SLA impact), and 5) include a safe fallback policy. Where code/data cannot be released, authors should still disclose hyperparameters, seeds, and environment dynamics sufficient for reimplementation, and clearly state whether defenses apply at training, inference, or both, as summarized in Table XXVII.

B. Future directions for 6G-oriented MTD

This subsection outlines forward-looking research directions for Moving Target Defense (MTD) in emerging 6G networks. As 6G introduces new physical-layer characteristics,

TABLE XXVII
SECURITY-OF-LEARNING THREAT SURFACES FOR RL-DRIVEN MTD, TYPICAL FAILURE MODES, AND REPRESENTATIVE REFERENCES.

Threat surface	Attack mechanism	RL failure mode	MTD-specific consequence	Representative sources
Test-time observations	Adversarial perturbations / telemetry spoofing	Suboptimal or unsafe actions under small input changes	Wrong shuffling/migration timing; increased churn; reduced security benefit	Huang et al., Behzadan and Munir [88, 89]
Training data / replay	Poisoning, policy induction during learning	Learned policy converges to attacker-favored behavior	Systematically under-reacts to real attacks; over-reacts to benign noise	Behzadan and Munir, Steinhardt et al. [89, 90]
Backdoors in policy	Trojan triggers via tiny poisoning or reward manipulation	Benign behavior except under trigger	Predictable or disabled MTD under trigger; stealthy degradation	Kiourti et al. [91]
Reward specification	Misspecified proxies ("reward hacking")	High return without desired objective	Optimizes proxy (e.g., fewer alerts) while attacker progresses	Amodei et al. [92]
Reward delivery path	Reward tampering / reward-input manipulation	Incentives to manipulate reward channel	Suppress IDS signals or create deceptive patterns to increase reward	Everitt et al. [93]
Defense (training)	Robust/adversarial training, minimax formulations	Improved robustness to disturbances/attacks	More stable MTD decisions under uncertainty/adversary	Pinto et al., Zhang et al. [94, 95]
Defense (inference)	Certification, smoothing, detection+denoising	Bounds or filtering against adaptive attacks	Safer runtime actions; fewer catastrophic mis-actions	Kumar et al., Xiong et al. [96, 97]
Operational framing	Threat modeling knowledge bases	Better coverage of real-world attack patterns	More realistic security testing and mitigations	Tabassi et al., MITRE [86, 87]

programmable propagation, and AI-native control, MTD must evolve beyond conventional network-layer mutations and explicitly account for tighter real-time constraints, cross-domain coupling, and control-loop stability.

1) MTD in THz and ultra-directional communications: 6G THz communications and ultra-directional beamforming shift the attack surface toward beam management and control procedures, including beam selection, beam tracking, and the signaling that coordinates highly directional links. While directionality can reduce some broad and opportunistic scanning, it can also enable new adversarial opportunities such as beam/angle manipulation, targeted disruption of control signaling, and inference attacks that exploit spatial consistency and localization cues. These characteristics motivate MTD mechanisms that diversify not only IP/route-level exposure but also radio/attachment behaviors and beam-access parameters in a controlled manner. Promising directions include beam-schedule diversification, randomized but QoS-bounded access-parameter mutation, and context-aware exposure control that adapts mutation intensity to mobility and link dynamics. A central challenge is ensuring that such mutations do not destabilize beam tracking or violate strict reliability and latency budgets, which requires joint design with radio resource management and careful evaluation under mobility and intermittent blockage.

2) RIS-aware MTD for programmable wireless environments: Reconfigurable intelligent surfaces (RIS) introduce a programmable layer to the wireless environment, where RIS phase configurations and their control interfaces directly influence propagation and link quality. This expands the attack surface to include RIS control channels, configuration inference, spoofing/manipulation attempts, and indirect denial-of-service via malicious surface settings. RIS-aware MTD aims to prevent attackers from exploiting stable or learnable propagation patterns by diversifying RIS configurations, activation patterns, and control policies over time. Key research directions include threat-aware and QoS-constrained RIS mutation policies, secure RIS control-plane designs that resist manipulation and inference, and coordinated RIS mutations that are jointly optimized with routing, slicing, and radio resource allocation. Because abrupt RIS changes can degrade QoS, especially under mobility and ultra-reliability requirements, RIS-aware MTD must prioritize smooth transitions, bounded reconfiguration rates, and stability guarantees.

3) Safe and stable MTD in AI-native 6G networks: 6G is expected to be increasingly AI-native, with learning-based control loops used for orchestration, slicing, routing, radio resource management, and security automation. This enables adaptive, context-aware MTD, but also introduces new vulnerabilities such as data/model poisoning, adversarial inputs, and manipulation of feedback signals, as well as new stability risks when multiple learning controllers interact with frequent mutation actions. Future work should therefore focus on safe and stable learning-assisted MTD that explicitly enforces QoS/SLA constraints, bounds mutation rates, penalizes oscillatory behavior, and remains robust under delayed or noisy telemetry. Additional directions include adversary-aware training and evaluation (where attackers adapt to mutation patterns), multi-objective formulations that jointly optimize security benefit and operational overhead, and verification/assurance techniques that provide predictable behavior guarantees for AI-controlled mutation policies. In practice, these methods must be validated under non-stationary workloads (mobility, bursty traffic, slice elasticity) and realistic multi-domain coordination to ensure that security improvements do not come at the cost of control-loop instability.

XIV. FINAL CONCLUSION

This survey has presented a consolidated view of Moving Target Defense for 5G and beyond networks by keeping one clear focus throughout the manuscript: how MTD can be designed, enforced, and evaluated in softwarized mobile infrastructures subject to slicing, MEC, SDN/NFV, and stringent service constraints. The reviewed literature shows that the field has progressed from general architectural models toward increasingly adaptive and learning-assisted mechanisms, but it also remains fragmented in how it models telemetry, mutation cost, SLA impact, orchestration complexity, and reproducibility.

By organizing prior work with a consistent 5G-oriented taxonomy, linking mutation design choices to operational domains and attack classes, and integrating the discussion

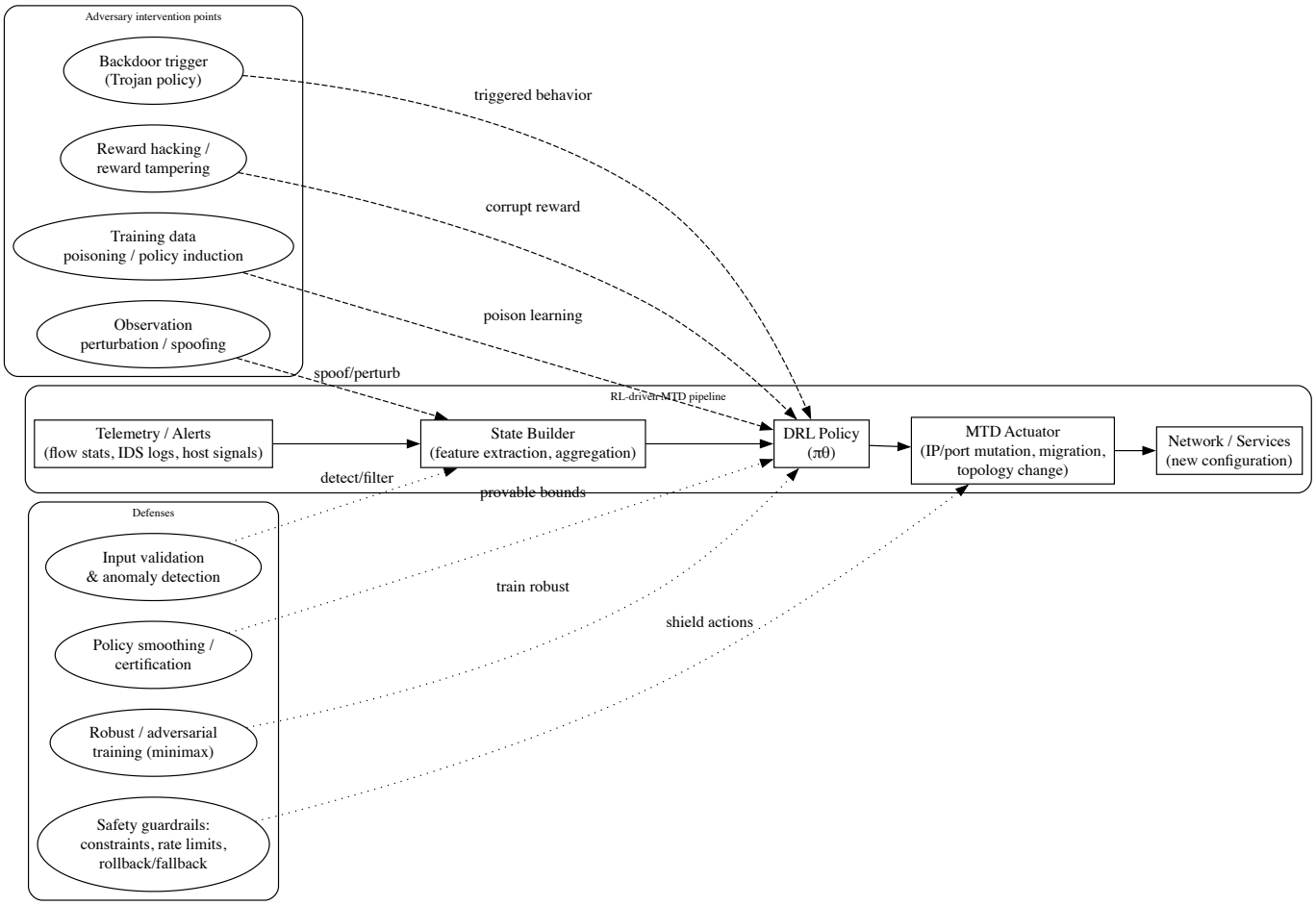


Fig. 8. Attack and defense insertion points in an RL-driven MTD pipeline.

of RL, secure learning, datasets, and deployment constraints into one narrative, the manuscript aims to provide a more coherent foundation for future work. The main message is that effective MTD for 5G/6G will depend not only on stronger mutation algorithms, but also on realistic evaluation, stable control loops, secure telemetry, benchmark availability, and implementation pathways that preserve service quality while raising attacker cost.

ACKNOWLEDGMENT

The research work is 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's Horizon Europe research and innovation programme under the 6G-Path project (Grant No. 101139172).

REFERENCES

[1] C. Benzaïd and T. Taleb, "Ai for beyond 5g networks: a cyber-security defense or offense enabler?" *IEEE Network*, vol. 34, no. 6, pp. 140–147, 2020.

[2] Y. Zhou, G. Cheng, Y. Zhao, Z. Chen, and S. Jiang, "Toward proactive and efficient ddos mitigation in iiot systems: A moving target defense approach," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 4, pp. 2734–2744, 2021.

[3] J. Zheng and A. S. Namin, "A survey on the moving target defense strategies: An architectural perspective," *Journal of Computer Science and Technology*, vol. 34, no. 1, pp. 207–233, 2019.

[4] 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.

[5] Ł. Jalowski, M. Zmuda, and M. Rawski, "A survey on moving target defense for networks: A practical view," *Electronics*, vol. 11, no. 18, p. 2886, 2022.

[6] P. Escaleira, V. A. Cunha, D. Gomes, J. P. Barraca, and R. L. Aguiar, "Moving target defense for the cloud/edge telco environments," *Internet of Things*, vol. 24, p. 100916, 2023.

[7] Z. Abdelhay, Y. Bello, and A. Refaey, "Toward zero-trust 6gc: A software defined perimeter approach with dynamic moving target defense mechanism," *IEEE Wireless Communications*, vol. 31, no. 2, pp. 74–80, 2024.

[8] 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.

[9] T. Zhang, F. Kong, D. Deng, X. Tang, X. Wu, C. Xu, L. Zhu, J. Liu, B. Ai, Z. Han *et al.*, "Moving target defense meets artificial intelligence-driven network: A comprehensive survey," *IEEE Internet of Things Journal*, 2025.

[10] J. Tan, H. Jin, H. Zhang, Y. Zhang, D. Chang, X. Liu, and H. Zhang, "A survey: When moving target defense meets game theory," *Computer Science Review*, vol. 48, p. 100544, 2023.

[11] S. Lakshminarayana, Y. Chen, C. Konstantinou, D. Mashima, and A. K. Srivastava, "Survey of moving target defense in power grids: Design principles, tradeoffs, and future directions," *arXiv preprint*

arXiv:2409.18317, 2024.

[12] N. Saputro, S. Tonyali, A. Aydeger, K. Akkaya, M. A. Rahman, and S. Uluagac, "A review of moving target defense mechanisms for internet of things applications," *Modeling and Design of Secure Internet of Things*, pp. 563–614, 2020.

[13] M. Torquato and M. Vieira, "Moving target defense in cloud computing: A systematic mapping study," *Computers & Security*, vol. 92, p. 101742, 2020.

[14] R. Sun, Y. Zhu, J. Fei, and X. Chen, "A survey on moving target defense: Intelligently affordable, optimized and self-adaptive," *Applied Sciences*, vol. 13, no. 9, p. 5367, 2023.

[15] A. Hamada, S. M. Hassan, S. Samy, M. Azab, and E. Fathalla, "A review: State-of-the-art of integrating ai models with moving-target defense for enhancing iot networks security," in *2024 IEEE 15th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*. IEEE, 2024, pp. 108–114.

[16] S. Sengupta, A. Chowdhary, A. Sabur, A. Alshamrani, D. Huang, and S. Kambhampati, "A survey of moving target defenses for network security," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 1909–1941, 2020.

[17] T. Zhang, C. Xu, Y. Lian, H. Tian, J. Kang, X. Kuang, and D. Niyato, "When moving target defense meets attack prediction in digital twins: A convolutional and hierarchical reinforcement learning approach," *IEEE Journal on Selected Areas in Communications*, 2023.

[18] M. A. Ribeiro, M. S. P. Fonseca, and J. de Santi, "Detecting and mitigating ddos attacks with moving target defense approach based on automated flow classification in sdn networks," *Computers & Security*, vol. 134, p. 103462, 2023.

[19] 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.

[20] T. Zhang, C. Xu, J. Shen, X. Kuang, and L. A. Grieco, "How to disturb network reconnaissance: a moving target defense approach based on deep reinforcement learning," *IEEE Transactions on Information Forensics and Security*, 2023.

[21] T. Eghtesad, Y. Vorobeychik, and A. Laszka, "Adversarial deep reinforcement learning based adaptive moving target defense," in *Decision and Game Theory for Security: 11th International Conference, GameSec 2020, College Park, MD, USA, October 28–30, 2020, Proceedings 11*. Springer, 2020, pp. 58–79.

[22] W. Soussi, M. Christopoulou, G. Xilouris, and G. Gür, "Moving target defense as a proactive defense element for beyond 5g," *IEEE Communications Standards Magazine*, vol. 5, no. 3, pp. 72–79, 2021.

[23] A. Chowdhary, D. Huang, A. Sabur, N. Vadnere, M. Kang, and B. Montrose, "Sdn-based moving target defense using multi-agent reinforcement learning," in *Proceedings of the first International Conference on Autonomous Intelligent Cyber defense Agents (AICA 2021), Paris, France*, 2021, pp. 15–16.

[24] S. Yoon, J.-H. Cho, D. S. Kim, T. J. Moore, F. Free-Nelson, and H. Lim, "Desolater: Deep reinforcement learning-based resource allocation and moving target defense deployment framework," *IEEE Access*, vol. 9, pp. 70 700–70 714, 2021.

[25] X. Chai, Y. Wang, C. Yan, Y. Zhao, W. Chen, and X. Wang, "Dq-motag: deep reinforcement learning-based moving target defense against ddos attacks," in *2020 IEEE Fifth International Conference on Data Science in Cyberspace (DSC)*. IEEE, 2020, pp. 375–379.

[26] C. Gao and Y. Wang, "Reinforcement learning based self-adaptive moving target defense against ddos attacks," in *Journal of Physics: Conference Series*, vol. 1812, no. 1. IOP Publishing, 2021, p. 012039.

[27] H. Li and Z. Zheng, "Robust moving target defense against unknown attacks: A meta-reinforcement learning approach," in *International Conference on Decision and Game Theory for Security*. Springer, 2022, pp. 107–126.

[28] S. Sengupta and S. Kambhampati, "Multi-agent reinforcement learning in bayesian stackelberg markov games for adaptive moving target defense," *arXiv preprint arXiv:2007.10457*, 2020.

[29] X. Xu, H. Hu, and X. Zhu, "Compressing deep learning model for agile moving target defense," in *Journal of Physics: Conference Series*, vol. 2171, no. 1. IOP Publishing, 2022, p. 012032.

[30] S. Sengupta and S. Kambhampati, "Learning movement policies in bayesian stackelberg markov games for adaptive moving target defense," 2020.

[31] A. H. Celdrán, P. M. S. Sánchez, J. Von Der Assen, T. Schenk, G. Bovet, G. M. Pérez, and B. Stiller, "Rl and fingerprinting to select moving target defense mechanisms for zero-day attacks in iot," *IEEE Transactions on Information Forensics and Security*, 2024.

[32] A. Charpentier, N. Boulahia Cuppens, F. Cuppens, and R. Yaich, "Deep reinforcement learning-based defense strategy selection," in *Proceedings of the 17th International Conference on Availability, Reliability and Security*, 2022, pp. 1–11.

[33] Q. Zhang, J.-H. Cho, T. J. Moore, D. D. Kim, H. Lim, and F. Nelson, "Evade: efficient moving target defense for autonomous network topology shuffling using deep reinforcement learning," in *International Conference on Applied Cryptography and Network Security*. Springer, 2023, pp. 555–582.

[34] C. Feng, A. H. Celdran, P. M. S. Sanchez, J. Kreischer, J. von der Assen, G. Bovet, G. M. Perez, and B. Stiller, "Cyberforce: A federated reinforcement learning framework for malware mitigation," *arXiv preprint arXiv:2308.05978*, 2023.

[35] R. Liu and L. Tahvildari, "Towards an uncertainty-aware adaptive decision engine for self-protecting software: an pomdp-based approach," *arXiv preprint arXiv:2308.02134*, 2023.

[36] K. Shang, W. He, S. Zhang, Z. Zhang, J. Xie, and X. Shi, "Deep reinforcement learning-based network moving target defense in dpdk," in *2023 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom)*. IEEE, 2023, pp. 491–499.

[37] E. M. Ghourab, S. Naser, S. Muhaidat, L. Bariah, M. Al-Qutayri, E. Damiani, and P. C. Sofotasios, "Moving target defense approach for secure relay selection in vehicular networks," *Vehicular Communications*, vol. 47, p. 100774, 2024.

[38] T. Zhang, C. Xu, P. Zou, H. Tian, X. Kuang, S. Yang, L. Zhong, and D. Niyato, "How to mitigate ddos intelligently in sd-iov: A moving target defense approach," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 1, pp. 1097–1106, 2022.

[39] W. Soussi, M. Christopoulou, G. Gür, and B. Stiller, "Merlins–moving target defense enhanced with deep-rl for nfv in-depth security," in *2023 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*. IEEE, 2023, pp. 65–71.

[40] W. Soussi, M. Christopoulou, G. Xilouris, E. M. de Oca, V. Lefebvre, G. Gür, and B. Stiller, "Closed-loop security orchestration in the telco cloud for moving target defense," in *NOMS 2023-2023 IEEE/IFIP Network Operations and Management Symposium*. IEEE, 2023, pp. 1–3.

[41] W. Soussi, G. Gür, and B. Stiller, "Moving target defense (mtd) for 6g edge-to-cloud continuum: A cognitive perspective," *IEEE Network*, 2024.

[42] R. Lallouche, A. Alioua, A. Boualouache, and M.-L. Messai, "Deep reinforcement learning-based moving target defense approach to secure network slicing in 5g and beyond," in *2024 20th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*. IEEE, 2024, pp. 695–700.

[43] E. Ghourab, S. Naser, S. Muhaidat, M. Alqutayri, E. Damiani *et al.*, "Physical layer security of wireless communications: When moving target defense meets unmanned aerial vehicles," *Shimaa and Muhaidat, Sami and Alqutayri, Mahmoud and Damiani, Ernesto, Physical Layer Security of Wireless Communications: When Moving Target Defense Meets Unmanned Aerial Vehicles*, 2024.

[44] T. Zhang, C. Xu, B. Zhang, J. Shen, X. Kuang, and L. A. Grieco, "Toward attack-resistant route mutation for vanets: an online and adaptive multiagent reinforcement learning approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 12, pp. 23 254–23 267, 2022.

[45] Y. Lian, T. Zhang, C. Xu, W. Dong, M. Xu, Z. Xiahou, J. Kang, J. Liu, and D. Niyato, "Deep reinforcement learning-based moving target defense for multicast in software-defined satellite networks," in *ICC 2024-IEEE International Conference on Communications*. IEEE, 2024, pp. 4786–4791.

[46] Y. Cao, K. Liu, Y. Lin, L. Wang, and Y. Xia, "Deep reinforcement learning based self-evolving moving target defense approach against unknown attacks," *IEEE Internet of Things Journal*, 2024.

[47] Q. Wang, S. Wu, Z. Wu, J. Hu, Q. He, Y. Ye, and Y. Tang, "Topology switching-based moving target defense against false data injection attacks on a power system," *International Journal of Electrical Power & Energy Systems*, vol. 163, p. 110350, 2024.

[48] 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*, vol. 19, no. 4, pp. 4834–4846, 2022.

[49] T. Zhang, X. Kuang, Z. Zhou, H. Gao, and C. Xu, "An intelligent route mutation mechanism against mixed attack based on security awareness," in *2019 IEEE Global Communications Conference (GLOBE-*

COM). IEEE, 2019, pp. 1–6.

[50] T. Zhang, C. Xu, B. Zhang, X. Kuang, Y. Wang, S. Yang, and G.-M. Muntean, "Dq-rm: Deep reinforcement learning-based route mutation scheme for multimedia services," in *2020 International Wireless Communications and Mobile Computing (IWCMC)*. IEEE, 2020, pp. 291–296.

[51] 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.

[52] Z. Li, Z. Zhou, T. Zhang, and X. Xing, "Marl-motag: Multi-agent reinforcement learning based moving target defense to thwart ddos attacks," in *2022 International Conference on Networking and Network Applications (NaNA)*. IEEE, 2022, pp. 316–321.

[53] G. Abdel Messih, T. Cody, P. Beling, and J.-H. Cho, "Resonant: Reinforcement learning-based moving target defense for credit card fraud detection," in *Proceedings of the 11th ACM Workshop on Adaptive and Autonomous Cyber Defense*, 2024, pp. 13–22.

[54] 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*, 2024.

[55] 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.

[56] 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.

[57] 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.

[58] A. Chowdhary, D. Huang, A. Sabur, N. Vadnere, M. Kang, and B. Montrose, "Sdn-based moving target defense using multi-agent reinforcement learning," in *Proceedings of the 2021 1st International Conference on Autonomous Intelligent Cyber Defense Agents*, 2021.

[59] 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.

[60] S. Macwan and C.-H. Lung, "Investigation of moving target defense technique to prevent poisoning attacks in sdn," in *2019 IEEE World Congress on Services (SERVICES)*, vol. 2642. IEEE, 2019, pp. 178–183.

[61] J. Narantuya, S. Yoon, H. Lim, J.-H. Cho, D. S. Kim, T. Moore, and F. Nelson, "Sdn-based ip shuffling moving target defense with multiple sdn controllers," in *2019 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks–Supplemental Volume (DSN-S)*. IEEE, 2019, pp. 15–16.

[62] 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.

[63] 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.

[64] N. Bandi, H. Tajbakhsh, and M. Analoui, "Fastmove: fast ip switching moving target defense to mitigate ddos attacks," in *2021 IEEE Conference on Dependable and Secure Computing (DSC)*. IEEE, 2021, pp. 1–7.

[65] A. Chowdhary, A. Alshamrani, D. Huang, and H. Liang, "Mtd analysis and evaluation framework in software defined network (mason)," in *Proceedings of the 2018 ACM International Workshop on Security in Software Defined Networks & Network Function Virtualization*, 2018, pp. 43–48.

[66] 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.

[67] S. Debroy, P. Calyam, M. Nguyen, R. L. Neupane, B. Mukherjee, A. K. Eeralla, and K. Salah, "Frequency-minimal utility-maximal moving target defense against ddos in sdn-based systems," *IEEE Transactions on Network and Service Management*, 2020.

[68] M. F. Hyder and M. A. Ismail, "Securing control and data planes from reconnaissance attacks using distributed shadow controllers, reactive and proactive approaches," *IEEE Access*, vol. 9, pp. 21 881–21 894, 2021.

[69] M. Christopoulou, W. Soussi, G. Xilouris, G. Gür, E. Montes de Oca, H. Koumaras, and B. Stiller, "Ai-enabled slice protection exploiting moving target defense in 6g networks," in *2021 Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit), online, 8-11 June 2021*. ZHAW Zürcher Hochschule für Angewandte Wissenschaften, 2021.

[70] R. Upeksha, M. Maduranga, N. C. Lasantha, and N. Aminda, "Hybrid machine learning and moving target defense (mtd) for comprehensive switchport attack detection," in *2024 8th SLAAI International Conference on Artificial Intelligence (SLAAI-ICAI)*. IEEE, 2024, pp. 1–6.

[71] Y. Zhou, G. Cheng, S. Yu, Z. Chen, and Y. Hu, "Mtdroid: A moving target defense based android malware detector against evasion attacks," *IEEE Transactions on Information Forensics and Security*, 2024.

[72] X. Xu, H. Hu, Y. Liu, H. Zhang, and D. Chang, "An adaptive ip hopping approach for moving target defense using a light-weight cnn detector," *Security and Communication Networks*, vol. 2021, 2021.

[73] V. A. Cunha, D. Corujo, J. P. Barraca, and R. L. Aguiar, "Moving target defense to set network slicing security as a kpi," *Internet Technology Letters*, 2020.

[74] S.-H. Lee, K. Kim, Y. Kim, and K.-W. Park, "Mtd-diorama: Moving target defense visualization engine for systematic cybersecurity strategy orchestration," *Sensors*, vol. 24, no. 13, p. 4369, 2024.

[75] V. A. Cunha, D. Corujo, J. P. Barraca, and R. L. Aguiar, "Totp moving target defense for sensitive network services," *Pervasive and Mobile Computing*, vol. 74, p. 101412, 2021.

[76] Z. Zhao, F. Liu, D. Gong, L. Chen, F. Xiang, and Y. Li, "An sdn-based ip hopping communication scheme against scanning attack," in *2017 IEEE 9th International Conference on Communication Software and Networks (ICCSN)*. IEEE, 2017, pp. 559–564.

[77] V. Kansal and M. Dave, "Proactive ddos attack mitigation in cloud-fog environment using moving target defense," *arXiv preprint arXiv:2012.01964*, 2020.

[78] H. Yuwen, L. Zhang, Z. Wang, and Y. Kong, "Probability-based delay scheme for resisting sdn scanning," in *2016 2nd IEEE International Conference on Computer and Communications (ICCC)*. IEEE, 2016, pp. 1096–1101.

[79] J.-G. Ki and K.-T. Lee, "Malicious attack success probability on the change of vulnerable surfaces in mtd-sdr system," *The Journal of the Institute of Internet, Broadcasting and Communication*, vol. 18, no. 5, pp. 55–62, 2018.

[80] J. Giraldo and A. A. Cardenas, "Moving target defense for attack mitigation in multi-vehicle systems," in *Proactive and Dynamic Network Defense*. Springer, 2019, pp. 163–190.

[81] Z. Zhang, R. Deng, D. K. Yau, P. Cheng, and J. Chen, "On hiddenness of moving target defense against false data injection attacks on power grid," *ACM Transactions on Cyber-Physical Systems*, vol. 4, no. 3, pp. 1–29, 2020.

[82] B. Liu and H. Wu, "Optimal d-facts placement in moving target defense against false data injection attacks," *IEEE Transactions on Smart Grid*, vol. 11, no. 5, pp. 4345–4357, 2020.

[83] W. Zhan, C. Xu, X. Sun, and J. Zou, "Toward optimal connection management for massive machine-type communications in 5g system," *IEEE Internet of Things Journal*, vol. 8, no. 17, pp. 13 237–13 250, 2021.

[84] J. Yu, Q. Li, and L. Li, "Localization of coordinated cyber-physical attacks in power grids using moving target defense and machine learning," *Electronics*, vol. 13, no. 12, p. 2256, 2024.

[85] J. Shen, T. Zhang, B. Zhang, W. Ji, X. Kuang, and C. Xu, "Ppo-rm: proximal policy optimization based route mutation for multimedia services," in *2021 International Wireless Communications and Mobile Computing (IWCMC)*. IEEE, 2021, pp. 35–40.

[86] E. Tabassi *et al.*, "A taxonomy and terminology of adversarial machine learning," NIST Interagency Report (NISTIR) 8269, 2019, national Institute of Standards and Technology (NIST), CSRC.

[87] MITRE, "MITRE ATLAS (adversarial threat landscape for ai systems)," Online knowledge base, 2025, https://atlas.mitre.org/.

[88] S. Huang, N. Papernot, I. Goodfellow, Y. Duan, and P. Abbeel, "Adversarial attacks on neural network policies," *arXiv preprint arXiv:1702.02284*, 2017.

[89] V. Behzadan and A. Munir, "Vulnerability of deep reinforcement learning to policy induction attacks," *arXiv preprint arXiv:1701.04143*, 2017.

[90] J. Steinhardt, P. W. Koh, and P. Liang, "Certified defenses for data poisoning attacks," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[91] P. Kiourti, K. Wardega, S. Jha, and W. Li, "Trojdrl: Trojan attacks on deep reinforcement learning agents," *arXiv preprint arXiv:1903.06638*,

2019.
[92] D. Amodei, C. Olah, J. Steinhardt, P. Christiano, J. Schulman, and D. Mané, "Concrete problems in ai safety," *arXiv preprint arXiv:1606.06565*, 2016.
[93] T. Everitt, M. Hutter, J. Leike *et al.*, "Reward tampering problems and solutions in reinforcement learning: A causal influence diagram perspective," *arXiv preprint arXiv:1908.04734*, 2019.
[94] L. Pinto, J. Davidson, R. Sukthankar, and A. Gupta, "Robust adversarial reinforcement learning," *Proceedings of the 34th International Conference on Machine Learning (ICML)*, 2017.
[95] H. Zhang, H. Chen, C. Xiao, B. Li, M. Liu, D. Boning, and C.-J. Hsieh, "Robust deep reinforcement learning against adversarial perturbations on state observations," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.
[96] A. Kumar, A. Levine, and S. Feizi, "Policy smoothing for provably robust reinforcement learning," in *International Conference on Learning Representations (ICLR)*, 2022.
[97] Z. Xiong, J. Eappen, H. Zhu, and S. Jagannathan, "Defending observation attacks in deep reinforcement learning via detection and denoising," *arXiv preprint arXiv:2206.07188*, 2022.
[98] 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.
[99] 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.
[100] 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.

APPENDIX A
ADDITIONAL ILLUSTRATIVE EXAMPLE OF MTD TARGET MUTATION

To further clarify the intuition behind the end-to-end MTD workflow in Fig. I, we provide an additional toy example that complements the concept shown in Fig. 1. This appendix figure visualizes how *target mutation* reshuffles host availability and access mappings so that reconnaissance results obtained before mutation become stale after mutation.

APPENDIX B
COMPARATIVE SCENARIO ANALYSIS

To understand the performance of different MTD methods and enable a fair comparison (Table XI), we define a *common baseline scenario*. This baseline, obtained by the model in [99], represents a simplified but generalizable network infrastructure consisting of five hosts/virtual machines (h1 h5) with heterogeneous vulnerability levels and two critical services (S1 and S2) as high-value targets. An active adversary is assumed to conduct reconnaissance (e.g., scanning) and to launch distributed denial-of-service (DDoS) attacks.

The reason for adopting such a shared scenario is fourfold: 1) it provides direct comparability, since all methods are evaluated on the same infrastructure, making their differences in target selection, mutation type (IP hopping, VM migration, slice-level resource allocation, or classifier switching), and the defensive effect is more evident; 2) it offers a visual and intuitive representation, allowing readers to easily see how each method operates and how the attacker's behavior is disrupted; 3) it supports a situation where benefits such as adaptability or attack mitigation can be contrasted with drawbacks such as computational overhead or QoS degradation; and 4) It highlights the relevance to 5G/6G environments, as the same baseline mapping allows us to assess which approaches are more practical and effective for large-scale, dynamic network infrastructures.

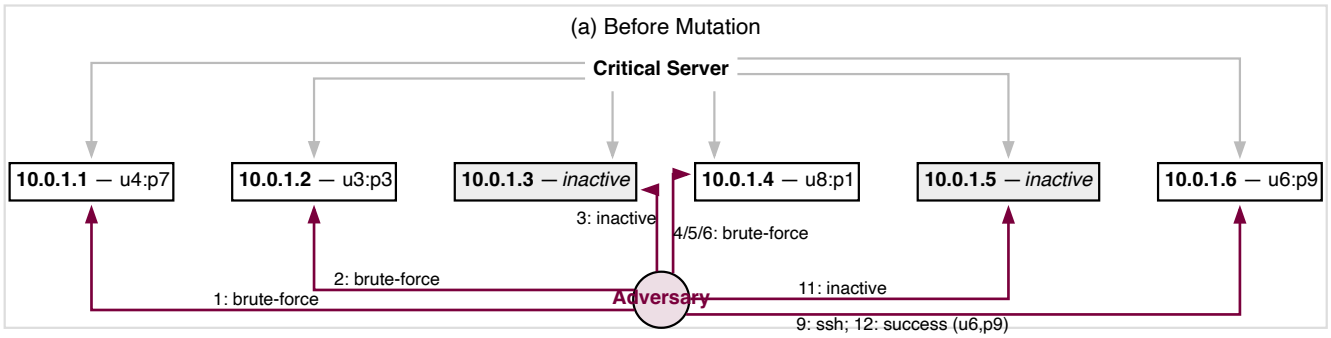
A. Baseline (MTD)

In the baseline scenario (inspired by [99, 100]), the network infrastructure contains five hosts/virtual machines (h1 h5) with different compromise probabilities and two critical services (S1 and S2). An attacker performs reconnaissance, such as scanning the hosts, and then launches a DDoS attack through compromised hosts against the critical services.

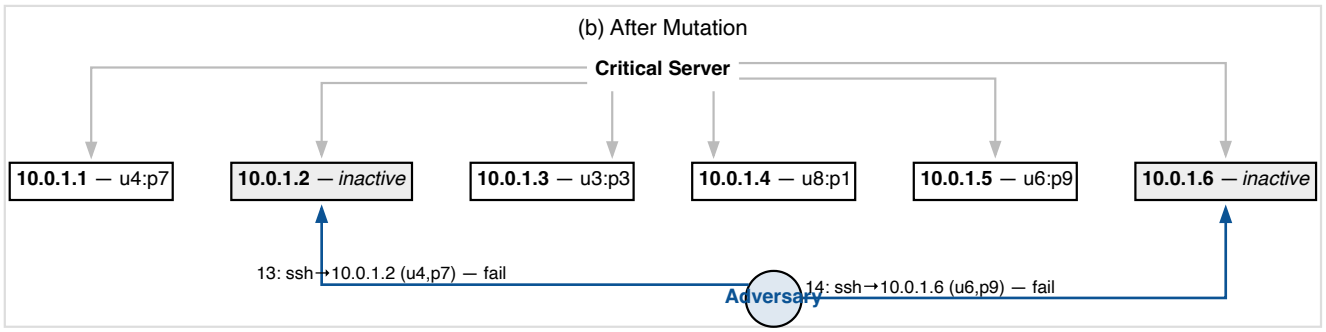
No Moving Target Defense (MTD) mechanisms are applied in this case: the IP addresses, ports, and VM placements remain static during the entire attack. As a result, all reconnaissance data collected by the attacker remains valid, allowing repeated and successful exploitation. This setup therefore, represents the reference point against which all subsequent MTD strategies are evaluated. The figure highlights the attacker's direct scanning of all hosts and the straightforward compromise path toward the critical services.

B. Zhang 2023 (Disturb)

The Disturb approach [20] leverages the Advantage Actor-Critic (A2C) reinforcement learning model to dynamically



(a) Before mutation



(b) After mutation

Fig. 9. Additional example (Appendix) of Moving Target Defense (MTD) target mutation, complementing the concept in Fig. 1. Panel (a) shows the pre-mutation setting where the adversary probes hosts and may temporarily succeed when an active target and valid credential mapping align with a reachable path. Panel (b) shows the post-mutation setting where the defender reshuffles target states and access mappings (e.g., active/inactive hosts and credential-to-host associations), invalidating previously discovered paths/targets and causing subsequent connection attempts to fail. (complementing the concept in Fig. I.)

select hosts for mutation. In the baseline network, hosts h3 and h4 exhibit the highest probability of being scanned and compromised. The RL agent identifies these high-risk hosts and periodically applies IP address shuffling to them.

This proactive mutation invalidates the reconnaissance data collected by the attacker, forcing repeated scanning and significantly reducing the success rate of compromise. The key advantage of this method is its ability to **disrupt reconnaissance at the host level**, making it harder for the attacker to maintain accurate knowledge of the network. However, frequent IP address changes introduce additional operational cost, especially in large-scale 5G deployments, where maintaining session continuity and QoS is critical. Fig. 11 illustrates the periodic IP shuffling applied to high-risk hosts to disrupt reconnaissance.

C. Chowdhary 2021 (SDN MARL)

The approach proposed by Chowdhary et al. [23] introduces a multi-agent reinforcement learning (MARL) model within a software-defined networking (SDN) environment. Unlike the Disturb method, which focuses only on IP shuffling, this model dynamically decides whether to apply *VM migration* or *IP mutation* for specific hosts. The attacker's level of access guides the decision: hosts with greater exposure (e.g., h2 and h4 in the baseline scenario) are prioritized for mutation or migration.

The main advantage of this approach lies in its adaptability to dynamic environments, as multiple agents cooperate to determine optimal defense actions in real time. However, VM migration introduces significant computational and network overhead, which may affect service availability in large-scale 5G networks. Fig. 12 shows how SDN MARL prioritizes high-exposure hosts for VM migration and/or IP mutation.

D. Yoon 2021 (DESOLATER)

The DESOLATER framework [24] applies a multi-agent reinforcement learning (MARL) strategy in vehicular and slice-based networks. Instead of targeting individual hosts, the defense actions are taken at the *slice level*. Each slice is characterized by metrics such as maximum traffic load, packet loss, drop rate caused by shuffling, and vulnerability score. Based on these states, the MARL agents decide both the *bandwidth allocation* and the *IP shuffling interval* for each slice.

For example, in the baseline scenario, Slice S1 may be assigned more bandwidth and a longer shuffling interval (e.g., BW$\uparrow$, $T = 12$s), while Slice S2 may receive reduced bandwidth and a shorter interval (e.g., BW$\downarrow$, $T = 8$s), depending

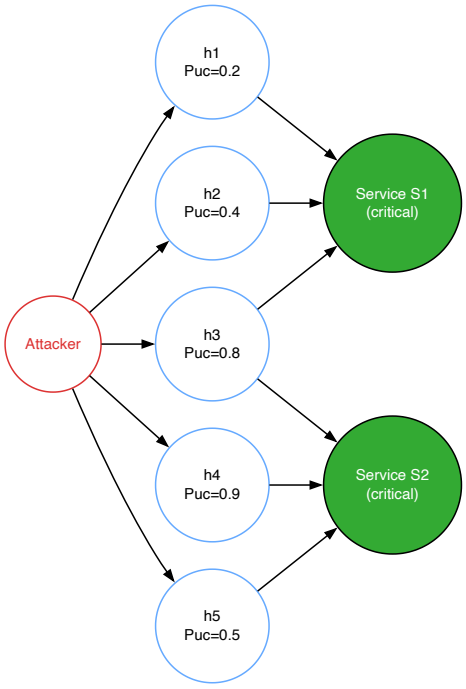


Fig. 10. Baseline scenario without MTD: static IPs and configurations allow the attacker to continuously exploit reconnaissance data and compromise critical services.

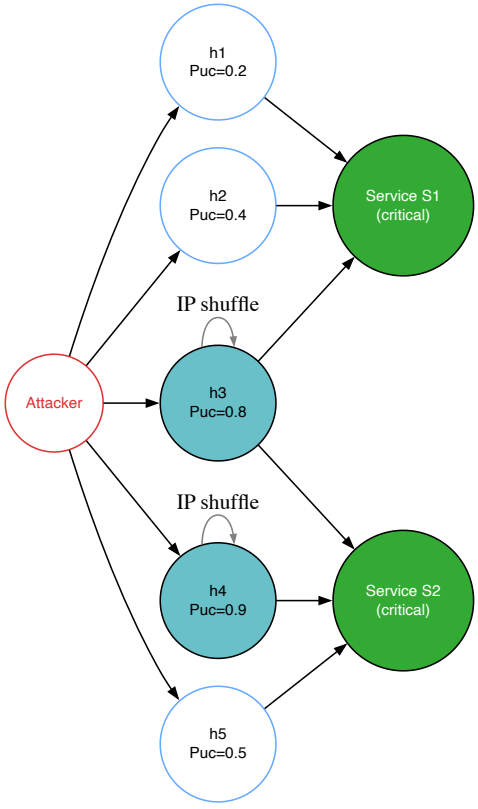


Fig. 11. Zhang 2023 (Disturb): high-risk hosts (e.g., h3 and h4) undergo periodic IP shuffling to invalidate reconnaissance and reduce the success of compromise.

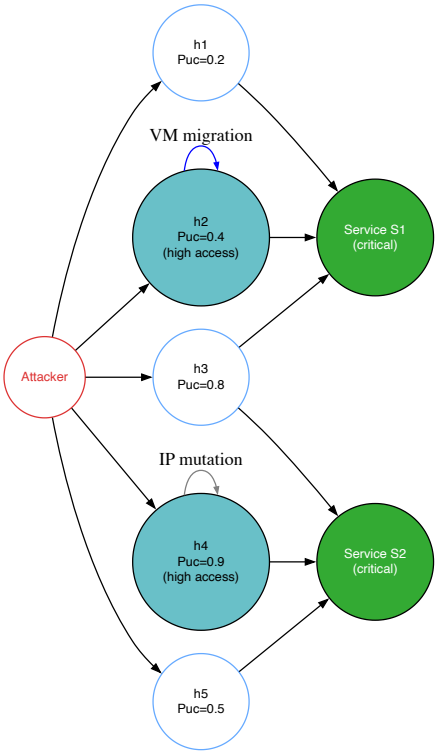


Fig. 12. Chowdhary 2021 (SDN MARL): hosts with higher attacker access (e.g., h2 and h4) are defended using VM migration or IP mutation based on multi-agent RL decisions.

on its risk profile. The strength of this approach lies in its ability to **integrate resource management with security**, which is critical in 5G networks where slicing is a core feature. The limitation, however, is the increased complexity of coordinating slice-level parameters and the possible QoS degradation if not tuned properly. Fig. 13 summarizes the slice-level MARL decisions over bandwidth allocation and IP shuffling intervals.

E. Chai 2020 (DQ-MOTAG)

The DQ-MOTAG framework [25] employs a Deep Q-Network (DQN) to determine the optimal shuffling period for proxy servers to defend against DDoS attacks. In the baseline scenario, multiple users connect to proxies (e.g., $s_{i,j}$ states), and the attacker attempts to overwhelm them with malicious traffic. The RL agent monitors these connections and learns to choose an effective shuffling period (e.g., 10 seconds) that minimizes the impact of DDoS while avoiding excessive reconfiguration overhead.

The primary benefit of this method is its ability to **automatically learn the most effective shuffle timing** based on observed attack dynamics. This adaptability significantly reduces the success rate of DDoS floods. However, its limitation lies in the need for extensive training with realistic traffic and attack datasets; otherwise, the learned policy may struggle to generalize to unseen attack patterns. Fig. 14 illustrates how the DQN agent learns an effective proxy shuffling period under DDoS pressure.

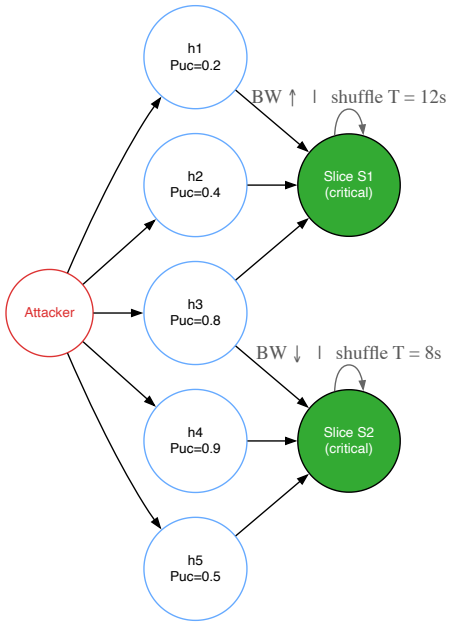


Fig. 13. Yoon 2021 (DESOLATER): MARL agents manage slice-level defenses by jointly deciding bandwidth allocation and shuffling intervals.

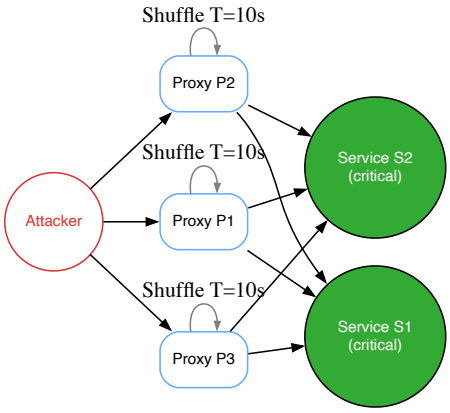


Fig. 14. Chai 2020 (DQ-MOTAG): a DQN agent learns the optimal proxy shuffling period (e.g., 10s) to minimize DDoS success while controlling reconfiguration cost.

F. Li 2022 (Meta-RL)

Li et al. [27] propose a Meta-Reinforcement Learning (Meta-RL) approach that rapidly adapts to new attack conditions. Instead of applying static policies, the Meta-RL agent generates the *next system configuration*, which may include VM migration, IP mutation, or topology adjustments. This enables the defense system to respond quickly when the attacker changes strategy.

In the baseline scenario, the RL policy detects that hosts h3 and h4 are under high compromise probability and produces a new configuration where these VMs are migrated to different physical nodes and their IP addresses are reshuffled. The reward function balances three factors: minimizing migration cost, reducing system loss from compromise, and adapting

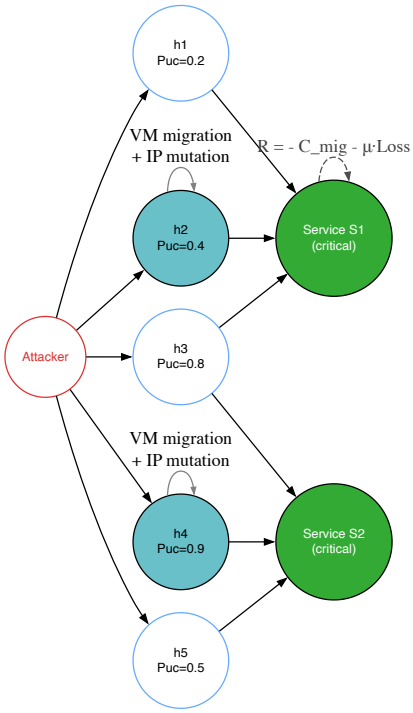


Fig. 15. Li 2022 (Meta-RL): the agent generates the next configuration (e.g., VM migration + IP shuffling) to adapt quickly to evolving attacks.

quickly to threats.

The key advantage of Meta-RL is its **fast adaptability**, which is particularly useful in dynamic 5G environments with frequent context changes. Its limitation, however, is the high computational overhead of maintaining a constantly adapting policy, which can be challenging for large-scale and real-time deployments. Fig. 15 depicts the Meta-RL agent generating the next configuration (e.g., VM migration and IP shuffling) to adapt to evolving attacks.

G. Zhou 2024 (Resource-aware DRL)

The resource-aware DRL framework [54] extends MTD by jointly considering both *security* and *resource management*. In this approach, the environment state of each service or slice includes utilization level, remaining resource quota, and vulnerability score. The DRL agent then decides among three actions: 1) VM migration, 2) IP/port mutation, and 3) service scaling or replication.

In the baseline scenario, the agent observes that Slice S1 has high vulnerability and limited resources. Consequently, the policy migrates its virtual functions to a safer node while also applying IP mutation. At the same time, Slice S2 may undergo port hopping to reduce exploitability.

The significant advantage of this approach is its **multi-dimensional defense**, which jointly optimizes QoS and security objectives. This makes it highly suitable for 5G and beyond networks, where slicing and resource constraints are tightly coupled with security. On the downside, this method requires precise and timely threat intelligence data, and its training complexity can be high due to the enlarged action/state

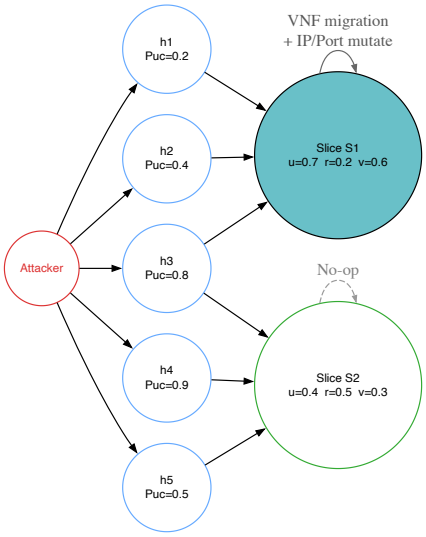


Fig. 16. Zhou 2024 (Resource-aware DRL): the agent jointly decides VM migration, IP/port mutation, and scaling to balance security and QoS.

space. Fig. 16 shows the joint optimization of migration, IP/port mutation, and scaling to balance security and QoS.

H. Abdel 2024 (RESONANT)

The RESONANT framework [53] applies deep reinforcement learning to dynamically switch between multiple classifiers used for intrusion detection and fraud prevention. Each candidate model (M1..Mk) is characterized by attributes such as accuracy, vulnerability to adversarial evasion, and switching cost. The DRL agent selects the optimal classifier for each round based on the current threat landscape.

In the baseline scenario, the agent observes adversarial traffic targeting the network. Instead of relying on a static IDS model, it switches from a high-accuracy but vulnerable classifier to a more robust one, even if its accuracy is slightly lower. The reward function balances three factors: Improved detection accuracy, reduced vulnerability, and minimized the cost of frequent switching.

The strength of RESONANT lies in its flexibility and resilience, as it ensures that no single classifier becomes a permanent target for adaptive attackers. However, its limitation is the added complexity of training and maintaining multiple models simultaneously, which may require significant resources in real-time 5G deployments. Fig. 17 illustrates the DRL-based classifier switching strategy used to balance accuracy, robustness, and switching cost.

I. Gao 2021 (Tabular RL)

Gao et al. [26] present a lightweight tabular reinforcement learning (Q-learning) approach for MTD. The environment state captures whether recent attack cycles have targeted specific hosts. Based on this historical state, the RL agent decides between two actions: *Defend* (activate MTD) or *No-defend*.

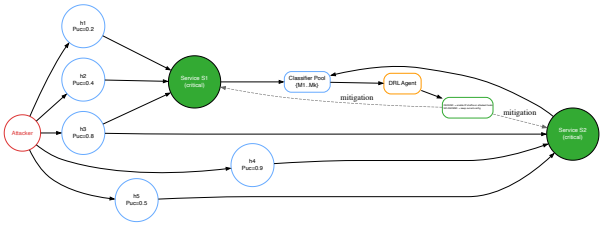


Fig. 17. Abdel 2024 (RESONANT): DRL-based dynamic classifier switching balances detection accuracy, robustness, and switching cost.

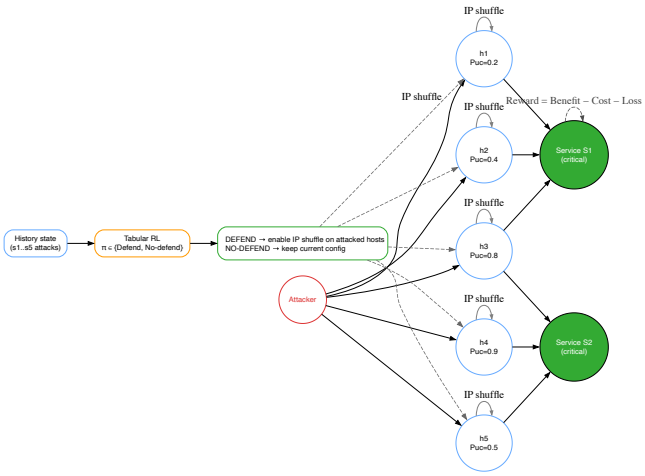


Fig. 18. Gao 2021 (Tabular RL): a simple Q-learning agent decides whether to activate IP shuffling globally based on historical attack states.

In the baseline scenario, when repeated attacks are detected against hosts such as h3 and h4, the agent triggers the defense action, which applies IP shuffling across all hosts. If no significant attack pattern is observed, the agent remains idle to reduce overhead. The reward function evaluates the trade-off between security benefits, defense costs, and system losses.

The main advantage of this method is its **simplicity and low implementation cost**, making it suitable for resource-constrained or testbed environments. However, its limitation lies in the reduced adaptability: tabular Q-learning does not scale well to the highly dynamic and large state spaces characteristic of real-world 5G networks.

J. Comparative Scenario Analysis and Visualization

To complement the taxonomy, models, and tabular summaries presented in the main body of the paper, we provide in this appendix a comparative scenario-based analysis of key RL-driven MTD methods. The goal of this visualization-driven appendix is (Table XI):

- It enables direct comparability of diverse MTD techniques (e.g., IP shuffle, VM migration, slice-level allocation, classifier switching) since each approach is mapped to the same underlying network template.
- It provides a visual and intuitive representation of how each method operates, showing attacker paths, defender actions, and the resulting disruption to reconnaissance or attack campaigns.

The sequence of figures (Fig. 10 18) presents nine scenarios: the baseline (no defense) and eight representative RL-based MTD strategies. Each subsection includes 1) a brief description of the model and its main mechanism, 2) the benefit it offers in terms of security (e.g., disrupting reconnaissance, reducing DDoS impact, balancing security and QoS), and 3) the limitations or overheads (e.g., IP churn cost, VM migration latency, training complexity).

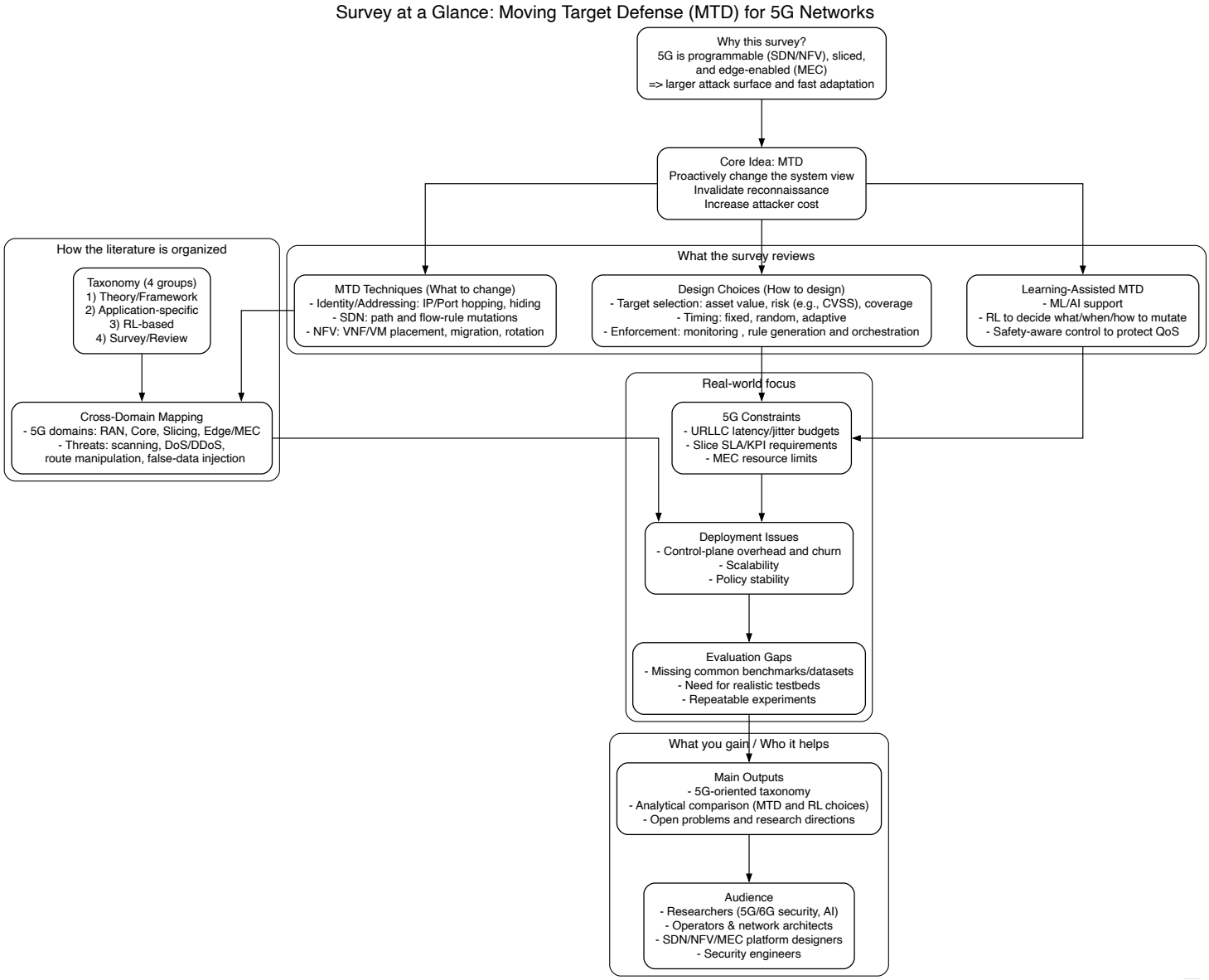


Fig. 19. Survey at a glance: scope, taxonomy, mappings, practical constraints, and target audience for Moving Target Defense (MTD) in 5G networks.

APPENDIX C
REFERENCE MAP OF THE REVIEWED LITERATURE

As shown in Figure 19, this survey explains why MTD is needed in 5G networks that use SDN/NFV, network slicing, and MEC. It reviews the main MTD technique families, including identity and addressing changes, SDN routing and flow-rule changes, and NFV service placement changes (e.g., migration and rotation). The survey also summarizes the main design steps: how to select mutation targets (based on asset importance, coverage goals, and risk indicators such as CVSS), how to choose mutation timing (fixed, random, or adaptive), and how to enforce changes using monitoring, rule generation, and orchestration. In addition, it highlights learning-assisted MTD, especially reinforcement learning, to decide what to change and when, while keeping service quality. Finally, it maps prior work to 5G domains and attack types, and it discusses real deployment limits (URLLC latency/jitter, control-plane overhead, scalability, and policy stability) as well as evaluation gaps such as missing benchmarks, datasets, and realistic testbeds.

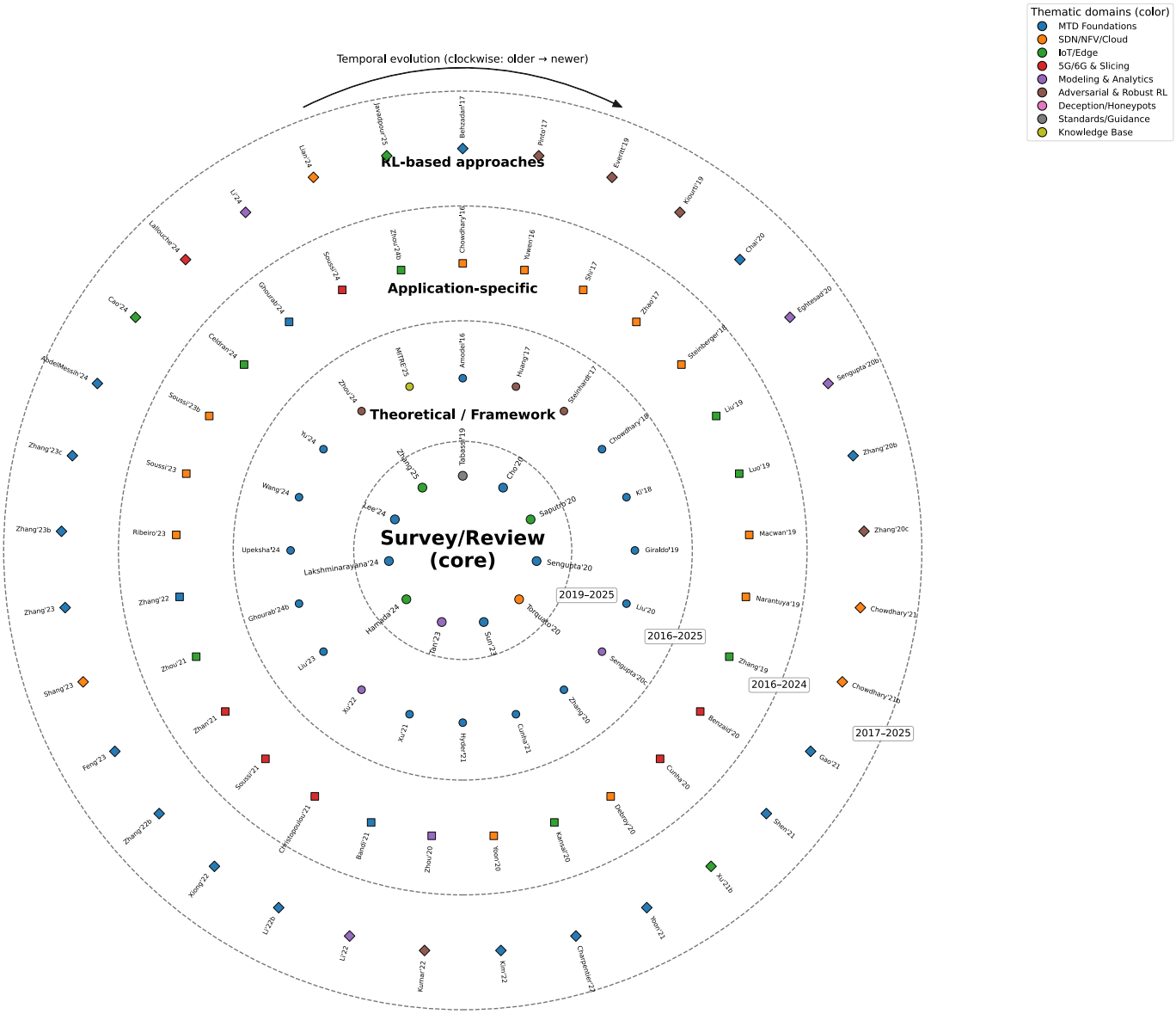


Fig. 20. Concentric reference map of the reviewed literature (center: Survey/Review; inner: Theoretical/Framework; middle: Application-specific; outer: RL-based). Colors show thematic domains and the ordering reflects time.

APPENDIX D
REFERENCE VISUAL MAP OF THE RELATED WORKS

This appendix provides a visual map of the related work used in this paper. Figure 20 shows a concentric-circle diagram that groups the reviewed references into four layers. The center node represents survey and review papers. The inner ring includes theoretical and framework studies. The middle ring contains application-specific works. The outer ring includes reinforcement learning (RL)-based approaches. Colors indicate different thematic domains, and the ordering inside each layer follows time, so the reader can see how the research has evolved.

Moving Target Defense in 5G and Beyond Networks: A Comprehensive Survey and Research Directions

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.

Bibliography Authors

Amir Javadpour is a Senior Cybersecurity Researcher MOSA!C Lab / ICTFICIAL Oy, cybersecurity researcher with extensive academic and professional experience in computer science, network security, and intelligent computing. He received his Ph.D. in Computer Science, with a research focus on mathematics and cybersecurity, from Guangzhou University, China, in 2020. His research interests encompass cybersecurity, cloud computing, Software-Defined Networking (SDN), big data analytics, Intrusion Detection Systems (IDS), the Internet of Things (IoT), Moving Target Defence (MTD), machine learning, and optimization algorithms. Throughout his academic and professional career, he has collaborated with international researchers and contributed to numerous peer-reviewed journal articles and conference papers addressing emerging challenges in cybersecurity, intelligent networks, cloud infrastructures, and data-driven computing systems. Dr. Javadpour has served as a reviewer for several internationally recognized journals, including IEEE Transactions on Cloud Computing, IEEE Transactions on Network Science and Engineering, ACM Transactions on Internet Technology, and various journals published by Springer and Elsevier. He has also actively contributed to the international research community as a Technical Program Committee (TPC) member for several academic conferences. His international research experience includes active participation in European-funded research projects and consortia, including INSPIRE-5Gplus and RIGOUROUS. Through these collaborations, he has contributed to research published in leading journals and conferences, including IEEE Transactions on Industrial Informatics, IEEE Transactions on Information Forensics and Security, IEEE Transactions on Network and Service Management, ACM Transactions on Sensor Networks, and IEEE GLOBECOM. In addition to his research activities, Dr. Javadpour has experience mentoring and supervising master's and doctoral students. His academic leadership, international collaborations, participation in funded research projects, and experience in conducting independent research have strengthened his ability to lead multidisciplinary research teams and develop innovative solutions for complex cybersecurity challenges. His current research focuses on secure, intelligent, and resilient computing systems, with particular emphasis on next-generation networks, cloud-edge infrastructures, AI-driven cybersecurity, adaptive defence mechanisms, and trustworthy intelligent systems. Dr. Javadpour has also been recognized among the World's Top 2% Scientists in the Stanford University ranking, reflecting the scholarly impact, international visibility, and sustained influence of his research contributions. This recognition is supported by his extensive publication record, participation in international and European-funded projects, contributions to high-impact journals and conferences, supervision of postgraduate researchers, and continued commitment to advancing scientific knowledge in cybersecurity and intelligent computing.

Prof. Tarik Taleb received the B.E. degree with distinction in Information Engineering and the M.Sc. and Ph.D. degrees in Information Sciences from Tohoku University, Sendai, Japan, in 2001, 2003, and 2005, respectively. He is currently a Full Professor at Ruhr University Bochum (RUB), Germany, where he leads research activities on next-generation mobile and distributed systems. Prior to joining RUB, he was a Professor at the University of Oulu, Finland (2018–2023), and an Associate Professor at Aalto University, Finland (2014–2021). Earlier in his career, he served as a Senior Researcher and 3GPP Standards Expert at NEC Europe Ltd., Heidelberg, Germany, where he contributed to the evolution of mobile network architectures and standardization activities. Before joining NEC, he was an Assistant Professor at the Graduate School of Information Sciences, Tohoku University, Japan, working in a research laboratory fully funded by KDDI. He also held a Research Fellowship at the Intelligent Cosmos Research Institute, Japan, from 2005 to 2006. Prof. Taleb is widely recognized for his pioneering contributions to mobile network softwarization, network slicing, cloud-edge continuum management, and autonomous networking. His current research interests include autonomous network and service management, edge-cloud continuum systems, network softwarization and slicing, software-defined security, and AI-native communication networks.

Forough Ja'fari is a Senior Researcher in cybersecurity and computer science. She received her Bachelor's degree from Sharif University of Technology and her Master's degree in Computer Network Engineering from Yazd University, Iran. She is a visiting scholar researcher at Guangzhou University, China. Cloud computing, software-defined Networking (SDN), cyber deception, Intrusion Detection Systems (IDS), Internet of Things (IoT), Moving Target Defence (MTD), and Machine Learning are some of her research interests. She is currently a Guest Editor (GE) of Cluster Computing (CLUS) Journal and a reviewer for several journals and conferences.

Chafika Benzaïd is currently a senior research fellow at University of Oulu, Finland. Between Nov. 2018 and Dec. 2021, she was senior researcher at Aalto University. Before that, she worked as an associate professor at University of Sciences and Technology Houari Boumediene (USTHB). She holds Engineer, Magister and "Doctorat ès Sciences" degrees from USTHB. Her research interests lie in the field of 5G/6G, SDN, Network Security, AI Security, and AI/ML for zero-touch security management. She is an ACM professional member.